Game Design Concepts:

An experiment in game design and teaching

Ian Schreiber

Syllabus and Schedule

By ai864

Schedule:

This class runs from **Monday, June 29** through **Sunday, September 6**. Posts appear on the blog Mondays and Thursdays each week at **noon GMT**. Discussions and sharing of ideas happen on a continual basis.

Textbooks:

This course has one required text, and two recommended texts that will be referenced in several places and provide good "next steps" after the summer course ends.

Required Text: *Challenges for Game Designers*, by Brathwaite & Schreiber. This book covers a lot of basic information on both practical and theoretical game design, and we will be using it heavily, supplemented with some readings from other online sources. Yes, I am one of the authors. The reason Brenda and I wrote this book was because we wanted a text to use in our classes, and nothing like it existed at the time… so we made our own.

Recommended Texts:

Understanding Comics: The Invisible Art, by McCloud. While this book claims to be about comics, many of the lessons within can be applied to game design and other forms of art. It also happens to be a comic book itself, and fun to read.

A Theory of Fun for Game Design, by Koster. This book shows the similarities between game design and education, with a good discussion of the concept of Flow. Half text and half cartoons, this short book flows nicely and can be read in an afternoon or two.

Course Description:

This course provides students with a theoretical and conceptual understanding of the field of game design, along with practical exposure to the process of creating a game. Topics covered include iteration, rapid prototyping, mechanics, dynamics, flow theory, the nature of fun, game balance, and user interface design. Primary focus is on non-digital games.

Course Objectives:

In this class, we will discuss games and game design. We will discover what the components of games are, and what parts of games are influenced by their design. We will learn several ways to approach the design of a game, and processes and best practices for prototyping, playtesting and balancing a game after it has been designed.

Student Learning Outcomes:

By the end of this course, you will be familiar with the (relatively small) body of work that is accepted in the game industry as the theoretical foundation of game design. You will also be comfortable enough in processes to start designing your own games, as well as critically analyzing other people's games.

A Note About Change:

As this entire course is an experiment, the schedule in this syllabus is subject to change based on the needs of the students and the overall pace of the course.

Content:

Date	Topics Covered
M 6/29	• Overview of games and design • Critical vocabulary: what is a game?
Th 7/2	• What is game design? • Iteration and rapid prototyping.
M 7/6	• Formal elements of games
Th 7/9	• Overview of the game design process • Idea generation, brainstorming, and paper prototyping
M 7/13	• Mechanics and dynamics • Special dynamics: feedback loops, emergence and intentionality
Th 7/16	• Games and art
M 7/20	• Decision-making, types of decisions • Flow theory
Th 7/23	• Kinds of fun • Player types
M 7/27	• Dramatic elements in games
Th 7/30	• Nonlinear storytelling
M 8/3	• Game design process in detail • Intro to the Design Project for this course
Th 8/6	• Solo testing techniques • Design Project: solo testing
M 8/10	• Designer testing techniques, critical analysis • Design Project: designer testing

Date	Topics Covered
Th 8/13	• Player testing techniques • Design Project: player testing
M 8/17	• Blindtesting techniques • Design Project: Blindtesting
Th 8/20	• Game balance techniques • Design Project: balancing
M 8/24	• User Interface design • Differences between digital and non-digital UI
Th 8/27	• Design Project: User Interface iteration
M 8/31	• Design Project: final materials and presentation • Critical analysis of design projects
Th 9/3	• Course summary • Next steps

Level 1: Overview / What is a Game?

By ai864

Welcome to Game Design Concepts! I am Ian Schreiber, and I will be your guide through this whole experiment. I've heard a lot of excitement throughout all of the registration process these last few months, and be assured that I am just as excited (and intimidated) at this whole process as anyone else. So let me say that I appreciate your time, and will do my best to make the time you spend on this worthwhile.

Course Announcements

Before we begin, I'd like to get a few quick administrative things out of the way:

- Registration. As I'm writing this, there is a large backlog of registrations sent in at the last minute. So, if you sent an email and have not yet received a reply, check your inbox in the next day or two. If you still haven't received a response by Wednesday, it means I did not receive your email and you should find your previous one and forward it. Double-check the email address: gamedesignconcepts@yahoo.com
- On that subject, keep in mind there are over a thousand people actively participating here. I value all of your feedback and contributions, but if you send an email directly to me, understand that I may receive a lot and it may take some time to get back to you.
- I've set up two resources for this course, a wiki and a blog.
- The wiki is at http://gamedesignconcepts.pbworks.comand is intended for two things: as a resource for group collaboration (for those of you who are taking this course with friends), and as an area for people to post translations of this blog into other languages (as some of you have offered). If you think of other uses, feel free! Right now it is a closed wiki and requires an email address and password. If you registered, expect to receive an email in the next few days giving you login information.
- The discussion board is at http://gamedesignconcepts.aceboard.com and is the primary place for interactive discussion. I've created separate discussion areas by interest group (such as: an area for college students, another for professional educators, another for professional video game designers, and so on). I will create forums by geographic region soon, to allow you to find others in your area and arrange to meet in person, if you'd like. This is also where you'll be able to post the work you do for this course, and give and receive peer review (these will become available as they are assigned). Lastly, there is a forum at the top called Meta Discussion, which is discussion for the course itself — what is working and what is not, in terms of using blog, wiki, forum and so on in order to communicate. **You will have to create an account and then wait for the moderator to add you. This process may take a couple of days, so please be patient.**
- If you twitter, use the tag #GDCU for any course-related tweets.
- If you have something to say about the course content itself, feel free to leave it as a comment here on this blog.

And with all of that out of the way, let's talk about game design!

Course Overview

Most fields of study have been around for thousands of years. Game design has been studied for not much more than ten. We do not have a vast body of work to draw upon, compared to those in most other arts and sciences.

On the other hand, we are lucky. Within the past few years, we have finally reached what I see as a critical mass of conceptual writing, formal analysis, and theoretical and practical understanding to be able to fill a college curriculum… or at least, in this case, a ten-week course.

Okay, that isn't entirely fair. There is actually a huge body of material in the field of game design, and many books (with more being released at an alarming rate). But the vast majority of it is either useless, or it is such dense reading that no one in the field bothers to read it. The readings we'll have in this course are those that have, for whatever reason, pervaded the industry; many professional designers are already familiar with them.

This course will be divided, roughly, into two parts. The first half of the course will focus on the theories and concepts of game design. We will learn what a game is, how to break the concept of a game down into its component parts, and what makes one game better or worse than another. In the second half of the course, the main focus is the practical aspect of how to create a good game out of nothing, and the processes that are involved in creating your own games. Throughout all of the course, there will be a number of opportunities to make your own games (all non-digital, no computer programming required), so that you can see how the theory actually works in practice.

What is a game?

Those of you who have read a little into the *Challenges* text may think this is obvious. My preferred definition is a **play activity** with **rules** that involves **conflict**. But the question "what is a game?" is actually more complicated than that:

- For one thing, that's my definition. Sure, it was adopted by the IGDA Education SIG (mostly because no one argued with me about it). There are many other definitions that disagree with mine. Many of those other definitions were proposed by people with more game design experience than me. So, you can't take this definition (or anything else) for granted, just because Ian Says So.
- For another, that definition tells us nothing about how to *design* games, so we'll be talking about what a game is in terms of its component parts: rules, resources, actions, story, and so on. I call these things "formal elements" of games, for reasons that will be discussed later.

Also, it's important to make distinctions between different games. Consider the game of ***Three to Fifteen***. Most of you have probably never heard of or played this game. It has a very simple set of rules:

- Players: 2
- Objective: to collect a set of exactly three numbers that add up to 15.
- Setup: start by writing the numbers 1 through 9 on a sheet of paper. Choose a player to go first.
- Progression of Play: on your turn, choose a number that has not been chosen *by either player*. You now control that number. Cross it off the list of numbers, and write the number on your side of the paper to show that it is now yours.
- Resolution: if either player collects a set of exactly three numbers that add up to exactly 15, the game ends, and that player wins. If all nine numbers are collected and neither player has won, the game is a draw.

Go ahead and play this game, either against yourself or against another player. Do you recognize it now?

The numbers 1 through 9 can be arranged in a 3×3 grid known as a "magic square" where every row, column and diagonal adds up to exactly 15:

6
7
2
1
5
9
8
3
4

Now you may recognize it. It is the game of ***Tic-Tac-Toe*** (or ***Noughts and Crosses*** or several other names, depending on where you live). So, is *Tic-Tac-Toe* the same game as *Three-to-Fifteen*, or are they different games? (The answer is, it depends on what you mean… which is why it is important to define what a "game" is!)

Working towards a Critical Vocabulary

When I say "vocabulary" what I mean is, a set of words that allows us to talk about games. The word "critical" in this case does not mean that we are *being* critical (i.e. finding fault with a game), but rather that we are able to analyze games *critically* (as in, being able to analyze them carefully by considering all of their parts and how they fit together, and looking at both the good and the bad).

Vocabulary might not be as fascinating as that game you want to design with robot laser ninjas, but it is important, because it gives us the means to talk about games. Otherwise we'll be stuck gesturing and grunting, and it becomes very hard to learn anything if we can't communicate.

One of the most common ways to talk about games is to describe them in terms of other games. "*It's like Grand Theft Auto meets The Sims meets World of Warcraft.*" But this has two limitations. First, if I haven't played *World of Warcraft*, then I won't know what you mean; it requires us to both have played the same games. Second, and more importantly, it does not cover the case of a game that is very different. How would you describe *Katamari Damacy* in terms of other games?

Another option, often chosen by those who write textbooks on game design, is to invent terminology as needed and then use it consistently within the text. I could do this, and we could at least communicate with each other about fundamental game design concepts. The problem here is what happens after this course is over; the jargon from this course would become useless when you were talking to anyone else. I cannot force or mandate that the game industry adopt my terminology.

One might wonder, if having the words to discuss games is such an important thing, why hasn't it been done already? Why hasn't the game industry settled on a list of terms? The answer is that it is doing so, but it is a slow process. We'll see plenty of this emerging in the readings, and it is a theme we will return to many times during the first half of this course.

Games and Play

There are many kinds of play: tossing a ball around, playing make-believe, and of course *games*. So, you can think of games as one type of play.

Games are made of many parts, including the rules, story, physical components, and so on. Play is just one aspect of games. Therefore, you can also think of play as one part of games.

How can two things both be *a subset* the other? It seems like a paradox, and it's something you are welcome to think about on your own. For our purposes, it doesn't matter — the point here is that *games* and *play* are concepts that are related.

So, what is a game, anyway?

You might have noticed I never answered the earlier question of what a game is. This is because the concept is very difficult to define, at least in a way that doesn't either leave things out that are obviously games (so the definition is too narrow), or accept things that are clearly not games (making the definition too broad)… or sometimes both.

Here are some definitions from various sources:

- A game has "ends and means": an objective, an outcome, and a set of rules to get there. (David Parlett)
- A game is an activity involving player decisions, seeking objectives within a "limiting context" [i.e. rules]. (Clark C. Abt)

- A game has six properties: it is "free" (playing is optional and not obligatory), "separate" (fixed in space and time, in advance), has an uncertain outcome, is "unproductive" (in the sense of creating neither goods nor wealth — note that wagering *transfers*wealth between players but does not create it), is governed by rules, and is "make believe" (accompanied by an awareness that the game is not Real Life, but is some kind of shared separate "reality"). (Roger Callois)
- A game is a "voluntary effort to overcome unnecessary obstacles." This is a favorite among my classroom students. It sounds a bit different, but includes a lot of concepts of former definitions: it is voluntary, it has goals and rules. The bit about "unnecessary obstacles" implies an inefficiency caused by the rules on purpose — for example, if the object of *Tic Tac Toe* is to get three symbols across, down or diagonally, the easiest way to do that is to simply write three symbols in a row on your first turn while keeping the paper away from your opponent. But you don't do that, because the rules get in the way… and it is from those rules that the play emerges. (Bernard Suits)
- Games have four properties. They are a "closed, formal system" (this is a fancy way of saying that they have rules; "formal" in this case means that it can be defined, not that it involves wearing a suit and tie); they involve interaction; they involve conflict; and they offer safety… at least compared to what they represent (for example, American Football is certainly not what one would call perfectly safe — injuries are common — but as a game it is an abstract representation of warfare, and it is certainly more safe than being a soldier in the middle of combat). (Chris Crawford)
- Games are a "form of art in which the participants, termed Players, make decisions in order to manage resources through game tokens in the pursuit of a goal." This definition includes a number of concepts not seen in earlier definitions: games are art, they involve decisions and resource management, and they have "tokens" (objects within the game). There is also the familiar concept of goals. (Greg Costikyan)
- Games are a "system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome" ("quantifiable" here just means, for example, that there is a concept of "winning" and "losing"). This definition is from the book *Rules of Play*by Katie Salen and Eric Zimmerman. That book also lists the other definitions given above, and I thank the authors for putting them all in one place for easy reference.

By examining these definitions, we now have a starting point for discussing games. Some of the elements mentioned that seem to be common to many (if not all) games include:

- Games are an **activity**.
- Games have **rules.**
- Games have **conflict**.
- Games have **goals**.
- Games involve **decision making**.
- Games are **artificial**, they are **safe**, and they are **outside ordinary life**. This is sometimes referred to as the players stepping into the "Magic Circle" or sharing a "lusory attitude".
- Games involve **no material gain** on the part of the players.

- Games are **voluntary**. If you are held at gunpoint and forced into an activity that would normally be considered a game, some would say that it is no longer a game *for you*. (Something to think about: if you accept this, then an activity that is voluntary for some players and compulsory for others may or may not be a game… depending on whose point of view you are looking at.)
- Games have an **uncertain outcome**.
- Games are a **representation** or **simulation** of something real, but they are themselves **make believe**.
- Games are **inefficient**. The rules impose obstacles that prevent the player from reaching their goal through the most efficient means.
- Games have **systems**. Usually, it is a **closed** system, meaning that resources and information do not flow between the game and the outside world.
- Games are a form of **art**.

Weaknesses of Definitions

Which of the earlier definitions is correct?

None of them are perfect. If you try to come up with your own definition, it will likely be imperfect as well. Here are a few common edge cases that commonly cause problems with definitions:

- **Puzzles**, such as crossword puzzles, Sudoku, Rubik's Cube, or logic puzzles. Are these games? It depends on the definition. Salen & Zimmerman say they are a *subset* of games where there is a set of correct answers. Costikyan says they are not games, although they may be contained within a game.
- **Role-playing games**, such as *Dungeons & Dragons*. They have the word "game" right in the title, yet they are often not considered games (for example, because they often have no final outcome or resolution, no winning or losing).
- **Choose-your-own-adventure books**. These are not generally thought of as games; you say you are "reading" a book, not "playing" it. And yet, it fits most of the criteria for most definitions of a game. To make things even more confusing, if you take one of these books, add a tear-out "character sheet" with some numeric stats, include "skill checks" on some pages where you roll a die against a stat, and call it an "adventure module" instead of a "choose-your-own-adventure book," we would now call it a game!
- **Stories**. Are games stories? On the one hand, most stories are linear, while games tend to be more dynamic. On the other hand, most games have some kind of story or narrative in them; we even have professional story writers that work on multi-million-dollar video game projects. And even beyond that, a player can tell a story about *their* game experience ("let me tell you about this Chess game I played last night, it was awesome"). For now, keep in mind that the concepts of **story** and **game** are related in many ways, and we'll explore this more thoroughly later in the course.

Let's Make a Game

You might be wondering how all of this is going to help you make games. It isn't, directly… but we need to at least take some steps towards a shared vocabulary so that we can talk about games in a meaningful way.

Here's a thing about games. I hear a lot from students that they're afraid they won't be able to make a game. They don't have the creativity, or the skills, or whatever. This is nonsense, and it is time to get that out of our systems now.

If you have never made a game before, it is time to get over your fear. You are going to make a game now. Take out a pencil and paper (or load up a drawing program like Microsoft Paint). This will take all of 15 minutes and it will be fun and painless, I promise.

I mean it, get ready. Okay?

We are going to make what is referred to as a race-to-the-end board game. You have probably played a lot of these; the object is to get your token from one area of a game board to another. Common examples include *Candyland*, *Chutes & Ladders*, and *Parcheesi*. They are the easiest kind of game to design, and you're going to make one now.

First, draw some kind of path. It can be straight or curved. All it takes is drawing a line. Now divide the path into spaces. You have now completed the first step out of four. See how easy this is?

Second, come up with a theme or objective. The players need to get from one end of the path to the other; why? You are either **running towards** something or **running away from** something. What are the players represented as in the game? What is their goal? In the design of many games, it is often helpful to start by asking what the objective is, and a lot of rules will fall into place just from that. You should be able to come up with something (even if it is extremely silly) in just a few minutes. You're now half way done!

Third, you need a set of rules to allow the players to travel from space to space. How do you move? The simplest way, which you're probably familiar with, is to roll a die on your turn and move that many spaces forward. You also need to decide exactly how the game ends: do you have to land on the final space by exact count, or does the game end as soon as a player reaches or passes it?

You now have something that has all the elements of a game, although it is missing one element common to many games: conflict. Games tend to be more interesting if you can affect your opponents, either by helping them or harming them. Think of ways to interact with your opponents. Does something happen when you land on the same space as them? Are there spaces you land on that let you do things to your opponents, such as move them forward or back? Can you move your opponents through other means on your turn (such as if you roll a certain result on the die)? Add at least one way to modify the standing of your opponents when it is your turn.

Congratulations! You have now made a game. It may not be a particularly *good* game (as that is something we will cover later in this course), but it is a functional game that can be played, and you made it in just a few minutes, with no tools other than a simple pencil and paper.

Credit for developing this exercise goes to my friend and co-author, Brenda Brathwaite, who noticed that there is this invisible barrier between a lot of people and game design, and created this as a way to get her students over their initial fear that they might not be able to design anything.

Lessons Learned

If you take away nothing else from this little activity, realize that you can have a playable game in minutes. It does not take programming skill. It does not require a great deal of creativity. It does not require lots of money, resources, or special materials. It does not take months or years of time. Making a *good* game may require some or all of these things, but the process of just starting out with a simple idea is something that can be done in a very short period of time with nothing more than a few slips of paper.

Remember this as we move forward in this course. When we talk about iteration and rapid prototyping, many people are afraid to commit to a design, to actually *build* their idea. They are afraid it will take too long, or that the idea will not turn out to be as good as it seems in their head. Part of the process involves killing weak ideas and making them stronger, by actually making and playing your game. The faster you can have something up and running, and the more times that you can play it, the better a game you can make. If it takes you more than a few minutes to make your first prototype of a new idea, it is taking too long.

Homeplay

Some classes assign "homework problems." I'm not sure what is less fun: the concept of work at home, or having problems. So, I call everything a "homeplay" because I want these to be fun and interesting.

Before this Thursday, read the following:

- *Challenges for Game Designers*, Chapter 1 (Basics). This is just a short introduction to the text.
- *I Have No Words and I Must Design*, by Greg Costikyan. To me (and I'm sure others will disagree), this essay is the turning point when game design started to become its own field of study. Since it all started here, for me at least, I think it only fitting to introduce it at the start of this course. (There is a newer version here[PDF] if you are interested, but I prefer the original for its historical importance.)
- *Understanding Games 1*, *Understanding Games 2*, *Understanding Games 3*, *Understanding Games 4*. These are not readings, but playings. They are a series of short Flash games that attempt to explain some basic concepts of games in the form of a game.

The name is a reference to *Understanding Comics*, a comic book that explains about comic books. Each one takes about five minutes.

Level 2: Game Design / Iteration and Rapid Prototyping

By ai864

Last time we asked the question: what is a game? Today, we look into a related question: what, exactly, is game design? Last time, we made a simple game. This time, we will look into the process of how games are made in general. While it is possible to make a race-to-the-end board game in 15 minutes, you will need to take a little longer if you are looking to make the next *Settlers of Catan* or *World of Warcraft*.

Course Announcements

Some administrative things that have come up since Monday:

- First, I would like to apologize to those who are registered for the misbehavior of the wiki. It was sending out hourly announcements of updates… and there were a lot of updates! I have attempted to turn off these updates, so you should hopefully not receive any more "wiki-spam" but if you do, you can shut it off manually by going to your own settings and changing notifications to "Never."
- As of 5am GMT this morning, the discussion forums are set up and operational. I look forward to seeing some really great discussion. There were quite a few spambots that tried to register, so I had to check every forum account against course registrations. If you got an email that your account was rejected, it just means I was unable to match it up to a course registration; please try again. If you have not yet created a forum account, you can make it easiest for me by using the same email on the forums that you used to register for this course… and if you're unwilling to do that, at least put some kind of identifying information in there (like your name and location) so I can find you in the list. Thank you.
- Lastly, to those of you who sent in a registration email after the course started (that is, if your email was timestamped after Monday, noon GMT), I apologize for not being able to add you after the fact. Registration emails have taken a lot of my time prior to the start of the course, and if I accept late registrations I will not have the time to do other things for the course. Whether you are registered or not, this blog is free and open to the public (as is the Twitter feed), and I hope you do still follow along and get a lot out of the experience.

Game Design

We will use the word "design" a lot in this course, and unfortunately it is a term that is a bit overused, so I will clarify what I mean here. As it says in *Challenges*, game design is the creation of the rules and content of a game. It **does not** involve programming, art or animation, or marketing, or any of the other myriad tasks required to make a game. All of these tasks collectively can be called "game development" and game design is one part of development.

Unfortunately, I have seen the term "design" used (particularly in some college curricula) to refer to all aspects of development. When used in the video game industry (or the board game industry), “game design” has a very specific meaning, and that is the meaning that we will use for this course.

Multiple Types of Game Design

As mentioned in *Challenges*, there are many tasks associated with game design: system design, level design, content design, user interface design, world building, and story writing. You could fill several 10-week courses with any one of these, so this summer course will *not* be a full treatment of the entire range of game design. We will touch lightly on UI, story writing and content when relevant, but the majority of this course focuses on **system design** (also sometimes called “systems design” or “core systems design”).

System design is about defining the basic rules of the game. What are the pieces? What can you control? What actions can you take on your turn (if there are “turns” at all)? What happens when you take each action, and how does it affect the game state? In general, system design is the creation of three things:

- Rules for setup. How does the game begin?
- Rules for progression of play. Once the game begins, what can the players do, and what happens when they do things?
- Rules for resolution. What, if anything, causes the game to end? If the game has an outcome (such as winning or losing), how is that outcome determined?

If you look back at *Three-to-Fifteen* from Monday, you’ll notice those very simple rules contain all of these parts. The creation of those rules is system design, and that is what we will be spending most of our time with over this summer.

What is a Game Designer?

As you may have noticed, game design is an incredibly broad field. Those of us who are professional designers sometimes have trouble explaining what we do to our families and friends. Part of the reason for this is that we do so many things. Here are some analogies I’ve seen when trying to explain what it is like to be a game designer:

- Game designers are **artists**. The term “art” is just as difficult to define as the word “game”… but if games can be a form of art (as we saw in Costikyan’s definition, at least), then designers would be artists.
- Game designers are **architects**. Architects do not build physical structures; they create blueprints. Video game designers also create “blueprints” which are referred to as “design docs.” Board game designers create “blueprints” as well — in the form of prototypes — which are then mass-produced by publishers.
- Game designers are **party hosts**. As designers, we invite players into our space and try our best to show them a good time.
- Game designers are **research scientists**. As I will touch on later today, we create games in a manner that is very close to the scientific method.

- Game designers are **gods**. We create worlds, and we create the physical rules that govern those worlds.
- Game designers are **lawyers**. We create a set of rules that others must follow.
- Game designers are **educators**. As we will see later when we start reading *Theory of Fun*, entertainment and education are strongly linked, and games are (at least sometimes) fun because they involve learning new skills.

If game design is all these things, where would it fit in a college curriculum? It could be justified in the school of education, or art, or architecture, or theology, or recreation management, or law, or engineering, or applied sciences, or half a dozen other things.

Is a game designer all of these things? None of them? It is open for discussion, but I think that game design has *elements* of many other fields, but it is still its own field. And you can see just how broad the field is! As the field of game design advances, we may see a day where game designers are so specialized that "game design" will be like the field of "science" — students will need to pick a specialty (Chemistry, Biology, Physics, etc.) rather than just "majoring in Science."

Speaking of Science…

How is a game designed? There are many methods.

Historically, the first design methodology was known as the **waterfall** method: first you design the entire game on paper, then you implement it (using programming in a video game, or creating the board and pieces for a non-digital game), then you test it to make sure the rules work properly, add some graphical polish to make it look nice, and then you ship it.

Waterfall is so named because, like water in a waterfall, you can only move in one direction. If you're busy making the final art for the game and it occurs to you that one of the rules needs to change, too bad — the methodology does not include a way to go back to the design step once you are done.

At some point, someone figured out that it might be a good idea to at least have the *option* of going back and fixing things in earlier steps, and created what is sometimes known as the **iterative** approach. As with waterfall, you first design the game, then implement it, and then make sure it works. But after this you add an extra step of evaluating the game. Play it, decide what is good and what needs to change. And then, make a decision: are you done, or should you go back to the design step and make some changes? If you decide the game is good enough, then that is that. But if you identify some changes, you now go back to the design step, find ways to address the identified problems, implement those changes, and then evaluate again. Continue doing this until the game is ready.

If this sounds familiar, it is because this is more or less the Scientific Method:

1. Make an observation. ("My experience in playing/making games has shown me that certain types of mechanics are fun.")
2. Make a hypothesis. ("I think that this particular set of rules I am writing will make a fun game.")
3. Create an experiment to prove or disprove the hypothesis. ("Let's organize a playtest of this game and see if it is fun or not.")
4. Perform the experiment. ("Let's play!")
5. Interpret the results of the experiment, forming a new set of observations. Go back to the first step.

With non-digital (card and board) games, this process works fine, because it can be done quickly. With video games, there is still one problem: implementation (i.e. programming and debugging) is expensive and takes a long time. If it takes 18 months to code the game the first time and you only have two years, you will not get a lot of time to playtest and modify the game.

In general, **the more times you iterate, the better your final game will be.**

Therefore, any game design process should involve iterating (that is, going through an entire cycle of designing, implementing and evaluating) as much as possible, and anything you can do that lets you iterate faster will usually lead to a better game in the end. Because of this, video game designers will often prototype on paper first, and then only get the programmers involved when they are confident that the core rules are fun. We call this **rapid prototyping**.

Iteration and Risk

Games have many kinds of risk associated with them. There is **design risk**, the risk that the game will not be fun and people won't like it. There is **implementation risk**, the possibility that the development team will not be able to build the game at all, even if the rules are solid. There is **market risk**, the chance that the game will be wonderful and no one will buy it anyway. And so on.

The purpose of iteration is to lower design risk. The more times you iterate, the more you can be certain that the rules of your game are effective.

This all comes down to one important point: the greater the design risk of your game (that is, if your rules are untested and unproven), the more you need iteration. An iterative method is not as critical for games where the mechanics are largely lifted from another successful game; sequels and expansion sets to popular games are examples of situations where a Waterfall approach may work fine.

That said, most game designers have aspirations of making games that are new, creative, and innovative.

Why This Course is Non-Digital…

Some of you would rather make board games anyway, so you don't care how video games are made. But for those of you who would love to make video games, you may have wondered why we will be spending so much time making board and card games in this course. Now you know: it is because iteration is faster and cheaper with cardboard. Remember from Monday: you can make a board game in 15 minutes. Coding that game will take significantly longer. When possible, prototype on paper first, because a 15-minute paper prototype and an hour-long playtest session can save you months of programming work.

Later in this course, we will discuss in detail methods of paper prototyping, both for traditional board games and also for various types of video games.

There is another reason why we will concentrate primarily on non-digital games this summer, particularly board and card games. This is a course in **systems design**, that is, creating the rules of the game. In board games, the rules are laid bare. There may be some physical components, sure, but the play experience is almost entirely determined by the rules and the player interactions. If the rules are not compelling, the game will not be fun, so working in this medium makes a clear connection between the rules and the player experience.

This is not as true in video games. Many video games have impressive technology (such as realistic physics engines) and graphics and sound, which can obscure the fact that the gameplay is stale. Video games also take much longer to make (due to programming and art/audio asset creation), making them an impractical choice for a ten-week course.

The connection between rules and player experience is also muddied in tabletop role-playing games. I realize that many of you have expressed an interest primarily in RPG design, so this may seem strange to you. However, keep in mind that an RPG is essentially a collaborative story-telling exercise (with a rules system in place to set boundaries for what can and can't happen). As such, a wonderful system can be ruined by players who have poor story-telling and improv skills, and a weak system can be salvaged by skillful players. As such, we will stay away from these game genres, at least in the early stages.

Trying it out

Take that 15-minute game you made last time, and play it, if you haven't already. As you are playing, ask yourself: is this more fun or less fun than playing your favorite published games? Why? What could you change about your game to make it better? You do not have to play the game to completion, but only for as long as it takes you to get the overall feeling of what it is like to play.

Then, after playing once, make at least one change. Maybe you'll change the rules for movement, or add a new way for players to interact. Maybe you'll change some of the spaces on the board. Whatever you do, for whatever reason, make a change and then play again. Note the differences. Has the change made the game better, or worse? Has this one change made you think of additional changes you could make? If the game got worse, would you just change the rule back, or would you change it again in a different way?

We will be looking at the playtest process in detail later in this course. For now, I just want everyone to get over that fear: "what if I play my game and it sucks?" With the game you designed on Monday, the odds are very high that your game *does* suck (seriously, did you expect to make the next *Gears of War* in 15 minutes?). This does not make you a "bad designer" by any means — but it should make it clear that the more time you put into a game and the more iterations you make, the better it gets.

Lessons Learned

The one big takeaway from today is that the more you iterate on a game, the better it becomes. Great designers do not design great games. They usually design *really bad* games, and then they iterate on them until the games *become* great.

This has two corollaries:

- You want to have a *playable* prototype of your game as early in development as possible. The faster you can playtest your ideas, the more time you have to make changes.
- Given equal amounts of time, a shorter, simpler game will give a better experience than a longer, complicated game. A game that takes ten hours to play to completion will give you fewer iterations than a game that can be played in five minutes. When we start on the Design Project later in this course, keep this in mind.

Homeplay

Before next Monday, read the following. I will be referencing these in Monday's content when we talk about the formal elements of games:

- *Challenges for Game Designers*, Chapter 2 (Atoms). This will act as a bridge between last Monday when we talked about a critical vocabulary, and next Monday when we will start breaking down the concept of a "game" into its component parts.
- *Formal Abstract Design Tools*, by Doug Church. This article builds on Costikyan's *I Have No Words*, offering some additional tools by which we can analyze and design games. While he does use many examples from video games, think about how the core concepts in the article can apply to other kinds of games as well.

Level 3: Formal Elements of Games

By ai864

Today marks the last day that we continue in building a critical vocabulary from which to discuss games; this Thursday we will dive right in to the game design process. Today I want the last pieces to fall into place: we need a way to dissect and analyze a game by discussing its component parts and how they all fit together. This can be useful when discussing other people's games (it would be nice if, for example, more professional game reviews could do this properly), but it is also useful in designing our own games. After all, how can you design a game if you don't know how all the different parts fit together?

Course Announcements

As usual, there are a few things I'd like to announce and clarify:

- I'm happy to announce that the course wiki is now open to the public (read-only access). This wiki is pretty much entirely run by the participants who registered for this course. Among other things, the blog posts here have already been translated into five different languages. I am impressed and humbled at the level of participation going on there, and encourage casual viewers to stop by and check it out.
- I noticed some confusion on this so I would like to clarify: for readings in the *Challenges* text, you do not have to actually do all of the challenges at the end of the chapter. You certainly can if you want, but most chapters have five long challenges and ten short ones, and I would call that an extreme workload for a class of this pace. Repeat, **you do not have to do all of the challenges** (except where expressly noted on this blog).

A note on the reading for today

One of the readings for today was Doug Church's *Formal Abstract Design Tools*. I want to mention a few things about this. First, he mentions three aspects of games that are worth putting in our design toolbox:

- **Player intention** is defined as the ability of the player to devise and carry out their own plans and goals. We will come back to this later on in this course, but for now just realize that it can be important in many games to allow the player to form a plan of action.
- **Perceivable consequence** is defined in the reading as a clear reaction of the game to the player's actions. Clarity is important here: if the game reacts but you don't know how the game state has changed, then you may have difficulty linking your actions to the consequences of those actions. I'll point out that "perceivable consequence" is known by a more common name: **feedback**.
- **Story** is the narrative thread of the game. Note that a game can contain two different types of story: the "embedded" story (created by the designer) and the "emergent" story (created by players). Emergent story happens, for example, when you tell your friends about a recent game you played and what happened to you during the play: "I had taken over all of Africa, but I just couldn't keep the Blue player out of Zaire." Embedded story is what we normally think of as the "narrative" of the game: "You are playing a brave

knight venturing into the castle of an evil wizard.” Doug’s point is that *embedded* story competes with intention and consequence — that is, the more the game is “on rails”, the less the player can affect the outcome. When Costikyan said in “I Have No Words” that games are not stories, Doug provides what I think is a better way of saying what Costikyan meant.

Here is an example of why player intention and perceivable consequence are important. Consider this situation: you are playing a first-person shooter game. You walk up to a wall that has a switch on it. You flip the switch. Nothing happens. Well, actually something *did* happen, but the game gives you no indication of *what* happened. Maybe a door somewhere else in the level opened. Maybe you just unleashed a bunch of monsters into the area, and you’ll run into them as soon as you exit the current room. Maybe there are a series of switches, and they all have to be in *exactly the right pattern* of on and off (or they have to be triggered in the right order) in order to open up the path to the level exit. But you have no way of knowing, and so you feel frustrated that you must now do a thorough search of everywhere you’ve already been… just to see if the switch did anything.

How could you fix this? Add better feedback. One way would be to provide a map to the player, and show them a location on the map when the switch was pulled. Or, show a brief cut scene that shows a door opening somewhere. I’m sure you can think of other methods as well.

On another subject, Doug also included an interesting note at the end of the article about how he values beta testing, and half of his readers found the first two pages slow, so start at page 3 if you’re in that half. This would be an example of **iteration** in the design of this essay, of exactly the sort we talked about.

Now, I’m sure this note was partly in jest, but let’s take it at face value. There’s a slight problem with this fix: you don’t see the note until you’ve already read all of the way through the article, and it’s too late to do anything about it. If Doug were to iterate on his design a second time, what would you suggest he do? (I’ve heard many suggestions from my students in the past.)

Qualities of Games

It was rightly pointed out in the comments of this blog that on the first day of this course, I contradicted myself: I insisted that a critical vocabulary was important, and then I went on to say that completely defining the word “game” is impossible. Let’s reconcile this apparent paradox.

Take a quick look at the definitions listed on the first day. Separate out all of the qualities listed from each definition that may apply to games. We see some recurring themes: games have rules, conflict, goals, decision-making, and an uncertain outcome. Games are activities, they are artificial / safe / outside ordinary life, they are voluntary, they contain elements of make-believe / representation / simulation, they are inefficient, they are art, and they are closed systems. Think for a moment about what other things are common to all (or most) games. This provides a starting point for us to identify individual game elements.

I refer to these as “formal elements” again, not because they have anything to do with wearing a suit and tie, but because they are “formal” in the mathematical and scientific sense: something that can be explicitly defined. *Challenges* refers to them as “atoms” — in the sense that these are the smallest parts of a game that can be isolated and studied individually.

What are atomic elements of games?

This depends on who you ask. I have seen several schemes of classification. Like the definition of “game,” none is perfect, but by looking at all of them we can see some emerging themes that can shed light on the kinds of things that we need to create as game designers if we are to make games.

What follows are some parts of games, and some of the things designers may consider when looking at these atoms.

Players

How many players does the game support? Must it be an exact number (4 players only), or a variable number (2 to 5 players)? Can players enter or leave during play? How does this affect play?

What is the relationship between players: are there teams, or individuals? Can teams be uneven? Here are some example player structures; this is by no means a complete list:

- Solitaire (1 player vs. the game system). Examples include the card game *Klondike* (sometimes just called “Solitaire”) and the video game *Minesweeper*.
- Head-to-head (1 player vs. 1 player). *Chess* and *Go* are classic examples.
- “PvE” (multiple players vs. the game system). This is common in MMOs like *World of Warcraft*. Some purely-cooperative board games exist too, such as *Knizia's Lord of the Rings*, *Arkham Horror*, and *Pandemic*.
- One-against-many (1 player vs. multiple players). The board game *Scotland Yard* is a great example of this; it pits a single player as Mr. X against a team of detectives.
- Free-for-all (1 player vs. 1 player vs. 1 player vs. …). Perhaps the most common player structure for multi-player games, this can be found everywhere, from board games like *Monopoly* to “multiplayer deathmatch” play in most first-person shooter video games.
- Separate individuals against the system (1 player vs. a series of other players). The casino game *Blackjack* is an example, where the “House” is playing as a single player against several other players, but those other players are not affecting each other much and do not really help or hinder or play against each other.
- Team competition (multiple players vs. multiple players [vs. multiple players...]). This is also a common structure, finding its way into most team sports, card games like *Bridge* and *Spades*, team-based online games like “Capture the Flag” modes from first-person shooters, and numerous other games.

- Predator-Prey. Players form a (real or virtual) circle. Everyone's goal is to attack the player on their left, and defend themselves from the player on their right. The college game *Assassination* and the trading-card game *Vampire: the Eternal Struggle* both use this structure.
- Five-pointed Star. I first saw this in a five-player *Magic: the Gathering* variant. The goal is to eliminate both of the players who are *not* on either side of you.

Objectives (goals)

What is the object of the game? What are the players trying to do? This is often one of the first things you can ask yourself when designing a game, if you're stuck and don't know where to begin. Once you know the objective, many of the other formal elements will seem to define themselves for you. Some common objectives (again, this is not a complete list):

- Capture/destroy. Eliminate all of your opponent's pieces from the game. *Chess* and *Stratego* are some well-known examples where you must eliminate the opposing forces to win.
- Territorial control. The focus is not necessarily on destroying the opponent, but on controlling certain areas of the board. *RISK* and *Diplomacy* are examples.
- Collection. The card game *Rummy* and its variants involve collecting sets of cards to win. *Bohnanza* involves collecting sets of beans. Many platformer video games (such as the *Spyro* series) included levels where you had to collect a certain number of objects scattered throughout the level.
- Solve. The board game *Clue* (or *Cluedo*, depending on where you live) is an example of a game where the objective is to solve a puzzle. Lesser-known (but more interesting) examples are *Castle of Magic* and *Sleuth*.
- Chase/race/escape. Generally, anything where you are running towards or away from something; the playground game *Tag* and the video game *Super Mario Bros.* are examples.
- Spatial alignment. A number of games involve positioning of elements as an objective, including the non-digital games *Tic-Tac-Toe* and *Pente* and the video game *Tetris*.
- Build. The opposite of "destroy" — your goal is to advance your character(s) or build your resources to a certain point. *The Sims* has strong elements of this; the board game *Settlers of Catan* is an example also.
- Negation of another goal. Some games end when one player performs an act that is forbiden by the rules, and that player loses. Examples are the physical dexterity games *Twister* and *Jenga*.

Rules (mechanics)

As mentioned last week, there are three categories of rules: setup (things you do once at the beginning of the game), progression of play (what happens during the game), and resolution (what conditions cause the game to end, and how is an outcome determined based on the game state).

Some rules are automatic: they are triggered at a certain point in the game without player choices or interaction ("Draw a card at the start of your turn" or "The bonus timer decreases by 100 points every second"). Other rules define the choices or actions that the players can take in the game, and the effects of those actions on the game state.

Let's dig deeper. Salen & Zimmerman's *Rules of Play* classifies three types of rules, which they call operational, constituative, and implied (these are not standard terms in the industry, so the concepts are more important than the terminology in this case). To illustrate, let's consider the rules of *Tic-Tac-Toe*:

- Players: 2
- Setup: Draw a 3×3 grid. Choose a player to go first as X. Their opponent is designated O.
- Progression of play: On your turn, mark an empty square with your symbol. Play then passes to your opponent.
- Resolution: If you get 3 of your symbol in a row (orthogonally or diagonally), you win. If the board is filled and there is no winner, it is a draw.

These are what *Rules of Play* calls the "operational" rules. Think for a moment: are these the only rules of the game?

At first glance, it seems so. But what if I'm losing and simply refuse to take another turn? The rules do not explicitly give a time limit, so I could "stall" indefinitely to avoid losing and still be operating within the "rules" as they are typically stated. However, in actual play, a reasonable time limit is implied. This is not part of the formal (operational) rules of the game, but it is still part of what *Rules of Play* calls the "implied" rules. The point here is that there is some kind of unwritten social contract that players make when playing a game, and these are understood even when not stated.

Even within the formal rules there are two layers. The 3×3 board and "X" and "O" symbols are specific to the "flavor" of this game, but you could strip them away. By reframing the squares as the numbers 1 through 9 and turning spatial alignment into a mathematical property, you can get *Three-to-Fifteen*. While *Tic-Tac-Toe* and *Three-to-Fifteen* have different implementations and appearances, the underlying abstract rules are the same. We do not normally think in these abstract terms when we think of "rules" but they are still there, under the surface. *Rules of Play* calls these "constituative" rules.

Is it useful to make the distinction between these three types of rules? I think it is important to be aware of them for two reasons:

- The distinction between "operational" and "constituative" rules helps us understand why one game is fun in relation to other games. The classic arcade game *Gauntlet* has highly similar gameplay to the first-person shooter *DOOM*; the largest difference is the position of the camera. For those of you who play modern board games, a similar statement is that *Puerto Rico* is highly similar to *Race for the Galaxy.* The similarity may not be immediately apparent because the games look so different on the surface, unless you are thinking in terms of game states and rules.

- Many first-person shooters contain a rule where, when a player is killed, they re-appear ("respawn") in a specific known location. Another player can stand near that location and kill anyone that respawns before they have a chance to react. This is known as "spawn-camping" and can be rather annoying to someone on the receiving end of it. Is spawn-camping part of the game (since it is allowed by the rules)? Is it good strategy, or is it cheating? This depends on who you ask, as it is part of the "implied" rules of the game. When two players are operating under different implied rules, you will eventually get one player accusing the other of cheating (or just "being cheap") while the other player will get defensive and say that they're playing by the rules, and there's no reason for them to handicap themselves when they are playing to win. The lesson here is that it is important for the game designer to define as many of these rules as possible, to avoid rules arguments during play.

Resources and resource management

"Resources" is a broad category, and I use it to mean everything that is under control of a single player. Obviously this includes explicit resources (Wood and Wheat in *Settlers of Catan*, health and mana and currency in *World of Warcraft*), but this can also include other things under player control:

- Territory in *RISK*
- Number of questions remaining in *Twenty Questions*
- Objects that can be picked up in video games (weapons, powerups)
- Time (either game time, or real time, or both)
- Known information (as the suspects that you have eliminated in *Clue*)

What kinds of resources do the players control? How are these resources manipulated during play? This is something the game designer must define explicitly.

Game State

Some "resource-like" things are not owned by a single player, but are still part of the game: unowned properties in *Monopoly*, the common cards in *Texas Hold 'Em*. Everything in the game together, including the current player resources and everything else that makes up a snapshot of the game at a single point in time is called the *game state*.

In board games, explicitly defining the game state is not always necessary, but it is sometimes useful to think about. After all, what are rules, but the means by which the game is transformed from one game state to another?

In video games, someone must define the game state, because it includes all of the data that the computer must keep track of. Normally this task falls to a programmer, but if the game designer can explicitly define the entire game state it can greatly aid in the understanding of the game by the programming team.

Information

How much of the game state is visible to each player? Changing the amount of information available to players has a drastic effect on the game, even if all other formal elements are the same. Some examples of information structures in games:

- A few games offer total information, where all players see the complete game state at all times. *Chess* and *Go* are classic board game examples.
- Games can include some information that is private to each individual. Think of *Poker* and other card games where each player has a hand of cards that only they can see.
- One player can have their own privileged information, while other players do not. This is common in one-against-many player structures, like *Scotland Yard*.
- The game itself can contain information that is hidden from all players. Games like *Clue* and *Sleuth* actually have the victory condition that a player discover this hidden information.
- These can be combined. Many "real-time strategy" computer games use what is called "fog of war" where certain sections of the map are concealed to any player that does not have a unit in sight range. Some information is therefore hidden from all players. Beyond that, players cannot see each other's screens, so each player is unaware of what information is and isn't available to their opponents.

Sequencing

In what order do players take their actions? How does play flow from one action to another? Games can work differently depending on the turn structure that is used:

- Some games are purely turn-based: at any given time it is a single player's "turn" on which they may take action. When they are done, it becomes someone else's turn. Most classic board games and turn-based strategy games work this way.
- Other games are turn-based, but with simultaneous play (everyone takes their turn at the same time, often by writing down their actions or playing an action card face-down and then simultaneously revealing). The board game *Diplomacy* works like this. There is also an interesting *Chess* variant where players write down their turns simultaneously and then resolve (two pieces entering the same square on the same turn are both captured) that adds tension to the game.
- Still other games are real-time, where actions are taken as fast as players can take them. Most action-oriented video games fall into this category, but even some non-digital games (such as the card games *Spit* or *Speed*) work this way.
- There are additional variations. For a turn-based game, what order do players take their turns? Taking turns in clockwise order is common. Taking turns in clockwise order and then skipping the first player (to reduce the first-player advantage) is a modification found in many modern board games. I've also seen games where turn order is randomized for each round of turns, or where players pay other resources in the game for the privilege of going first (or last), or where turn order is determined by player standing (player who is currently winning goes first or last).

- Turn-based games can be further modified by the addition of an explicit time limit, or other form of time pressure.

Player Interaction

This is an often-neglected but highly important aspect of games to consider. How do players interact with one another? How can they influence one another? Here are some examples of player interactions

- Direct conflict ("I attack you")
- Negotiation ("If you support me to enter the Black Sea, I'll help you get into Cairo next turn")
- Trading ("I'll give you a Wood in exchange for your Wheat")
- Information sharing ("I looked at that tile last turn and I'm telling you, if you enter it a trap will go off")

Theme (or narrative, backstory, or setting)

These terms do have distinct meanings for people who are professional story writers, but for our purposes they are used interchangeably to mean the parts of the game that do not directly affect gameplay at all.

If it doesn't matter in terms of gameplay, why bother with this at all? There are two main reasons. First, the setting provides an emotional connection to the game. I find it hard to really care about the pawns on my chessboard the way I care about my *Dungeons & Dragons* character. And while this doesn't necessarily make one game "better" than another, it does make it easier for a player to become emotionally invested in the game.

The other reason is that a well-chosen theme can make a game easier to learn and easier to play, because the rules make sense. The piece movement rules in *Chess* have no relation to the theme and must therefore be memorized by someone learning the game. By contrast, the roles in the board game *Puerto Rico* have some relation to their game function: the builder lets you build buildings, the mayor recruits new colonists, the captain ships goods off to the Old World, and so on. It is easy to remember what most actions do in the game, because they have some relation to the theme of the game.

Games as Systems

I'd like to call two things about these formal elements to your attention.

First, if you change even one formal element, it can make for a *very* different game. Each formal element of a game contributes in a deep way to the player experience. When designing a game, give thought to each of these elements, and make sure that each is a deliberate choice.

Second, note that these elements are interrelated, and changing one can affect others. Rules govern changes in Game State. Information can sometimes become a Resource. Sequencing can lead to different kinds of Player Interaction. Changing the number of Players can affect what kinds of Objectives can be defined. And so on.

Because of the interrelated nature of these parts, you can frame any game as a **system**. (One dictionary definition of the word "system" is: a combination of things or parts that form a complex whole.)

In fact, a single game can contain several systems. *World of Warcraft* has a combat system, a quest system, a guild system, a chat system, and so on…

Another property of systems is that it is hard to fully understand or predict them just by defining them; you gain a far deeper understanding by seeing the system in action. Consider the physical system of projectile motion. I can give you a mathematical equation to define the path of a ball being thrown, and you could even predict its behavior… but the whole thing makes a lot more sense if you see someone actually throwing a ball.

Games are like this, too. You can read the rules and define all the formal elements of a game, but to truly understand a game you need to play it. This is why most people do not immediately see the parallel between *Tic-Tac-Toe* and *Three-to-Fifteen* until they have played them.

Critical Analysis of Games

What is a critical analysis, and why do we care?

Critical analysis is not just a game review. We are not concerned with how many out of five stars, or any numbers from 0 to 10, or whether or not a game is "fun" (whatever *that* means), or aiding in the consumer decision of whether or not to buy a game.

Critical analysis does not just mean a list of things that are wrong with the game. The word "critical" in this context does not mean "fault-finding" but rather a thorough and unbiased look at the game.

Critical analysis is useful when discussing or comparing games. You can say "I like the card game *Bang!* because it's fun" but that does not help us as designers to learn *why* it is fun. We must look at the parts of games and how they interact in order to understand how each part relates to the play experience.

Critical analysis is also useful when examining our own works in progress. For a game that you're working on, how do you know what to add or remove to make it better?

There are many ways to critically analyze a game, but I offer a three-step process:

1. Describe the game's formal elements. Do not interpret at this point, simply state what is there.

2. Describe the results of the formal elements when put in motion. How do the different elements interact? What is the play of the game like? Is it effective?
3. Try to understand why the designer chose those elements and not others. Why this particular player structure, and why that set of resources? What would have happened if the designer had chosen differently?

Some questions to ask yourself during a critical analysis at various stages:

- What challenges do the players face? What actions can players take to overcome those challenges?
- How do players affect each other?
- Is the game perceived by the players as fair? (Note that it may or may not *actually* be fair. Perception and reality often differ.)
- Is the game replayable? Are there multiple paths to victory, varied start positions, or optional rules that cause the experience to be different each time?
- What is the game's intended audience? Is the game appropriate for that audience?
- What is the "core" of the game — the one thing you do over and over that represents the main "fun" part?

Lessons Learned

We covered a lot of content today. The main takeaways I offer:

- Games are systems.
- Understanding a game is much easier if you have played it.
- Analyzing a game requires looking at all of the game's working parts, and figuring out how they fit together and how a play experience arises from them.
- Designing a game requires the creation of all of the game's parts. If you haven't defined the formal elements of your game in some way, then you don't really have a game… you just have the seed of an idea. This is fine, but to make it into a *game* you must actually design it.

Homeplay

It was brought to my attention that I have been using the word "homeplay" to refer to the reading, and that reading is not play no matter how I dress it up. This criticism is valid; normally in my classroom courses I use "homeplay" to refer to actual game design assignments and not readings, and I mixed the terms up here. I will make an attempt to avoid this confusion in the future, because I believe that trying to treat learning as an inherently Not-Fun activity (as evidenced by the need to use fancy fun-sounding words to describe it) is damaging to our collective long-term well being. As we will see when we get into flow theory, the reality is actually the opposite.

With that said, here is an activity that I hope you will find fun. It is based off of Challenge 2-5 in the *Challenges* text, with some minor changes just to keep you on your toes.

Here’s how it works. First, choose your difficulty level based on your previous experience with game design. Skiiers may find this familiar:

Here is your challenge:

Most war-themed games have an objective of either territorial control or capture/destroy (as described earlier). For this challenge, you’ll be pushing beyond these traditional boundaries. You should design a non-digital game that includes the following:

The theme must relate to **World War I**. The primary objective of players cannot be territorial control, or capture/destroy.

You cannot use territorial control or capture/destroy as game dynamics. That is, your game is not allowed to contain the concepts of territory or death in any form.

As above, and the players may not engage in direct conflict, only indirect.

I have created three new areas on the forums (one for each difficulty level). Post your game rules in the appropriate forum by **Thursday, July 9, noon GMT**. You are encouraged to post earlier if possible.

Then, after you have posted, **read at least two other posts from your difficulty level** and offer a constructive analysis and critique. If you are at blue-square or black-diamond difficulty, **also read at least two other posts from the difficulty level immediately below yours** and offer the benefit of your experience to those who you could mentor. Try to choose posts that have no responses already, so that everyone can get at least some feedback. Also complete this by Thursday, noon GMT.

A note about research…

Note that you may have to do some actual research to complete this project (even if only looking to Wikipedia for inspiration). This is typical of much game design in the field. Many laypersons imagine game designers as these people that just sit and think at their desk all day, then eventually stand up and proclaim, “*I have this Great Idea for a game! Ninjas… and lasers… in space! Go forth and build it, my army of programmer and art lackeys. I shall sit here now until I come up with another Great Idea, while collecting royalties from my last five ideas.*” This is not even close to reality. A great deal of design is the details: defining the rules, certainly, but also doing research to make sure that the rules fit the constraints and are appropriate for the project.

A note about IP law…

At this point, some of you may be thinking that by posting your game to the forum, you run the risk that someone will Steal Your Great Idea. How can you protect yourself from the threat of someone taking your basic idea, turning it into a working, sellable game, and leaving you with nothing?

One of the participants of this course, Dan Rosenthal, has kindly written an article that details the basics of IP (intellectual property) law as it pertains to games. The article admits to being US-centric, but the core idea (which is worth repeating here) should be sound no matter where you are:

Remember, ideas are not copyrightable, they're not trademarkable, not trade secretable, and both difficult and prohibitively expensive to patent. You can't protect them anyway, and you shouldn't try — instead you should try to come up with new ones, and start working on the good ones. Don't freak out when you see things like Game Jams, or this course and think "Ian says I should post my work to the discussion forum, but I came up with a Great Idea(tm) and I don't want other people to steal it." Ideas are commonplace in games, and the value of your idea is nothing compared to the value of the implementation of that idea, your expertise and hard work in developing it into something that's going to make you real money. But most importantly, our industry is very lateral, very tight-knit, very collaborative. You'll find people sharing their ideas at GDC, doing collaborative projects between studios, or using inspiration from one game's mechanics to improve another. Don't fight it. That's the way things work, and by embracing that open atmosphere, you'll be far better off.

Level 4: The Early Stages of the Design Process

By ai864

We have already made some games in this course, so we have already been through the creation process on a small scale. But our method of design, for the most part, has been ad-hoc: here are a bunch of elements, just throw them together and call it a game. The results of this type of design can be expected to be hit-and-miss.

What about for larger projects where the stakes are higher? Is there a process that can be followed that will lead to *better* games? There is the iterative process, to be sure, but we have not gone into detail on any of the iterative steps (design, playtesting, evaluation). How exactly do you come up with an initial design? What is the most effective way to playtest? When evaluating a game, what do you look for, and how do you know what to change? These are the things we will be concerned with throughout the rest of this course.

Today, we examine the first step of the iterative process: initial design.

Course Announcements

Just a couple of comments based on feedback received from the first forum challenge:

- When the assignment is to design a game, I am looking for something that is in a state where you could print out the rules, create the components, and play. Some people posted, say, a card game and just handwaved over this: "The deck contains 50 cards, here are a couple of samples but I haven't designed the others yet." This is not a complete design. I encourage you to make complete designs, when you are designing a **game** and not just a concept. It is these details that are the essence of design.
- If there is confusion over what I am asking for, as I'm sure will happen just about every time, do feel free to ask for clarifications in the blog comments. However, I also encourage you to make your own interpretations to the best of your ability. In classroom courses (and usually in the Real World) you can ask for immediate clarification if a constraint is unclear; but in this course, since we go so quickly, there is not always time to post a comment, wait for me to respond, and then read the comment. Be creative but fair in your interpretations.

A Note on Constraints

An interesting thing happened for this Monday's challenge: more people attempted the Black Diamond (highest difficulty) than those who did all of the other difficulties combined. By contrast, when signing up, only about a tenth of the participants identified themselves as experienced game designers. What is going on here?

Part of it may be pride. Even though people will admit a total lack of experience to me in private email, broadcasting it on forums is another thing entirely. Part of it may be the thrill of the challenge. People want to know just how far they can push themselves.

However, part of it may be that adding constraints makes a challenge *easier*. This sounds unintuitive; after all, isn't a new constraint just one more thing you can't do? With more roadblocks, shouldn't a task be harder? Not always, in the case of game design.

To understand this, we can look at the process of game design as a **successive layering of constraints** on a game. Every new rule you add, every resource you define, is just one more constraint on the players. At the start of the design process you may have nothing, and the players could do anything at all; by the end, the player experience is sharply defined and heavily constrained in a way that is fun. (We will address what "fun" actually is, later in the course.)

To put this in perspective, consider the so-called genre of "open world" video games (popularized by *Grand Theft Auto*). The typical player reaction is that these games let you do anything, they give complete freedom to the player, and that is why they are fun. However, a critical look at the games shows that they do *not* give complete freedom. The games actually constrain the player in many ways: there are only certain ways a player can move, a defined set of objects they can interact with, and the autonomous computer-controlled agents that wander around are governed by specific algorithms. The player has many decisions and a relatively open set of goals, to be sure, but there are a great deal of constraints that lead to this illusion of "being able to do anything."

If you accept this explanation, that design is the creation of constraints, then you can see that constraints imposed from outside can be thought of as providing some of the initial design. By adding constraints, there is less design work to be done. Thus explains the paradox.

Constraints can also provide a useful anchor for your ideas. If I just say "go make a game" with no constraints, many people would just sit there like a deer in headlights, wondering where to begin. By adding a constraint (such as "World War I"), the question is no longer "where do I start" but rather, "what do I do with this." And that is a much easier question to answer.

Most of the challenges in this course will involve constraints. In fact, most design in the Real World happens within constraints: a publisher asking for a game that uses a certain IP or within a certain genre or within a given time and budget, for example. One of the reasons I mention this, then, is to remind you that these constraints may sometimes seem ridiculous ("do I *really* have to come up with a concept for a My Little Pony game for DS?") but that in fact they can often make a designer's life much, much easier.

There is one other reason I mention constraints. For those rare times in your life when there are truly no constraints imposed on you by others (this is more common with "indie" development and hobbyist designers than with professionals), if you have trouble getting started, one way is to generate some constraints for yourself. Give yourself a time limit ("Game Jam" events typically challenge people to make a game in as little as 24 or 48 hours). Choose a subject matter that interests you and use it for the theme. Select a core mechanic that you'd like to explore. It can be completely arbitrary, but if you are stuck and don't know what direction to take your game, go ahead and just choose an extra constraint to get yourself moving. (With iteration, you can always remove that arbitrary constraint later if you find it's holding back your design.)

Generating Ideas

The first thing that happens in a design is that you must come up with the basic core of an idea. This isn't necessarily fully-formed, but just a basic concept. There are many different starting points for a game's design. Here are some examples, in no particular order:

- Start with the **core "aesthetics"** — what do you want the player to feel? How do you want them to react? What should the play experience be like? Then work backwards from the player experience to figure out a set of rules that will achieve the desired aesthetic. Think about the best experience you've ever had while playing a game; what game rules led to that experience?
- Start with a **rule** or **system** that you observe in everyday life, particularly one that requires people to make interesting decisions. Look at the world around you; what systems do you see that would make good games?
- Start with an **existing, proven design**, then make modifications to improve on it (the "clone-and-tweak" method). This often happens when making sequels and ports of existing games. Think of a game that you thought had potential, but didn't quite take the experience as far as they could; how would you make it better?
- Start with **technology**, such as a new game engine (for video games) or a special kind of game piece (like a rotateable base for miniature figures). Find a way to make use of it in a game. What kinds of items do you have lying around your living space that have never been used in a board game before, but that would make great game "bits"?
- Start with **materials** from other sources, such as existing art or game mechanics that didn't make it in to other projects. Design a game to make use of them. Do you have an art portfolio, or earlier game designs that you didn't turn into finished products? What about public domain works, such as Renaissance art? How could you design a game around these?
- Start with a **narrative** and then design game rules to fit, making a story-driven game. What kinds of stories work well in games?
- Start with **market research**: perhaps you know that a certain demographic is underserved, and want to design a game specifically for them. Or maybe you just know that a certain genre is "hot" right now, and that there are no major games of that type coming out in a certain range of dates, so there is an opportunity. How do you turn this knowledge into a playable game?

- Combinations of several of these. For example, starting with **core aesthetics** and **narrative** at the same time, you can make a game where the story and gameplay are highly integrated.

When you think of new ideas for games, what kinds of ideas do you have? What are your starting points? What does this say about you as a designer, and the kinds of games you are likely to make?

Other Methods of Idea Generation

If you are stuck with "designer's block" (the game design equivalent of "writer's block") there are a number of strategies you'll see mentioned in various places. Here are a few:

- Keep a permanent collection of all of your ideas for games, mechanics, stories, and everything else. Look back through it from time to time to see if there's anything from years ago that you can use. Add to it whenever an idea occurs to you that you can't use immediately, but that you want to return to later.
- Think of something random. Try to find a way to integrate it into your game.
- Do some research. Learn about some aspect of the game in more depth, and you will likely find new ideas.
- Go back to the basic. Think of the formal elements of your game. What are the player goals? Rules? Resources? And so on. Note that you'll need to define these anyway in order to have a game, so by focusing on these one at a time it may give you new questions to answer.
- Formalized brainstorming, either alone or in a group. Some people swear by this method, while others say the results are questionable. The best I can say is that the results are highly unpredictable… as is the case with most R&D.
- Think critically about games. You may have *my* textbook on game design that contains some of what Brenda and I have learned over the years, but you should write your *own* book over the course of your lifetime (whether you publish it or not, at least keep it for yourself). When you discover something that does or doesn't work in a game and you think you can identify the root cause as a "law" (or at least a guideline) of game design that is broadly applicable, write it down! If you don't know why, write *that* down too, and come back to it periodically until you find the answer.
- Play lots of games! But… play as a *designer* and not just a player. Don't just play for enjoyment. Instead, play critically. Ask yourself what choices were made by the designer of the game, and why you think those choices were made, and whether or not they work. Play games in genres that you don't like or have never tried, and try to figure out why other people find them fun. Also, published hint guides can be useful to read — they are basically glorified design documents that detail all of the systems of a game!
- And lastly, practice. Work on your own projects. The more you make games, the better you get at making them… just like any other art form.

Prototyping

Remember, the more times you can iterate on your idea, the better the final game will be. Once you have a basic idea, the next step is to get it in playable form as quickly and cheaply as possible. That will leave you with as much time as possible to playtest and iterate.

As mentioned last time, iteration is the most critical for those parts of your game that have high **design risk**. For "clone-and-tweak" games where you are mostly lifting gameplay from an existing game, rapid prototyping is less important. This does not mean that "clone" games do not benefit from iteration, but simply that you should use it selectively in those areas where you *are* innovating. Keep this in mind for today's challenge.

"Laws" of Prototyping

Remember that the entire purpose of prototyping is to maximize the number of iterative cycles. Corollary: do everything you can to reduce the time required in each iteration. Now, consider that each iterative cycle consists generally of four steps: design, prototyping, playtesting, and evaluation. Of these steps, where can you save time?

- You can't really reduce the time it takes to design the rules of the game, without compromising your goals. You can't rush creativity.
- You *can* reduce time spent in playtesting by being efficient about scheduling and designing playtests to give maximum information for minimum play time… but there is a natural limit to this, and beyond a certain point you can't rush through playing the game.
- Evaluation doesn't take very long; you're making a simple yes/no decision of whether the game is "done" or "good enough" based on playtest results. There is little to be gained by rushing through this further.
- So, that leaves reducing the time it takes to create a prototype.

Some things to keep in mind when building a playable prototype:

- Build it as fast as possible. Cut corners. Make it as ugly and cheap as you can get away with.
- Minimize what you need to build. Only do what is absolutely necessary to evaluate your game. If you're trying to test out a new combat system, you do not need to build the entire exploration system. If you're making a card game, hand writing on index cards is faster to make than typing everything into Powerpoint, printing on heavy card stock, and cutting it all out manually. There is a time and place for making nice-looking components, and the early stages of game design is **not** that time or that place.
- Make your prototype easy to change. You **will** find problems in playtesting, so make it easy to adjust on the fly.

All of these guidelines push designers towards one inevitable direction…

Prototyping in Paper

You can call it "paper" or "cardboard" or "non-digital" or "analog" or any number of things, but the idea is to have a physical, tabletop game that is playable without computers (or at least, without requiring programming code). Programming is wonderful and powerful but it is also slow and expensive in comparison to paper prototypes. Here are some advantages of paper prototyping:

- It is cheap. Most systems can be prototyped with little more than a pencil and some paper, although I will give suggestions for other components for those of you that have some money to spend.
- It's fast. You don't have to mess around with programming, or layouts, or artwork. Just write a few words on a scrap of paper.
- It's easy to change. Don't like one of your numbers? Erase it and write in a new one.
- There is no guilt about throwing it away. You came up with an idea that didn't work? Oh well, you lost a whole half hour. Big deal. It's like making stick-figure drawings: if your first attempt at drawing a stick figure doesn't work, it only took you a few seconds, so just cross it out and try again.
- Paper can be used to model most gameplay systems. Yes, even most of the ones we normally associate with being specific to video games.
- By making something *playable*, you are forced to actually **design the systems**. No more handwaving of "this game will have 50 undefined cards". You have to actually do your job as the game designer, and design the game!

Limitations of Paper

Paper prototypes do have some limitations that you should be aware of:

- They cannot always handle "twitch" (dexterity or timing based) mechanics… although be aware that there are many dexterity-based non-digital games. Consider the similarities and differences between the *Street Fighter* series of video games, and James Ernest's real-time card battle game *Brawl*. Some things carry over well… others, not so much.
- Information that is hidden to both players but that still requires bookkeeping, such as the "Fog of War" mechanics prevalent in Real-Time Strategy video games. Again, note that this can sometimes be worked around — the classic children's game *Battleship* has "fog-of-war-like" mechanics, and the board game *Clue* has information hidden from all players.
- Extremely complex calculations are tedious on paper, and the systems that use them may be better suited to "prototyping" in a spreadsheet program like *Excel*. However, if the complex systems are a necessary and core part of the game, it may be a sign that "the computer is having more fun than the player" (to quote Sid Meier), and that perhaps some simplification would make the game more accessible.
- "Eye candy" such as high-quality art and animation is obviously not prototyped easily with stick-figure drawings and handwritten cards. Then again, these are not part of the game mechanics. If your game relies on visuals rather than systems, that is a sign that you are not doing a strong enough job as the systems designer.
- Paper prototypes are not very well suited for testing the user interface (UI) of a video game. Computer UIs are dynamic, but paper is static. You can get an idea of the visual

layout with some paper sketches, but to know how it will actually be used on a computer, you'd need a digital prototype.

As you can see, the advantages of paper prototyping are very general and the limitations are specific, so the ability to prototype in paper is an important skill for any game designer to develop, whether they work in video games or board games or anything in between.

The Non-Digital Designer's Prototyping Kit

What follows is a list of materials that I have personally found useful when prototyping. Other designers may have their favorite materials, so I look forward to seeing the discussion that will inevitably be generated by this list:

- Paper, of several varieties: blank, lined, and graph. These are useful for general note-taking, and for the fast construction of makeshift game boards and other surfaces.
- Colored pens and pencils. Obviously you need something to write with. Colors give an easy way to differentiate between game elements, or to annotate your game components.
- Index cards (3″x5″). These make it easy to make cards. They shuffle reasonably well. You can cut them in halves or thirds for different card sizes. You can also just write ideas down on these and tape them on the wall, making it easy to arrange your thoughts visually. These are versatile and cheap.
- Scissors and tape. For breaking things apart and sticking them together, respectively. These are to game design what WD-40 and duct tape are to handymen, for the same reasons.
- Paper clips and/or binder clips. This lets you store related materials in one place. For example, if you create several "decks" of index cards, this lets you hold them together so they don't get mixed up with each other (or worse, mixed up with cards from *other* prototypes).
- Glass beads (sometimes called "Pente stones") in different colors. These make great markers, counters, and playing pieces.
- Dice, of varying types (4, 6, 8, 10, 12 and 20 sided). Several of each type, in different colors. These provide independent random variables (as opposed to the dependent randomness of card draws). For more information on the uses of dice and cards, see Chapter 5 in the *Challenges* text. Note that dice can also make decent playing pieces that can simultaneously "store" a single number on them — for example, a six-sided die could represent a warrior with up to 6 hit points.
- A small bag of low-value coins (pennies in the United States, and I admit ignorance to how other countries handle coin-based currency). Coins make good markers, they can be flipped for a random variable, they have two sides so they can represent either of two states (such as which of two players currently controls them), and they can be stacked more easily than dice or glass beads.
- Colored sticky-dots (small round adhesive labels). You can put them on stones, dice or pennies to mark them, differentiate them, or customize them. You can write on the dots to provide additional information if needed.

- A paper notebook that is kept with your prototypes and used *exclusively* for taking notes in playtests. This is not something you want to accidentally lose track of!

Where do you find these things? It depends where you live. Most, you can get at an office supply store (Staples, Office Depot and Office Max are the big stores near where I live; you may have others), except for dice, glass beads, and coins. The coins, you can get at a bank (in the United States, you can get a hundred pennies for only one dollar — a bargain by any standard for quality game components). Dice are generally found in hobby-game stores or comic-book shops, or purchased online. Glass beads can be found in a variety of places. Hobby-game stores have them. Pet stores that sell fish equipment may sell them as aquarium stones. Art/craft hobby stores may sell them as glass beads for jewelry and craft projects. They also come as components with many games (notably *Pente*), if you can find a game with glass beads for cheap.

Craft and hobby stores, both the retail chains and online (Michael's is the big chain store in my area), can offer great inspiration for game designers. I've found large quantities of unpainted and colored wooden cubes (great as resource markers and also as custom dice) and wooden discs (they feel better and are larger than pennies). Once, I found a set of flat painted wooden cut-outs, maybe an inch square, of bunnies and another set of carrots; I don't know what I'll ultimately do with them, but there is a game that will be made with them some day. Craftparts has wooden people-shaped pawns and square tiles in various sizes. These kinds of quality components may not be immediately suitable for quick-and-dirty paper prototypes, but they can certainly come into use as your project becomes more developed.

Your First Paper Prototype

Here are the rules for the classic children's game *Battleship*:

- Players: 2
- Objective: sink all five ships in your opponent's fleet before they do the same to you.
- Setup: Each player has a 10×10 grid of squares, with the rows labeled with numbers 1 through 10 and the columns labeled with letters A through J. Each player has five ships: one ship that is 2 squares long, two ships that are each 3 squares long, one ship that is 4 squares long and one ship that is 5 squares long. Each player secretly places their ships on their own grid, in such a way that each ship is oriented sideways or up-and-down (not diagonally) and that ships do not overlap. A player is chosen to go first.
- Progression of play: On a player's turn, they call out a single square by its coordinates (such as "B-5″ or "H-10″). If the named square is not occupied by any of the opponent's ships, the opponent says "Miss". If the square *is* occupied, the opponent says "Hit". Additionally, if the square was a "hit" *and* the ship that was hit has had all of its sections hit, the ship is considered "sunk" and the opponent must tell you which ship was sunk. No matter what the result, after the action is resolved, play passes to the opponent.
- Resolution: When one player sinks all five ships of the opponent's fleet, that player is the winner.

Normally, this game is available in toy stores. It comes on a plastic board with plastic pegs. Some fancy electronic versions require batteries and have sound. But I bet if you think about it, you could prototype this game in paper in less than five minutes. How would you do this?

If you couldn't guess, all you'd have to do is draw two 10×10 grids on a sheet of paper for each of the two players (one to keep track of your fleet, and one to track the results of your shots against the opponent). This is all you need to play, and it gives pretty much the same experience as the "real" version!

Now, try this thought experiment: critically analyze *Battleship* as a game. What are the weaknesses of its design? How would you modify the rules of the game to make it better? If you are taking this course in a group, discuss this with your colleagues. Then, consider: how would you modify your paper prototype to test out your new rules in a playtest to see if they work? Usually, this is trivial to do. Here are some examples from the times when I've taught this course in a classroom:

- Allow players to move their own ships if they haven't yet been hit. (To modify the prototype: just allow players to erase and re-draw their ships.)
- Allow players to use a "sonar sweep" instead of firing a shot on their turn: they name any 3×3 square area on the board, and the opponent says the number of squares in that area (from 0 to 9) that are occupied by ships. (No modifications necessary, just play with this as a new rule.)
- Let players take another turn immediately if they score a "hit". (Again, no modifications necessary, just play with this new rule.)
- Use differently-shaped ships: instead of lines, have a T-shaped or square-shaped ship, like *Tetris* pieces. (To modify the prototype, just draw the ships in different shapes.)
- Give each player one area-effect bomb that hits everything in an entire 3×3 square area. They can use it on their turn instead of taking a normal shot, but only once per game per player. (Again, just play with the modified rules.)
- Shorten the game by playing on a 6×6 grid instead of 10×10. (Just draw the grid differently on paper.)

As you can see, modifying the rules to a paper prototype is very fast and easy, and you could go through many iterations in a short period of time. Don't be afraid that your idea will be "bad"! Of *course* it will be bad. Even experienced designers create "bad" games in their first iteration. But you will never turn it into a good game unless you start somewhere. A paper prototype is very often the ideal starting point.

Prototyping Realtime Systems

For a turn-based game like *Battleship*, a non-digital prototype is easy enough to put together. What if you wanted to prototype a First-Person Shooter video game like *Halo*? Is there any *possible* way to do that on paper, when most of the game is running around and shooting things in real time? The answer is yes, absolutely. Here are some hints:

- One "turn" of a board game is equivalent to some amount of time (say, 3 seconds) of real-time play
- For "twitch" mechanics like dodging and accuracy that require accurate timing, either a player succeeds or fails at these based on how difficult they are and how skilled the player is. This can be modeled with a random die roll. Note that even though the video game's system is not random at all, it may as well be random from the *opponent's* perspective: if I shoot at you and you either do or do not successfully dodge, I have no control over that.
- Many real-time games take place on an open 3D map that is not subdivided into "spaces". This does not prevent you from making a game board that has spaces anyway.

For example, consider these rules:

- Players: 2 to 6, free-for-all
- Objective: shoot your opponents
- Setup: Players start at designated starting locations on the board. The board is subdivided into hexagons ("hexes"). Each player is facing in one of the six directions leading away from their space towards another hex. Each player takes a set of the following cards: Move, Turn, Move/Turn, Fire.
- Progression of Play: Each turn, all players select one of their cards and play face-down. Cards are revealed simultaneously. First, any players who selected Move get to move up to 2 spaces away in any direction(s), but they cannot turn and must continue to face the same direction they started the turn facing. Next, any players who selected Move/Turn may move up to 1 space and also change their facing by a single hex (60 degrees) in either the clockwise or counterclockwise direction. Next, any players who selected Turn can change their facing to any direction they want. Finally, any player who chose Fire immediately hits and kills any opponent(s) that they can see. Any player who is killed is eliminated from the game. After the turn, players collect the card they played. They may play this card or another one on the next turn.
- Resolution: When one player is left standing, that player wins. If two or more players shoot and kill each other on the same turn simultaneously, the game is a draw.

Then you just draw up a quick hex map, maybe fill in a few hexes to represent obstacles that players cannot walk or shoot through, and play. Try it out!

And if you do try it out, you'll immediately notice the game needs some iteration. For example, I didn't define what a player "can see" so there is no way from the rules above to tell if a shot hits or not. You will have to define this more explicitly on your own (maybe it means in a straight line, or maybe within a certain range, or maybe something else). You may also notice that the game is not very deep; there are no respawns, power-ups, ammo, health packs, special weapons, or anything else. The game does not immediately support common variants like Capture-the-Flag or King-of-the-Hill. All of these things could be added, however, in just a few minutes.

What would this kind of prototype be useful for? You could use it to playtest a proposed level layout, before implementing it in the game's level editing tools. If you add enemy monsters and play as a cooperative team, and you add limited ammunition and health as new mechanics, you could balance the number of monsters versus the amount of ammo and health on a level to get a pretty decent first stab at a level that would provide a desired level of challenge. If you add different weapon types with varying range, damage and accuracy, you could get a pretty good idea of which weapons would be the most powerful on a given map. You would still need to revisit these things if you turn it into a digital game, because things do not transition 100% perfectly from one medium to another… but you will have a better starting point, and a better understanding of the game's mechanics and how they are likely to interact.

And maybe even if the digital game fails, you'll still at least have a fun little tabletop game to play with your friends.

I hope this example serves to show you that most video games can have at least *some* of their elements prototyped in paper. And naturally, games that are *meant* to be released in non-digital form can be prototyped that way as well. Even some systems from tabletop RPGs and LARPs can be prototyped in this way, in their early stages.

A Short Note about Grids

There are many ways to make a game board, but here are three common ways to get you started:

- Subdivide into a grid of squares. Square grids are easy to navigate and are familiar to most players, so they will not intimidate casual players as much as some other methods. For grids that include lots of obstacles and movement challenges, grids are ideal because it is easy to block off a path: a single impassable square forces you to go quite a bit out of your way to get to the other side. The drawback of squares is that you inevitably run into a problem with diagonal movement: does it count as one space or two in order to move diagonally? One space feels too fast; two spaces feels too slow. (The actual value is the square root of 2, or about 1.4 spaces… but if you're dealing with whole-number values this obviously does not work.)
- Subdivide into a grid of hexes. Hexes have some nice mathematical properties to them, in that something that is 3 hexes away is *always* that many hexes, no matter which of several paths you take; this gets around the "how fast to move along a diagonal" problem of square grids. On the down side, hex boards make it much easier to move around obstacles, so movement is a lot less constrained. This may be desireable or not, depending on the nature of your game. Also, hexes are quite "geeky" and are likely to put off players who are not that experienced with this style of play.
- Open area, no board. Use a tape measure instead, and move your pieces a certain number of inches (or centimetres, or what have you) per turn. This gives the most fluid and precise movement, although it has many of the same disadvantages as hex maps, and is also vulnerable to someone accidentally bumping the table and sending pieces slightly off of where they were.

Adding Features versus Keeping It Simple

As mentioned earlier, our First-Person Shooter prototype is just begging for extra features, such as health and ammo. Why not start with all of these extra systems already in place, as opposed to starting with just the simple core system? There are a few reasons to start with a simple, core rule set and then add on one rule at a time, instead of trying to design the entire game in one big effort:

- If the basic, core rules don't work, then adding extra rules on top of it will generally not make it work. Get the basics working first, *before* you start adding complexity.
- In fact, if you build extra rules on an unstable foundation, the real underlying problems in your design could be obscured! Something might *seem* wrong, but if there are a lot of systems and resources and game objects it can be hard to tell if you're experiencing a problem with the core mechanics, or the balance of a particular resource, or the design of the map, or something else.

Early on in a design process, it's generally better to keep things as simple as possible. For every rule or mechanic or object or resource that you want to include, ask yourself: is this *really* necessary *right now*? At this point, let your laziness override your creativity. It is far easier to add something to your design than to take it away, so add the minimum possible to have a working, playable game.

If you have trouble with this, try writing down a list of all of the ideas you have that you want to include in the game, and then cross off as many as you can. Ask if whatever items are left on your list would make a complete, playable game. If so, try to cross off more, until you absolutely can't anymore.

It may also help to run your idea by another designer who is not personally and emotionally attached to your pet idea. Invite them to be merciless in deciding which of your rules can be trashed. For the purposes of this course, you can offer a trade with any colleagues in your area: you look at my prototype, I'll look at yours!

Moving Forward

Once you have the core gameplay, and it *works*, then you can add new features. The temptation at this point is to add everything you originally thought of. Resist this temptation. Instead, add *one* new feature, and playtest again until the new feature works, or you have decided that it doesn't work and it needs to be abandoned.

Why not add everything at once? Because every new thing you add may have some problems with it. If you only add one new rule and a critical game system becomes broken in playtesting, you know *exactly* where the problem is, because you only changed one thing. If you add ten new rules and something breaks, it's harder to isolate which rule (or combination of rules) caused the problem. Incidentally, this part is similar to programming: if you write code in small chunks and then unit test, it's easier to find bugs than if you write ten thousand lines of code between tests.

Yes, this is tedious. You have to playtest, then change one rule, then playtest again, then change another rule, and keep doing this dozens (or even hundreds) of times. The first few playtests are fun, but you will quickly become sick of the whole business. This is part of the process of design. Sometimes, game design is hard work that is not particularly fun. This is something you need to accept if you have aspirations to become a professional designer. Just remember that the purpose of this is to make a game that *is* fun, and if it's not there yet, that should be your incentive to change something and playtest again until you reach your goal.

In making an actual game, the next step after the physical prototype (once you're happy with it) is to **document** it. These documents do not have to be 500-page Game Design Bibles. They can be small sets of rules and design and playtest notes that you've accumulated as you went through the iterative process, but formatted into something that is readable and understandable by someone who has not seen the project before. This documentation will be valuable reference material for later, if you ever forget what you were doing. Sometimes you have to put an idea to the side for a few months and return to it later, and I guarantee you will forget all of those details that used to seem second-nature to you when you were fiddling with the rules early on.

Readings

There are some additional readings this week:

- *Challenges for Game Designers*, Chapter 4. This details the process of prototyping a video game in paper. Even if your interest is in board game design, note that many commercially-successful board games originated in the video game world (there are, for example, board-game versions of *DOOM*, *Warcraft 3*, *Civilization*, *Age of Empires*, and *World of Warcraft*, among many others). Some of them are even worth playing.
- *Don't Be a Vidiot*, by Greg Costikyan. If you want to be a video game designer, this article provides both an incentive to study board games, and also a starting point for the kinds of games that are out there beyond *Monopoly* and *RISK*.

Homeplay

Do Challenge 4-1 in the text. It does not have to be an Activision/Blizzard game. It cannot be a game that already has a commercially-available board game adaptation. (Check BoardGameGeek if in doubt.)

Design a board-game adaptation of any video game. Post your complete rule set on the forums. Include a list of all components necessary to play. This game should be playable without the player having to design anything!

As above, and once you've finished your design, make a playable prototype of the core systems in under an hour. On the forum, give a complete list of materials used.

As above, and the video game in question must be an adaptation of an Atari 2600 title. And make it more fun than the original!

I would ask this time that you **stay within your experience level**. For example, if you have no game design experience prior to this course, do the basic challenge, even if you are capable of doing the others, and post in the Green Circle forum. You can certainly tackle the more advanced constraints on your own, but I'd like to try it this way to see if you get superior peer feedback. Thank you for cooperating.

Make a post on the Forums before next Monday. Then, as with last time, find at least two peers at the same difficulty level, and (if you are Blue Square or Black Diamond) three people at the next lower difficulty level, and offer constructive feedback.

Mini-Challenge

Here's another quick thing you can try if you get through all of that. Propose a rule change to *Battleship* that will make it better than the original, and find a way to express it in less than 135 characters. Post to Twitter with the #GDCU tag. You have until Monday. One rule change per participant, please!

Additional Resources

While not required reading, I can recommend these two articles for their relevance to today's topic:

- Veteran designer Raph Koster provides his own list of game bits that work well for prototypes.
- An article on paper prototyping, written for an audience of video game developers.

Level 5: Mechanics and Dynamics

By ai864

Until this point, we have made lots of games and game rules, but at no point have we examined what makes a *good* rule from a *bad* one. Nor have we really examined the different *kinds* of rules that form a game designer's palette. Nor have we talked about the relationship between the game rules and the player experience. These are the things we examine today.

Course Announcements

No major announcements today, but for your curiosity I did compile a list of tweets for the last challenge (add or change a rule to *Battleship* to make it more interesting):

- "Reveal" was a common theme (such as, instead of firing a shot, give the number of Hits in a 3×3 square – thus turning the game from "what number am I thinking of" into "two-player competitive Minesweeper")
- Skip a few turns for a larger shot (for example, skip 5 turns to hit everything in an entire 3×3 area). The original suggestion was an even number (skip 9 turns to nuke a 3×3 square) but note that there isn't much of a functional difference between this and just taking one shot at a time.
- Like *Go*, if you enclose an area with a series of shots, all squares in the enclosed area are immediately hit as well (this adds an element of risk-taking and short-term versus long-term tradeoffs to the game – do you try to block off a large area that takes many turns but has an efficient turn-to-squares-hit ratio, or do you concentrate on smaller areas that give you more immediate information but at the cost of taking longer in aggregate?)
- When you miss but are in a square adjacent to an enemy ship, the opponent must declare it as a "near miss" (without telling you what direction the ship is in), which doesn't exactly get around the guessing-game aspect of the original but should at least speed play by giving added information. Alternatively, with *any* miss, the opponent must give the distance in squares to the nearest ship (without specifying direction), which would allow for some deductive reasoning.
- Skip (7-X) turns to rebuild a destroyed ship of size X. If the area in which you are building is hit in the meantime, the rebuild is canceled. (The original suggestion was skip X turns to rebuild a ship of size X, but smaller ships are actually more dangerous since they are harder to locate, so I would suggest an inverse relationship between size and cost.)
- Each time you sink an enemy ship, you can rebuild a ship of yours of the same size that was already sunk (this gives some back-and-forth, and suggests alternate strategies of scattering your early shots to give your opponent less room to rebuild)
- Once per game, your Battleship (the size-4 ship) can hit a 5-square cross (+) shaped area in a single turn; using this also forces you to place a Hit on your own Battleship (note that

this would also give away your Battleship's location, so it seems more like a retaliatory move when your Battleship is almost sunk anyway)

We will revisit some of these when we talk about the kinds of decisions that are made in a game, next Monday.

Readings

This week I'm trying something new and putting one of the readings up front, because I want you to look at this first, before reading the rest of this post.

- *MDA Framework* by LeBlanc, Hunicke and Zabek. This is one of the few academic papers that achieved wide exposure within the game industry (it probably helps that the authors are experienced game designers). There are two parts of this paper that made it really influential. The first is the Mechanics/Dynamics/Aesthetics (MDA) conceptualization, which offers a way to think about the relationship of rules to player experience, and also the relationship between player and designer. The second part to pay attention to is the "8 kinds of fun" which we will return to a bit later in the course (Thursday of next week).

Now, About That MDA Framework Thing…

LeBlanc *et al.* define a game in terms of its Mechanics, Dynamics, and Aesthetics:

- Mechanics are a synonym for the "rules" of the game. These are the constraints under which the game operates. How is the game set up? What actions can players take, and what effects do those actions have on the game state? When does the game end, and how is a resolution determined? These are defined by the mechanics.
- Dynamics describe the *play* of the game when the rules are set in motion. What strategies emerge from the rules? How do players interact with one another?
- Aesthetics (in the MDA sense) do not refer to the visual elements of the game, but rather the *player experience* of the game: the effect that the dynamics have on the players themselves. Is the game "fun"? Is play frustrating, or boring, or interesting? Is the play emotionally or intellectually engaging?

Before the MDA Framework was written, the terms "mechanics" and "dynamics" were already in common use among designers. The term "aesthetics" in this sense had not, but has gained more use in recent years.

The Process of Design

With the definitions out of the way, why is this important? This is one of the key points of the MDA paper. The game designer only creates the Mechanics directly. The Dynamics emerge from the Mechanics, and the Aesthetics arise out of the Dynamics. The game designer may *want* to design the play experience, or at least that may be the ultimate goal the designer has in mind… but as designers, we are stuck building the rules of the game and hoping that the desired experience emerges from our rules.

This is why game design is sometimes referred to as a **second-order design problem**: because we do not define the solution, we define something that creates something else that creates the solution. **This is why game design is hard**. Or at least, it is one reason. Design is not just a matter of coming up with a "Great Idea" for a game; it is about coming up with a set of rules that will implement that idea, when two-thirds of the final product (the Dynamics and Aesthetics) are not under our direct control.

The Process of Play

Designers start with the Mechanics and follow them as they grow outward into the Aesthetics. You can think of a game as a sphere, with the Mechanics at the core, the Dynamics surrounding them, and the Aesthetics on the surface, each layer growing out of the one inside it. One thing the authors of MDA point out is that this is *not* how games are experienced from the player's point of view.

A player sees the surface first – the Aesthetics. They may be *aware* of the Mechanics and Dynamics, but the thing that really makes an immediate impression and that is most easily understood is the Aesthetics. This is why, even with absolutely no knowledge or training in game design, *anyone* can play a game and tell you whether or not they are having a good time. They may not be able to articulate *why* they are having a good time or *what* makes the game "good" or "bad"… but anyone can tell you right away how a game makes them feel.

If a player spends enough time with a game, they may learn to appreciate the Dynamics of the game and now their experience arises from them. They may realize that they do or don't like a game because of the specific kinds of interactions they are having with the game and/or the other players. And if a player spends even more time with that game, they may eventually have a strong enough grasp of the Mechanics to see how the Dynamics are emerging from them.

If a game is a sphere that is designed from the inside out, it is *played* from the outside in. And this, I think, is one of the key points of MDA. The designer creates the Mechanics and everything flows outward from that. The player experiences the Aesthetics and then their experience flows inward. As designers, we must be aware of **both** of these ways of interacting with a game. Otherwise, we are liable to create games that are fun for designers but not players.

One Example of MDA in action

I mentioned the concept of "spawn camping" earlier in this course, as an example of how players with different implicit rule sets can throw around accusations of "cheating" for something that is technically allowed by the rules of the game. Let us analyze this in the context of MDA.

In a First-Person Shooter video game, a common mechanic is for players to have “spawn points” – dedicated places on the map where they re-appear after getting killed. Spawn points are a **mechanic**. This leads to the **dynamic** where a player may sit next to a spawn point and immediately kill anyone as soon as they respawn. And lastly, the **aesthetics** would likely be frustration at the prospect of coming back into play only to be killed again immediately.

Suppose you are designing a new FPS and you notice this frustration aesthetic in your game, and you want to fix this so that the game is not as frustrating. You cannot simply change the aesthetics of the game to “make it more fun” – this may be your goal, but it is not something under your direct control. You cannot even change the dynamics of spawn camping directly; you cannot tell the players how to interact with your game, except through the mechanics. So instead, you must change the mechanics of the game – maybe you try making players respawn in random locations rather than designated areas – and then you hope that the desired aesthetics emerge from your mechanics change.

How do you know if your change worked? Playtest, of course!

How do you know *what* change to make, if the effects of mechanics changes are so unpredictable? We will get into some basic tips and tricks near the end of this course. For now, the most obvious way is designer intuition. The more you practice, the more you design games, the more you make rules changes and then playtest and see the effects of your changes, the better you will get at making the right changes when you notice problems… and occasionally, even creating the right mechanics in the first place. There are few substitutes for experience… which, incidentally, is why so much of this course involves getting you off your butt and making games .

“If the computer or the game designer is having more fun than the player, you have made a terrible mistake.”

This seems as good a time as any to quote game designer Sid Meier. His warning is clearly directed at video game designers, but applies just as easily to non-digital projects. It is a reminder that we design the Mechanics of the game, and designing the Mechanics is fun for us. But it is *not* the Mechanics that are fun for our *players*. A common design mistake is to create rules that are fun to create, but that do not necessarily translate into fun gameplay. Always remember that you are creating games for the players and not yourself.

Mechanics, Dynamics and Complexity

Generally, adding additional mechanics, new systems, additional game objects, and new ways for objects to interact with one another (or for players to interact with the game) will lead to a greater complexity in the dynamics of the game. For example, compare *Chess* and *Checkers*.

Chess has six kinds of pieces (instead of two) and a greater number of actions that each piece can take, so it ends up having more strategic depth.

Is more complexity good, or bad? It depends. *Tetris* is a very simple but still very successful game. *Advanced Squad Leader* is an incredibly complex game, but still can be considered successful for what it is. Some games are so simple that they are not fun beyond a certain age, like *Tic-Tac-Toe*. Other games are too complex for their own good, and would be better if their systems were a bit more simplified and streamlined (I happen to think this about the board game *Agricola*; I'm sure you can provide examples from your own experience).

Do more complex mechanics *always* lead to more complex dynamics? No – there are some cases where very simple mechanics create extreme complexity (as is the case with *Chess*). And there are other cases where the mechanics are extremely complicated, but the dynamics are simple (imagine a modified version of the children's card game *War* that did not just involve comparison of numbers, but lookups on complex "combat resolution" charts). The best way to gauge complexity, as you may have guessed, is to play the game.

Feedback Loops

One kind of dynamic that is often seen in games and deserves special attention is known as the **feedback loop**. There are two types, **positive feedback loops** and **negative feedback loops**. These terms are borrowed from other fields such as control systems and biology, and they mean the same thing in games that they mean elsewhere.

A positive feedback loop can be thought of as a reinforcing relationship. Something happens that causes the same thing to happen again, which causes it to happen yet again, getting stronger in each iteration – like a snowball that starts out small at the top of the hill and gets larger and faster as it rolls and collects more snow.

As an example, there is a relatively obscure shooting game for the NES called *The Guardian Legend*. Once you beat the game, you got access to a special extra gameplay mode. In this mode, you got rewarded with power-ups at the end of each level based on your score: the higher your score, the more power-ups you got for the next level. This is a positive feedback loop: if you get a high score, it gives you more power-ups, which make it easier to get an even higher score in the next level, which gives you even more power-ups, and so on.

Note that in this case, the reverse is also true. Suppose you get a low score. Then you get fewer power-ups at the end of that level, which makes it harder for you to do well on the next level, which means you will probably get an even lower score, and so on until you are so far behind that it is nearly impossible for you to proceed at all.

The thing that is often confusing to people is that *both* of these scenarios are *positive* feedback loops. This seems counterintuitive; the second example seems very "negative," as the player is doing poorly and getting fewer rewards. It is “positive” in the sense that the effects get stronger in magnitude on each iteration.

There are three properties of positive feedback loops that game designers should be aware of:

1. They tend to destabilize the game, as one player gets further and further ahead (or behind).
2. They cause the game to end faster.
3. The put emphasis on the early game, since the effects of early-game decisions are magnified over time.

Feedback loops usually have two steps (as in my *The Guardian Legend* example) but they can have more. For example, some Real-Time Strategy games have a positive feedback loop with four steps: players explore the map, which gives them access to more resources, which let them buy better technology, which let them build better units, which let them explore more effectively (which gives them access to more resources… and the cycle repeats). As such, detecting a positive feedback loop is not always easy.

Here are some other examples of positive feedback loops that you might be familiar with:

- Most “4X” games, such as the *Civilization* and *Master of Orion* series, are usually built around positive feedback loops. As you grow your civilization, it lets you generate resources faster, which let you grow faster. By the time you begin conflict in earnest with your opponents, one player is usually so far ahead that it is not much of a contest, because the core positive feedback loop driving the game means that someone who got ahead of the curve early on is going to be *much* farther ahead in the late game.
- Board games that feature building up as their primary mechanic, such as *Settlers of Catan*. In these games, players use resources to improve their resource production, which gets them more resources.
- The physical sport *Rugby* has a minor positive feedback loop: when a team scores points, they start with the ball again, which makes it slightly more likely that they will score again. The advantage is thus given to the team who just gained an advantage. This is in contrast to most sports, which give the ball to the opposing team after a successful score.

Negative feedback loops are, predictably, the opposite of positive feedback loops in just about every way. A negative feedback loop is a balancing relationship. When something happens in the game (such as one player gaining an advantage over the others), a negative feedback loop makes it harder for that same thing to happen again. If one player gets in the lead, a negative feedback loop makes it easier for the opponents to catch up (and harder for a winning player to extend their lead).

As an example, consider a “Kart-style” racing game like *Mario Kart*. In racing games, play is more interesting if the player is in the middle of a pack of cars rather than if they are way out in front or lagging way behind on their own (after all, there is more interaction if your opponents are close by). As a result, the *de facto* standard in that genre of play is to add a negative feedback loop: as the player gets ahead of the pack, the opponents start cheating, finding better power-ups and getting impossible bursts of speed to help them catch up. This makes it more difficult for the player to maintain or extend a lead. This particular feedback loop is sometimes referred to as "rubber-banding" because the cars behave as if they are connected by rubber bands, pulling the leaders and losers back to the center of the pack.

Likewise, the reverse is true. If the player falls behind, they will find better power-ups and the opponents will slow down to allow the player to catch up. This makes it more difficult for a player who is behind to fall further behind. Again, both of these are examples of *negative* feedback loops; "negative" refers to the fact that a dynamic becomes weaker with iteration, and has nothing to do with whether it has a positive or negative effect on the player's standing in the game.

Negative feedback loops also have three important properties:

1. They tend to stabilize the game, causing players to tend towards the center of the pack.
2. They cause the game to take longer.
3. They put emphasis on the late game, since early-game decisions are reduced in their impact over time.

Some examples of negative feedback loops:

- Most physical sports like *Football* and *Basketball*, where after your team scores, the ball is given to the opposing team and they are then given a chance to score. This makes it less likely that a single team will keep scoring over and over.
- The board game *Starfarers of Catan* has a negative feedback loop where every player with less than a certain number of victory points gets a free resource at the start of their turn. Early on, this affects all players and speeds up the early game. Later in the game, as some players get ahead and cross the victory point threshold, the players lagging behind continue to get bonus resources. This makes it easier for the trailing players to catch up.
- My grandfather was a decent *Chess* player, generally better than his children who he taught to play. To make it more of a challenge, he invented a rule: if he won a game, next time they played, his opponent could remove a piece of his from the board at the start of the game (first a pawn, then two pawns, then a knight or bishop, and so on as the child continued to lose). Each time my grandfather won, the next game would be more challenging for him, making it more likely that he would eventually start losing.

Use of Feedback Loops

Are feedback loops good or bad? Should we strive to include them, or are they to be avoided? As with most aspects of game design, it depends on the situation. Sometimes, a designer will deliberately add mechanics that cause a feedback loop. Other times, a feedback loop is discovered during play and the designer must decide what (if anything) to do about it.

Positive feedback loops can be quite useful. They end the game quickly when a player starts to emerge as the winner, without having the end game be a long, drawn-out affair. On the other hand, positive feedback loops can be frustrating for players who are trying to catch up to the leader and start feeling like they no longer have a chance.

Negative feedback loops can also be useful, for example to prevent a dominant early strategy and to keep players feeling like they always have a chance to win. On the other hand, they can also be frustrating, as players who do well early on can feel like they are being punished for succeeding, while also feeling like the players who lag behind are seemingly rewarded for doing poorly.

What makes a particular feedback loop "good" or "bad" from a player perspective? This is debatable, but I think it is largely a matter of player perception of fairness. If it feels like the game is artificially intervening to help a player win when they don't deserve it, it can be perceived negatively by players. How do you know how players will perceive the game? Playtest, of course.

Eliminating Feedback Loops

Suppose you identify a feedback loop in your game and you want to remove it. How do you do this? There are two ways.

The first is to shut off the feedback loop itself. All feedback loops (positive and negative) have three components:

- A "sensor" that monitors the game state;
- A "comparator" that decides whether to take action based on the value monitored by the sensor;
- An "activator" that modifies the game state when the comparator decides to do so.

For example, in the earlier kart-racing negative feedback loop example, the "sensor" is how far ahead or behind the player is, relative to the rest of the pack; the "comparator" checks to see if the player is farther ahead or behind than a certain threshold value; and the "activator" causes the opposing cars to either speed up or slow down accordingly, if the player is too far ahead or behind. All of these may form a single mechanic ("If the player is more than 300 meters ahead of all opponents, multiply everyone else's speed by 150%"). In other cases there may be three or more separate mechanics that cause the feedback loop, and changing any one of them will modify the nature of the loop.

By being aware of the mechanics causing a feedback loop, you can disrupt the effects by either removing the sensor, changing or removing the comparator, or modifying or removing the effect of the activator. Going back to our *The Guardian Legend* example (more points = more power-ups for the next level), you could deactivate the positive feedback loop by either modifying the sensor (measure something other than score… something that does not increase in proportion to how powered-up the player is), or changing the comparator (by changing the scores required so that later power-ups cost more and more, you can guarantee that even the best players will fall behind the curve eventually, leading to a more difficult end game), or changing the activator (maybe the player gets power-ups through a different method entirely, such as getting a specific set of power-ups at the end of each level, or finding them in the middle of levels).

If you do not want to remove the feedback loop from the game but you do want to reduce its effects, an alternative is to add *another* feedback loop of the opposing type. Again returning to the kart-racing example, if you wanted to keep the "rubber-banding" negative feedback loop, you could add a positive feedback loop to counteract it. For example, if the opposing cars get speed boosts when the player is ahead, perhaps the player can go faster as well, leading to a case where being in the lead makes the entire race go faster (but not giving an advantage or disadvantage to anyone). Or maybe the player in the lead can find better power-ups to compensate for the opponents' new speed advantage.

Emergence

Another dynamic that game designers should be aware of is called **emergent gameplay** (or **emergent complexity**, or simply **emergence**). I've found this is a difficult thing to describe in my classroom courses, so I would welcome other perspectives on how to teach it. Generally, emergence describes a game with simple mechanics but complex dynamics. "Emergent complexity" can be used to describe *any* system of this nature, even things that are not games.

Some examples of emergence from the world outside of games:

- In nature, insect colonies (such as ants and bees) show behavior that is so complex, it appears to be intelligent enough that we call it a "hive mind" (much to the exploitation of many sci-fi authors). In reality, each individual insect is following its own very simple set of rules, and it is only in aggregate that the colony displays complex behaviors.
- Conway's Game of Life, though not actually a "game" by most of the definitions in this course, is a simple set of sequential rules for simulating cellular life on a square grid. Each cell is either "alive" or "dead" on the current turn. To progress to the next turn, all living cells that are adjacent to either zero or one other living cells are killed (from isolation), and living cells adjacent to four or more other living cells are also killed (from overcrowding); all dead cells adjacent to exactly three living cells are "born" and changed to living cells on the next turn; and any cell adjacent to exactly two living cells stays exactly as it is. Those are the only rules. You start with an initial setup of your choice, and then modify the board to see what happens. And yet, you can get incredibly complex behaviors: structures can move, mutate, spawn new structures, and any number of other things.
- Boid's Algorithm, a way to simulate crowd and flocking behavior that is used in some CG-based movies as well as games. There are only three simple rules that individuals in a flock must each follow. First, if there are a lot of your companions on one side of you and few on the other, it means you're probably at the edge of the flock; move towards your companions. Second, if you are close to your companions, give them room so you don't

crowd them. Third, adjust your speed and direction to be the average of your nearby companions. From these three rules you can get some pretty complex, detailed and realistic crowd behavior.

Here are some examples of emergent gameplay:

- In fighting games like the *Street Fighter* or *Tekken* series, "combos" arise from the collision of several simple rules: connecting with certain attacks momentarily stuns the opponent so that they cannot respond, and other attacks can be executed quickly enough to connect before the opponent recovers. Designers may or may not intentionally put combos in their games (the earliest examples were not intended, and indeed were not discovered until the games had been out for awhile), but it is the mechanics of stunning and attack speed that create complex series of moves that are unblockable after the first move in the series connects.
- In the sport of *Basketball*, the concept of "dribbling" was not explicitly part of the rules. As originally written, the designer had intended the game to be similar to how *Ultimate Frisbee* is played: the player with the ball is not allowed to move, and must either throw the ball towards the basket (in an attempt to score), or "pass" the ball to a teammate (either through the air, or by bouncing it on the ground). There was simply no rule that prevented a player from passing to himself.
- Book openings in *Chess*. The rules of this game are pretty simple, with only six different piece types and a handful of special-case moves, but a set of common opening moves has emerged from repeated play.

Why do we care about emergent dynamics? It is often desired for practical reasons, especially in the video game world, because you can get a lot of varied and deep gameplay out of relatively simple mechanics. In video games (and to a lesser extent, board games) it is the *mechanics* that must be implemented. If you are programming a video game, emergent gameplay gives you a great ratio of hours-of-gameplay to lines-of-code. Because of this apparent cost savings, "emergence" as a buzzword was all the rage a few years ago, and I still hear it mentioned from time to time.

It's important to note that emergence is not always planned for, and for that matter it is not always desirable. Here are two examples of emergence, both from the *Grand Theft Auto* series of games, where unintended emergent gameplay led to questionable results:

- Consider these two rules. First, running over a pedestrian in a vehicle causes them to drop the money they are carrying. Second, hiring a prostitute refills the player's health, but costs the player money. From these two unrelated rules, we get the emergent strategy that has been affectionately termed the "hooker exploit": sleep with a prostitute, then run her over to regain the money you spent. This caused a bit of a scandal in the press back in the day, from people who interpreted this dynamic as an intentional design that glorified violence against sex workers. Simply saying "it's emergent gameplay!" is not sufficient to explain to a layperson why this was not intentional.

- Perhaps more amusing was the combination of two other rules. First, if the player causes damage to an innocent bystander, the person will (understandably) defend themselves by attacking the player. Second, if a vehicle has taken sufficient damage, it will eventually explode, damaging everything in the vicinity (and of course, nearly killing the driver). These led to the following highly unrealistic scenario: a player, driving a damaged vehicle, crashes near a group of bystanders. The car explodes. The player crawls from the wreckage, barely alive… until the nearby crowd of "Samaritans" decides that the player damaged them from the explosion, and they descend in a group to finish the player off!

As you can see, emergence is not always a good thing. More to the point, it is not *necessarily* cheaper to develop a game with emergent properties. Because of the complex nature of the dynamics, emergent games require a lot more playtesting and iteration than games that are more straightforward in their relationships between mechanics and dynamics. A game with emergence may be easier to program, but it is much harder to design; there is no cost savings, but rather a *shift* in cost from programmers to game designers.

From Emergence to Intentionality

Player intentionality, the concept from Church's *Formal Abstract Design Tools* mentioned earlier in this course, is related in some ways to emergence. Generally, you get emergence by having lots of small, simple, interconnected systems. If the player is able to figure out these systems and use them to form complicated chains of events intentionally, that is one way to have a higher degree of player intention.

Another Reading

- *Designing to Promote Intentional Play* by Clint Hocking. This was a lecture given live at GDC in 2006, but Clint has kindly made his Powerpoint slides and speaker notes publicly available for download from his blog. It covers the concept of player intentionality and its relation to emergence, far better than I can cover here. The link goes to a Zip file that contains a number of files inside it; start with the Powerpoint and the companion Word doc, and the presentation will make it clear when the other things like the videos come into play. I will warn you that, like many video game developers, Clint tends to use a lot of profanity; also, the presentation opens with a joke about Jesus and Moses. It may be best to skip this one if you are around people who are easily offended by such things.

Lessons Learned

The most important takeaway from today is that game design is not a trivial task. It is difficult, mainly because of the nature of MDA. The designer creates rules, which create play, which create the player experience. Every rule created has a doubly-indirect effect on the player, and this is hard to predict and control. This also explains why making one small rules change in a game can have ripple effects that drastically alter how the game is played. And yet, a designer's task is to create a favorable player experience.

This is why playtesting is so important. It is the most effective way to gauge the effects of rules changes when you are uncertain.

Homeplay

Today we will practice iterating on an existing design, rather than starting from scratch. I want you to see first-hand the effects on a game when you change the mechanics.

Here are the rules for a simplified variant of the dice game called *Bluff* (also called *Liar's Dice*, but known to most people as *that weird dice game that they played in the second Pirates of the Caribbean movie*):

- Players: 2 or more, best with a small group of 4 to 6.
- Objective: Be the last player with any dice remaining.
- Setup: All players take 5 six-sided dice. It may also help if each player has something to hide their dice with, such as an opaque cup, but players may just shield their dice with their own hands. All players roll their dice, in such a way that each player can see their own dice but no one else's. Choose a player to go first. That player must make a bid:
- Bids: A "bid" is a player's guess as to how many dice are showing a certain face, *among all players*. Dice showing the number 1 are "wild" and count as *all* other numbers. You cannot bid any number of 1s, only 2s through 6s. For example, "three 4s" would mean that between every player's dice, there are *at least* three dice showing the number 1 or 4.
- Increasing a bid: To raise a bid, the new bid must be *higher* than the previous. Increasing the number of dice is *always* a higher bid, regardless of rank (nine 2s is a higher bid than eight 6s). Increasing the rank is a higher bid if the number of dice is the *same or higher* (eight 6s is a higher bid than eight 5s, both of which are higher than eight 4s).
- Progression of Play: On a player's turn, that player may either raise the current bid, or if they think the most recent bid is incorrect, they can challenge the previous bid. If they raise the bid, play passes to the next player in clockwise order. If they challenge, the current round ends; all players reveal their dice, and the result is resolved.
- Resolution of a round: If a bid is challenged but found to be correct (for example, if the bid was "nine 5s" and there are actually eleven 1s and 5s among all players, so there were indeed at least nine of them), the player who challenged the bid loses one of their dice. On subsequent rounds, that player will then have fewer dice to roll. If the bid is challenged *correctly* (suppose on that bid of "nine 5s" there were actually only eight 1s and 5s among all players), the player who made the incorrect bid loses one of their dice instead. Then, all players re-roll all of their remaining dice, and play continues with a new opening bid, starting with the player who won the previous challenge.
- Game resolution: When a player has lost all of their dice, they are eliminated from the game. When all players (except one) have lost all of their dice, the one player remaining is the winner.

If you don't have enough dice to play this game, you can use a variant: dealing cards from a deck, for example, or drawing slips of paper numbered 1 through 6 out of a container with many such slips of paper thrown in.

If you don't have any friends, spend some time finding them. It will make it much easier for you to playtest your projects later in this course if you have people who are willing to play games with you.

At any rate, your first "assignment" here is to **play the game**. Take particular note of the dynamics and how they emerge from the mechanics. Do you see players bluffing, calling unrealistically high numbers in an effort to convince their opponents that they have more of a certain number than they actually do? Are players hesitant to challenge, knowing that any challenge is a risk and it is therefore safer to *not* challenge as long as you are not challenged yourself? Do any players calculate the odds, and use that information to influence their bid? Do you notice any feedback loops in the game as play progresses – that is, as a player starts making mistakes and losing dice, are they *more* or *less* likely to lose again in future rounds, given that they receive fewer dice and therefore have less information to bid on?

Okay, that last question kind of gave it away – yes, there is a positive feedback loop in this game. The effect is small, and noticeable mostly in an end-game situation where one player has three or more dice and their one or two remaining opponents only have a single die. Still, this gives us an opportunity to fiddle with things as designers.

Your next step is to **add, remove, or change one rule in order to remove the effect of the positive feedback loop**. Why did you choose the particular change that you did? What do you expect will happen – how will the dynamics change in response to your modified mechanic? Write down your prediction.

Then, **play the game again** with your rules modification. Did it work? Did it have any other side effects that you didn't anticipate? How did the dynamics *actually* change? Be honest, and don't be afraid if your prediction wasn't accurate. The whole *point* of this is so you can see for yourself how hard it is to predict gameplay changes from a simple rules change, without actually playing.

Next, **share what you learned** with the community. I have created a new page on the course Wiki. On that page, write the following:

1. What was your rules change?
2. How did you expect the dynamics of the game to change?
3. How did they *really* change?

You don't need to include much detail; a sentence or two for each of the three points is fine.

Finally, your last assignment (this is mandatory!) is to **read at least three other responses**. Read the rules change first, and without reading further, ask yourself how you think that rule change would modify gameplay. Then read the other person’s prediction, and see if it matches yours. Lastly, read what actually happened, and see how close you were.

You may leave your name, or you may post anonymously.

Mini-Challenge

Take your favorite physical sport. Identify a positive or negative feedback loop in the game. Most sports have at *least* one of these. Propose a rule change that would eliminate it. Find a way to express it in less than 135 characters, and post to Twitter with the #GDCU tag. You have until Thursday. One sport per participant, please!

Level 6: Games and Art

By ai864

At this point I'd like to take a brief diversion to go into the whole "can games be art?" thing. This may seem like a strange topic to cover in the middle of some heftier design principles. It's also one of those tired old arguments that have been going on for years now, so why waste our time retreading old ground? I have a few reasons for including this in the syllabus, and you are free to debate the merits and drawbacks of its inclusion in this course.

The first reason for this topic is that for the next few weeks we'll be talking about the whole concept of "fun" and how to make games more enjoyable. For most practicing game designers, this is their prime directive: Take This Game And Make It Fun. Before we go down that road, I want to make it clear that fun is not the *only* purpose of game design, and in fact that some games can be critically successful in their design goals even if they are not particularly "fun" in the way that most games are.

Second, as a debate that *has* been going on for ages, I want those who are new to the party to get a basic grounding in the debate. It's one of those things that will certainly come up in conversation among designers from time to time, and I want the novices among you to be prepared to enter that discussion. For those of you who are quite familiar with this already, I hope to up the ante so that we can all proceed in these discussions at a higher level of discourse.

Third, so-called "art games" – that is, games that are made primarily for the purpose of artistic expression (as opposed to entertainment) – are reaching a critical mass. There are a lot of very talented people doing very interesting things in this space right now. A lot of art games are very simple and small in scope, made by a single person in a relatively short period of time. A lot of potential avenues are yet to be explored. This makes art games a wonderful opportunity for those who are looking to establish themselves as game designers.

And finally, I know just enough about art history and art criticism to be dangerous. I am therefore driven, to an extent, to talk about an area of personal interest… even though I acknowledge that it will undoubtedly get me into trouble at some point.

Course Announcements

For those paying close attention, I recently changed my Twitter username from @ai864 to @IanSchreiber, after urging from co-author Brenda Brathwaite (@bbrathwaite). The theory is that my actual name will be easier to remember… provided people learn to spell it correctly. Keep in mind, for those of you who tweet about this course regularly.

Mini-Challenge Results

Here are a small selection of the answers to the mini-challenge from last time (identify a physical sport with a feedback loop, and propose a rule change to eliminate it):

- 8-ball (pocket billiards): Negative feedback loop is that the more of your own balls you sink, the fewer legal targets you have. Rule change: sunk balls are reset on the table, first to sink any seven of their balls can attempt to sink the 8-ball for the win. Alternate rule change: after failing to sink a ball, opponent automatically gets a point.
- Martial arts, boxing, and similar: Positive feedback loop is that the more you injure your opponent, the less likely they are to retaliate. Rule change: wait a day between rounds. (Impractical perhaps, although it would probably cut down on serious injury.)
- Soccer, Basketball, and most other team sports: Negative feedback loop where after scoring, the ball is given to the other team. Rule change: after scoring, use a "jump ball" or equivalent to give both teams an equal chance to reclaim the ball.
- Croquet: Positive feedback loop is that you get bonus swings for hitting wickets. Rule change: make the bonus swings optional, keep track of total number of swings throughout the game, lowest number of swings wins.
- Most professional sports: Positive feedback loop is that a team that wins a lot gets more money (from fans, sponsorships, etc.), which lets them buy better players, which makes it more likely they will continue to win. Rule change: not given. (This is actually a real struggle with some professional sports, because it is more exciting to watch a game if you feel like both teams have a chance to win. In the real world, some proposals to fix this include drafts and salary caps. Sports that don't do something to prevent this feedback loop tend to lose popularity. I'm looking at *you*, American Baseball.)
- Cycling, auto racing, and similar: Negative feedback loop is the ability to draft behind the person in front of you, letting you save energy so that you can overtake them later. Rule change: race in a vacuum. (Very funny, wise guy.)

Readings

Due to the positive response from Monday, I'll continue putting the readings up front. Go do these now:

- *Challenges for Game Designers*, Chapter 17 (Games as Art)
- *A Theory of Fun for Game Design*, Chapter 12 (Taking Their Rightful Place), if you chose to acquire this book.
- *Understanding Comics*, Chapter 7 (The Six Steps), if you chose to acquire this book.

What is Art?

Understanding Comics is, as you may have seen, a comic book about the art form of comic books. If you have read Chapter 7, you will immediately see a number of parallels between comic book art and game design… aside from the public image problem that both have of being "not serious" and "just for kids."

McCloud starts off by making an attempt to define "art" as **anything that is not done for the intent of survival or reproduction**. Most students I've talked to think this is an overly broad definition, but of course few can offer anything better. For what it's worth, if you accept this definition, then "games are art" is not a difficult leap – after all, when you're deeply concentrating on clearing the next *Left 4 Dead* level, or considering your next move in a game of *Chess*, you are probably not doing much to aid in either your real-world survival or reproduction (unless you play *Chess* in a uniquely erotic way, in which case I really do not need to hear about it).

I've heard the definition of art as something that is **communicative** and **transformative**. This is also overly broad, but also clearly includes games.

Dictionary.com defines art as **the quality, production, expression, or realm, according to aesthetic principles, of what is beautiful, appealing, or of more than ordinary significance**. By that definition, a game with lots of "eye candy," *or* a game that is more than "just a game" for any reason, can be considered art.

Wikipedia defines art as **the process or product of deliberately arranging elements in a way that appeals to the senses or emotions**. Games have formal elements that can be deliberately designed. Games can appeal to senses, obviously, through their visual properties if nothing else. Games can also appeal to emotions – two oft-cited examples from the video game world are the death of Floyd in *Planetfall* and the death of Aerith in *Final Fantasy VII*, although notice how emotional some people can get even by watching a sports game on television, or how many friendships and romantic relationships have ended over a game of *Diplomacy*.

My intent is not to define the term *art*; it is about as difficult and about as fruitless as the quest to define the word *game*. Rather, my point is that no matter what definition of "art" you find, it does not seem particularly difficult to include games within that definition.

Why the problem, then?

If games fit any reasonable definition of "art" that we can think of, you might wonder why this is even a debate. Why the claims, real or imagined, that "games cannot be art"?

Here it is useful to make a distinction between just plain old "art" and the more highbrow "fine art" or "high art" – the kind of timeless art that captures and communicates the essence of the human experience. Shakespeare. Da Vinci. Monet. That kind of thing. The argument, then, might not be that games are not "art" in the sense that they can be purposefully and deliberately crafted, but rather that games cannot reach the status of "high art" due to something inherent in the medium.

I won't say much about this because it is largely covered in Clint Hocking's essay *On Authorship in Games*, which he generously gave permission to reprint in the *Challenges* text.

At any rate, if you're noticing a theme in the readings, you can see where I personally stand on the issue. We may not have the game equivalent of *Mona Lisa* or *Citizen Kane* yet, but that just means an opportunity. So, let us move past this. Let's assume for the moment that games *can* be a valid medium for artistic expression, and start talking about *how* one might go about doing so.

Six Steps

I wanted to make two key points in reference to the reading in *Understanding Comics*. The first was McCloud's definition of art, above. The second is the six layers of art:

- Idea/Purpose. What is the message to be expressed, the seed of an idea for a story that must be told? Why are you creating a work of art at all?
- Form. What artistic medium will you use to express your message? Oil paints? Sculpture? Interpretive dance? Comic books? Games?
- Idiom. What McCloud calls *idiom* is more commonly called *genre* when referring to games. First-person shooter, real-time strategy, vehicle simulation, MMORPG, and so on (or for board games: resource management games, roll-and-move track games, trivia games, dice games, tile-laying games, gambling games…).
- Structure. In stories, this is the basic plot arc, characters, and other building blocks. In games, we might call this the "core mechanics" of the game. What are the structural components that form the core of the user/viewer/player experience?
- Craft. In comic books, this includes how well a story is told. With games, it is the ability to make your rules and play experience streamlined and natural, so that the players are not struggling with the rules but rather enjoying the play.
- Surface. This is the outer-layer experience: the colors, sounds, visuals, beauty, attention to the details that are immediately sensed. The "eye candy" of the piece.

McCloud notes that a typical comic book viewer experiences a work from the surface inward. First you see the surface; then, looking deeper, you enjoy the story. Looking even further, you can see the ideas behind the story, and perhaps appreciate a groundbreaking artist even if their drawing quality isn't as high as you'd like. With even more study, you can see divisions between different genres and styles, and even understand the reason why certain story elements or other conventions occur within a genre. Looking ever deeper, you can eventually gain an appreciation for the medium of comic books, understanding the ways that make it unique as an art form; and, you can see the ideas behind a work, the purpose behind what is essentially timeless literature.

You might notice that this all applies to games as well.

McCloud also notes that, while a work is encountered from the "outside in" it is still created from the "inside out" – the creator must first choose an idea and a form and then choose an idiom within that form, before ever putting pen to paper. These choices might be deliberate or they might be made rashly or emotionally, but they must be decided *first*. Then the structure must be defined; then the details fleshed out in craft; and finally the surface must be created.

Does this sound familiar? It should. It is, for the most part, a restatement of the MDA Framework.

Actually, I think of McCloud's six steps as an *extension* of MDA. Mechanics are roughly equivalent to McCloud's Structure; Dynamics are analogous to Craft; and Aesthetics are similar to Surface. It's not a direct parallel, but it is close. In both cases, the consumer is concerned with the surface, while the true artist looks toward the inner core of the artistic process.

Towards an Artistic Process

If MDA represents the outer three layers of a piece of art, how do we represent the inner three layers? To answer this, we again turn to McCloud's model.

McCloud takes one additional, important step behind LeBlanc *et al*. He states that while works are experienced from the outside in and created from the inside out, artists and other creators follow a process of learning *from the outside in*.

Think about it. What was your first "Great Idea" for a game? It probably concerned the surface characteristics of a game that you liked. *"It'll be just like Pac-Man meets Space Invaders. Only better!"* Many people start out by "modding" a game that they like, taking existing gameplay and simply changing some of the surface characteristics – changing the appearance of characters in a game, reskinning everything so that instead of Marines Shooting Aliens In Space, the game now looks like Wizards Battling Dragons In The Mountains. Same mechanics, same dynamics, different surface.

And then what happened? Maybe you played enough games to see past the surface, to see that some games with dragons and fireballs and wizards are fun but others are not, and that the difference comes not from the story or the genre but from the gameplay. And you start to see the different types of play, and which types are and aren't fun. With further experience and study you can get a good feel for what kinds of mechanics lead to compelling gameplay. And maybe that's all you want or need, to become an established designer who is known for making games in established genres that are fun.

But if you look a little past that, you'll start asking: where do genres come from? Who decides that a certain set of core mechanics can be copied from game to game, with variations, and that this particular set of mechanics creates good gameplay? How are *new* genres created? Is there a process for doing what no one else has done before, finding an elusive set of compelling mechanics that have not been discovered yet? And you could become renowned for creating one or more new genres of gameplay. You may or may not be able to take those genres and polish them as far as they can go, but you can create something that *other* people can take, those who work closer to the surface, who can then use your core ideas and perfect them.

Is that all you can do? It is probably more than any of us would aspire to in our lifetimes. And yet, you might wonder if there is something more. And there are two paths to explore: Idea and Form.

If you explore form, you can push the boundaries of the medium. What are games capable of? Can they generate emotions in the player (other than adrenaline rush and power fantasy)? What kinds of things can be expressed through games better than any other artistic medium? How can you use games in ways that the medium has not been used before – not just new gameplay styles, and not just new ways to have fun, but as a means of expression or transformation in the player? Can you change someone's mind? Can you change their life? Can you touch them in ways that a painting or movie cannot? How? And then you create experimental work, probably very small games, that explore some aspect of what games as a medium can and can't do. These games might not be particularly interesting or compelling to a wide audience, but they will give a lot of ideas to others who work within the medium, who can then use your experiments and modify them to express their own meaningful ideas.

If you explore idea/purpose, you instead have a message you want to communicate to the world, and you have chosen games as your preferred method of expression. Here, the challenge is to communicate in a medium where the player, and not the designer, is in control of the experience. You must use every trick you know in order to provide meaning through gameplay. What ideas do you want to express? What deep meaning exists in your life, that you want to share?

Lots of questions, few answers…

You might be wondering at this point if I have any answers at all about how to do this. I do not, but this is because of the nature of art.

This course is concerned primarily with the outer three layers of the art of game design: Mechanics, Dynamics and Aesthetics (or if you prefer, Structure, Craft and Surface). Teaching you how to make compelling games by creating their rules is already a daunting task for a ten-week course; teaching you to create new genres or to push the boundaries of the medium are a bit much.

But beyond that, as McCloud says, the inner three layers can't be learned from a class or a book. To reach an understanding of the inner core of an art form, you will have to spend your entire career, maybe 20 or 30 years or more, working towards this *on your own*. And that's only if you want to. You may have no interest in this, and that is perfectly okay. The world may need more people pushing the boundaries of games as an artistic medium… but the world still needs some good games, too. Only you know how far you can or want to take your art. It is not my place to tell you, but rather to point you to a road map that will let you get where you want to go.

And now, a little art history.

This is the part that always gets me into trouble, because a lot of people are mistrustful and cynical of contemporary art. Some guy calling himself an "artist" can poop in a tin can and sell it to an art gallery for 20,000 Euros, and the rest of us wonder how *we* can get people to pay us that much for our own excrement. This is art? And if so… is *this* what games aspire to be? It helps to take a step back and look at how we got this way, because games fit in quite nicely with contemporary art, and we should really understand why.

Let's take a trip back to the Renaissance, when painting was elevated to an art form. At that time, art was supposed to be a faithful representation of the world; the picture frame could be thought of as a window through which you could view this other reality. The more realistic a piece of art depicted a scene, the better the artist. Judging art was as simple as seeing how lifelike it was – simple! And then around the 1890s, the photographic camera was invented and it ruined everything.

Now, with photographs being able to create a 100% perfect reproduction, the old form of art suddenly became obsolete. Painters had to ask themselves the question: what now?

The artist Wassily Kandinsky, followed by many others, started by asking if art could be *its own* object, rather than a *representation* of something else: what if a canvas is a "screen" rather than a "window" or "mirror"? And thus came what is now called *abstract art*, art that is not symbolic or representative of anything except itself.

How do you judge this kind of art? How can you tell an artist that is genuinely talented and inspired, from a poser who just flings paint randomly at a canvas and waits for undeserved accolades?

The influential critic Clement Greenberg offered a solution: judge art purely on its aesthetic value. Technical execution is king. The artist is the creator, and a good artwork should provide the same aesthetic value, regardless of who, if anyone, is viewing it. Greenberg formalized what is now referred to as "modern art" (*modern* here refers to a specific time period in the general range of 1910 through 1950, and is not to be confused with *contemporary* art which is the art of today).

Over the next few decades, the art world experienced a rejection of Greenbergian formalism, insisting that art should not be passive but *interactive*; it should be a *dialogue* between artist and viewer; art is allowed to have *multiple interpretations*; art should carry *meaning*. This era was referred to as "postmodern" art.

During the 1960s and 70s especially, art ran into a potential problem: it became a hot commodity and suddenly big-name artists were worth a lot of money. (I hear you saying: gosh, we should all have such "problems.") Still, a lot of artists felt their work was being devalued in the sense that they made it for a *purpose* and instead it was being treated as a *commodity*. The work is not as important as the name attached to it. And while it was nice for some artists to earn a healthy living, it was at the cost of “selling out”… something that no all artists were willing to compromise on.

Now look at games. Does any of this sound familiar?

How do we judge games? Numeric review scores. Technical execution. Critical praise for the audio or graphics. Game reviews give “fun” a rating from 1 to 5, implying that what is fun for the reviewer will be equally fun for every reader. The current state of game critique is the equivalent of Greenbergian formalism. We are, apparently, stuck in an odd time warp that takes us back to 1930.

Are games more Modern or Postmodern? Are they passive, or interactive? Do games produce different play experiences for different individuals, or does a game provide the same experience for everyone? Do games simply carry visuals, or are they capable of carrying a greater meaning embedded in their mechanics? You may disagree, but I see games as very much of a Postmodern art form. I hope that some day, game reviewers start looking at games in this light.

What about money? Games are definitely suffering the “problem” of being commoditized. The video game industry exceeds $20 Billion per year these days. I don’t have any figures for the board game industry, but given how many millions of copies of *Scrabble* and *Monopoly* are sold each year, I’d imagine it is significant. Most major studios exist to make games that *make money*, and sometimes developers must compromise between their desire to make something *unique* and something that *will sell*.

What is the point of all of this? Simply that if this is an area of interest for you, it is worth your time to study art history. The world of art critics and art historians already figured out how to judge if something is “art” or not, back in 1917 when Duchamp signed a pseudonym to a urinal and called it art. In fact, while many developers imagine the art world snobbily refusing to acknowledge games as worthy of attention, this is just fantasy; the reality is that games have been on the radar of art critics for awhile now. My own literature search turned up articles as early as 1994 (this was a year before the first *PlayStation* was released, mind you). In all of the cases I could find – and I’m talking peer-reviewed academic art journals – not only are video games being analyzed, but there is an implicit assumption that games are art. I did not find any articles that wasted any time defending games as a means of artistic expression; it was an *a priori* assumption. Let’s get over our delusions of persecution, then, and make some art.

What are Art Games?

How does one go about designing a game that is artistic in its purpose rather than purely entertainment-driven? This really depends on what counts as “art,” as there are many games out there already that are primarily made as a form of expression. As you'll see, they fall into several categories. There may be other categories I am missing here… partly because there are undoubtedly games that could be called "art" that I have not yet seen, and partly because this is a largely unexplored space. But these should give you some starting points.

I'll give you a few games to play. Go ahead and play them first, if you can. Then, read down for further discussion. The following games are all playable in just a few minutes, usually five or less. Those that take longer, will at least give you the general idea of gameplay right away, and you can play them for longer or not. Play some or all, as your time allows.

Playings

- *Passage* and/or *Gravitation*, by Jason Rohrer. (5 and 8 minute play times, respectively.)
- *The Marriage*, by Rod Humble. (Playable in just a few minutes.)
- *Stars Over Half-Moon Bay*, by Rod Humble. (Playable in just a few minutes.)
- *September 12*, by Gonzalo Frasca. (Plays indefinitely, but the mechanics are simple and immediately apparent within the first minute or two.)
- *Samarost*, by Amanita. (Takes awhile to play through completely.)
- *Cloud*, by Jenova Chen. (Takes awhile to play to completion, but it shows you the major mechanics in the first level, which only takes a few minutes)

Lessons Learned

The question of whether games can be "art" will continue to be debated for some time, I'd imagine. For our purposes, it is a rather fruitless debate; if you are interested in making an artistic expression through the medium of games, then do so.

Studying art and the artistic process further can be useful to game designers. If you're wondering what to do after this course ends, that is one of many potential avenues you can explore to deepen your understanding of design.

Even if you are not looking to be an *artiste*, you may still be creating art in a sense, and it is good to understand a little bit of what art is and what artists do. As Koster says in today's *Theory of Fun* reading:

*"Most importantly, games and their designers need to acknowledge that there is no distinction between art and entertainment… all art and all entertainment are posing problems to the audience. All art and all entertainment are prodding us toward greater understanding of the chaotic patterns we see swirl around us. Art and entertainment are not terms of **type** – they are terms of **intensity**."*

Now, About Those Playings…

By looking at some of the games that seem to be referenced a lot in discussions of art, we can get some clues about how we might go about creating our own artistic statements through gameplay. I should be clear that what follows are my own personal interpretations of these works, and your experiences (and the artists' intent) may vary. I do not see this as a problem; Postmodern art allows for multiple interpretations and multiple layers of meaning.

Samarost is "art" mostly in the visual sense. It is like an interactive painting: very pleasant graphics, and a nice form of exploration. The creators are going for a particular visual reaction in the player.

Cloud takes this a step further, deliberately trying to create an emotional response in the viewer (specifically, the emotion of childlike wonder when gazing up at the clouds). Some of my students have found it coming off a bit heavy-handed in this department, and I remind them that this was an exploratory work that was trying to answer the question of whether games could induce emotion *at all*, so one can expect it to be a little wide of the mark.

Passage and *Gravitation* both express a specific idea or feeling (that of death and dying, or parenthood, respectively). Rohrer took his own emotions and did his best to translate them directly into gameplay. The difference between these games and *Cloud* is that *Cloud*'s goal is to create an emotion; with *Passage* and *Gravitation*, the goal is self-expression of the creator's emotions.

The Marriage is similar to Rohrer's work, but *The Marriage* is expressing an idea rather than an emotion (specifically, Humble is attempting to detail the mechanics behind a long-term relationship, hence the title).

September 12 also expresses an idea (mainly, that declaring war on terror is a flawed concept), but it takes things one step further. While Humble's and Rohrer's work is simply an expression of the artist's ideas and emotions, *September 12* is an attempt to persuade the audience. This is not exploration, but rhetoric, making it slightly different in purpose than the others.

Stars Over Half-Moon Bay is similar to *The Marriage* in that it is expressing an idea (in this case, it is making some statements on the creative process and how you start with an open sky of possibilities, then things get cloudy as you enter this mysterious process of creativity and innovation, and at the end things crystallize and you put together the pieces to make something permanent. As game designers and other artists struggle with the creative process, *Stars* is a bit more "meta" than *The Marriage*. While *The Marriage* could theoretically speak to an audience of anyone who wants to understand long-term relationships, *Stars* is speaking directly to an audience of other game designers on the challenges of their medium.

Looking at these in the context of McCloud's six steps, we can see some patterns emerging. Here are some potential starting points for art games:

- Use games as a medium of self-expression ("Idea/Purpose"). You might express a feeling, an idea, or an ideology. You may simply be presenting your expression, or actually persuading the audience to your point of view. For emotional expression, start with the Aesthetics (in the MDA sense) and work backwards: what emotion do you want the player to feel, what Dynamics would cause that emotion, and finally what Mechanics can create that kind of play? For expression of ideas, remember that games are systems; find the systems *behind* the ideas that you want to express, and then find the gameplay inherent in those systems. (I should mention my co-author's series of games in progress, including *Train*, which are exploring the systems behind human atrocity. Unfortunately these games are non-digital and therefore I cannot simply give you a link to play them. But I did want to point them out, lest anyone think that only *video* games are capable of being artistic.)
- Use games to explore the limitations of games-as-artistic-medium ("Form"). In this case, start with a question: can games do X (whatever "X" is)? Then, try to answer that question by designing a game that tries to do X.
- Create a traditional work of art, with interactive game-like elements ("Surface"). In this case your creative process may be different than that of game design.

Are there other artistic works you can do in the other steps? I think there are, but we have not heavily explored them yet.

Homeplay

Today I offer a choice of designs, based not on experience level (I must admit that most of us are novices in this area, even if we are experienced game designers) but on area of interest. Here are four options, all inspired by the "non-digital shorts" at the end of the *Challenges* chapter:

Option 1 (Creating emotions): Design a non-digital game that introduces children to the concept of grief. Post the rules and required components. If desired, also include commentary on how you approached this problem and why you think your game does (or does not) succeed.

Option 2 (Persuasion): Modify the board game *RISK* to advocate world peace. Post your changes to the original rules. If desired, also include commentary on what you were trying to do, whether you think you were successful, and why or why not.

Option 3 (Exploring the boundaries of games): Design a game that has intentionally incomplete rules, requiring player authorship of rules during the play of the game in order for it to be playable. Post your (incomplete) rules.

Option 4 (Exploring the nature of the medium): Choose a digital game that you consider to be artistic and inspiring. Create the rules for a non-digital version of it. Note how the difference in medium affects the experience; think about what kinds of artistic ideas are best expressed in digital or non-digital form. Post your rules, plus commentary.

Choose whichever of these most interests you, and **make a post** in the relevant forum. I have created one discussion area for each of these four options. Make your post before the next scheduled lesson here: on or before Monday, July 20, noon GMT. Then, **offer constructive feedback** to at least three other posts in the same forum (that is, with the same interest area as you). Do this on or before Thursday, July 23, noon GMT.

Call for Content

Do you have an "art game" that you have played, that is not mentioned in this lesson or in the *Challenges* book? Post it, along with a link and brief summary, to the course Wiki under the "Art Game" section. (If you are not registered for this course but still want to contribute, leave it in the comments here or post it to Twitter with the #GDCU tag, and I or someone else will add it.)

Mini-Challenge

Come up with a core concept for any kind of artistic game that can be expressed in 135 characters or less. Post it to Twitter with the #GDCU tag. One concept per person, please! Tweet before July 20, noon GMT.

Level 7: Decision-Making and Flow Theory

By ai864

I'm excited about this week, because this is where we're going to really get into the essence of game design, starting today with decision-making and continuing this Thursday with the nature of fun. These are some of my favorite topics to discuss, because it is the interactivity between players and systems that sets games apart from most other traditional media. This is where the magic of play happens, and as a systems designer, this strikes at the heart of what I deal with when making a game.

Course Announcements

I understand that with 1400+ people, there will be a wide variance in free time. Some of you are students on holiday and can throw yourselves full-bore into this course with all its challenges. Others of you have day jobs and must focus on providing for yourself and your family first. Some of you are going through transitions of a personal nature, such as moving to a new city or dealing with the death of a loved one. This is all understandable, and expected.

So, I would like to clarify my expectations. My expectation is that each of you give what you can to your education. My hope is that you will take out of this course more than you put in. That is all.

If you have a week where you can do no more than read the blog, then that is what you should do. Obviously you will learn more if you do the challenges, but if you can't, better to do something than to drop everything. This course does not have to be an all-or-nothing venture, and I never intended it to be that way.

If you must prioritize, I would suggest the following:

1. First, read the blog. This is the cornerstone of everything else in the course, so if you can do nothing else, do this. This includes the readings referenced in the blog from other sources. (If you fall behind, then hopefully you can catch up on the reading later. I will leave this blog online after the course is finished.)
2. If you have some extra time after that, but not enough to do everything, please provide feedback and peer review to your fellow participants on the forums, in the Project Postings section. I think the people that take the time to do the challenge and post it deserve to have feedback, and time constraints prevent me from giving it myself, so this must be participant-driven and grass-roots in nature. As a courtesy to others, do this if you possibly can.
3. If you have all the time you need, do the challenge and post it.

Mini-Challenge Results

Here are a small selection of the answers to the mini-challenge from last time (propose a concept for an art game):

- A game where players have a secret goal and try to guess other players' goals, as a metaphor for growing up gay.
- A game similar to The Sims, except it becomes harder to perform functions as time progresses, to get across the feeling of being afflicted with Alzheimer's.
- A first-person shooter where players are only shown a single line showing the player's line-of-sight, showing the view of a 2D hero in a Flatland world.
- A sorcery-themed game where the effects of spells are designed during play.

Readings

Read the following:

- *Challenges for Game Designers*, Chapters 5 and 6 (Chance / Strategic Skill)
- *A Theory of Fun for Game Design*, Chapters 1, 2 and 3 (Why Write This Book / How the Brain Works / What Games Are)… if you chose to acquire this book. In an earlier comment, someone suggested only reading the cartoons on each page first, then going back and reading the text afterward.
- Building a Princess Saving App (PDF), by Dan Cook. This article is actually written for an audience of interaction designers to explain what productivity applications can learn from games. In the process, though, it also happens to touch on some core concepts of game design and the nature of "fun" which is exactly what we're talking about today.

Decisions

As Costikyan pointed out in *I Have No Words*, we often use the buzzword "interactivity" when describing games when we actually mean "decision-making." Decisions are, in essence, what players *do* in a game. Remove all decisions and you have a movie or some other linear activity, not a game. As pointed out in *Challenges*, there are two important exceptions, games which have no decisions at all: some children's games and some gambling games. For gambling games, it makes sense that a lack of decisions is tolerable. The "fun" of the game comes from the thrill of possibly winning or losing large sums of money; remove that aspect and most gambling games that lack decisions suddenly lose their charm. At home when playing only for chips, you're going to play games like *Blackjack* or *Poker* that have real decisions in them; you are probably not going to play *Craps* or a slot machine without money being involved.

You might wonder, what is it about children's games that allow them to be completely devoid of decisions? We'll get to that in a bit.

Other than those two exceptions, most games have some manner of decision-making, and it is here that a game can be made more or less interesting. Sid Meier has been quoted as saying that a good game is a series of interesting decisions (or something like that), and there is some truth there. But what makes a decision "interesting"? *Battleship* is a game that has plenty of decisions but is not particularly interesting for most adults; why not? What makes the decisions in *Settlers of Catan* more interesting than *Monopoly*? Most importantly, how can you design your own games to have decisions that are actually compelling?

Things Not To Do

Before describing good kinds of decisions, it is worth explaining some common kinds of **uninteresting** decisions commonly found in games. Note that the terminology here (obvious, meaningless, blind) is my own, and is not "official" game industry jargon. At least not yet.

- **Meaningless decisions** are perhaps the worst kind: there is a choice to be made, but it has no effect on gameplay. If you can play either of two cards but both cards are identical, that's not really much of a choice.
- **Obvious decisions** at least have an effect on the game, but there is clearly one right answer, so it's not really much of a choice. Most of the time, the number of dice to roll in the board game *RISK* falls into this category; if you are attacking with 3 or more armies, you have a "decision" of whether to roll 1, 2, or 3 dice… but your odds are better rolling all 3, so it's not much of a decision except in very special cases. A more subtle example would be a game like *Trivial Pursuit*. Each turn you are given a trivia question, and if you know the correct answer it could be said that you have a decision: say the right answer, or not. Except that there's never any reason to *not* say the right answer if you know it. The fun of the game comes from showing off your mastery of trivia, not from making any brilliant strategic maneuvers. This is also, I think, why quiz shows like *Jeopardy!* are more fun to watch than to play.
- **Blind decisions** have an effect on the game, and the answer is not obvious, but there is now an additional problem: the players do not have sufficient knowledge on which to make the decision, so it is essentially random. Playing *Rock-Paper-Scissors* against a truly random opponent falls into this category; your choice affects the outcome of the game, but you have no way of knowing what to choose.

These kinds of decisions are, by and large, not much fun. They are not particularly interesting. All three represent a waste of a player's time. Meaningless decisions could be eliminated, obvious decisions could be automated, and blind decisions could be randomized without affecting the outcome of the game at all.

In this context, it is suddenly easy to see why so many games are not particularly compelling.

Consider the trivia game that popularized the genre, *Trivial Pursuit*. First you roll a die, and move in any direction, so which location you land on is a decision. Only a few spaces on the board help you towards your victory condition, so if you can land on one of those it is an obvious decision. If you can't, your choice generally amounts to which category you're strongest at, which is again obvious (or blind, to the extent that you don't know what question you would get in each category until after you choose). Once you finish moving, you're asked a trivia question. If you don't know the answer, there is no decision to be made. If you *do* know the answer, there is a decision of whether to say it or not… but there is no reason not to, so again it is an obvious decision.

Or consider the board game *Battleship* which seems to keep coming up in our discussions. Just about every decision made in this game is blind. You are given no information on which to base your decision of what space to fire at. Once you hit an enemy ship you do have *some* information, but you still don't know which direction the ship is oriented (horizontally or vertically) or where its endpoints are, so the decision is more constrained but not any less blind.

Or consider *Tic-Tac-Toe*, which has interesting strategic decisions until you reach the age where you master it and realize the way to always win or draw, at which point the decisions become obvious.

What Makes Good Decisions?

Now that we know what makes *weak* decisions, the easiest answer is "don't do that!" But we can take it a little further. Generally, *interesting* decisions involve some kind of **tradeoff**. That is, you are giving up one thing in exchange for another. These can take many different forms. Here are a few examples (again I use my own invented terminology here):

- **Resource trades**. You give one thing up in exchange for another, where both are valuable. Which is *more* valuable? This is a value judgment, and the player's ability to correctly judge or anticipate value is what determines the game's outcome.
- **Risk versus reward**. One choice is safe. The other choice has a potentially greater payoff, but also a higher risk of failure. Whether you choose safe or dangerous depends partly on how desperate a position you're in, and partly on your analysis of just *how* safe or dangerous it is. The outcome is determined by your choice, plus a little luck… but over a sufficient number of choices, the luck can even out and the more skillful player will generally win. (Corollary: if you want more luck in your game, reduce the total number of decisions.)
- **Choice of actions**. You have several potential things you can do, but you can't do them all. The player must choose the actions that they feel are the most important at the time.
- **Short term versus long term**. You can have something right now, or something better later on. The player must balance immediate needs against long-term goals.
- **Social information**. In games where bluffing, deal-making and backstabbing are allowed, players must choose between playing honestly or dishonestly. Dishonesty may let you come out better on the current deal, but may make other players less likely to deal with you in the future. In the right (or wrong) game, backstabbing your opponents may have very negative real-world consequences.
- **Dilemmas**. You must give up one of several things. Which one can you most afford to lose?

Notice the common thread here. All of these decisions involve the player judging the value of something, where values are shifting, not always certain, and not obvious.

The next time you play a game that you really like, think about what kinds of decisions you are making. If you have a particular game that you strongly *dis*like, think about the decisions being made there, too. You may find something about yourself, in terms of the kinds of decisions that you enjoy making.

What About Action Games?

At this point, the video gamers among you are wondering how any of this applies to the latest First-Person Shooter. After all, you're not exactly strategizing about resource management tradeoffs in the middle of a heated battle where bullets and explosions are flying all around you.

The short answer here is that you *are* making interesting decisions in such games, and in fact you are making them at a much faster rate than normal – often several meaningful decisions per second. To compensate for the intense time pressure, the decisions tend to be much simpler: fire or dodge? Aim or move? Duck or jump?

Time limits can, in fact, be used to turn an obvious decision into a meaningful one. Another way I prefer to say this is that **time pressure makes us stupid**. For a more thorough discussion of action games and how skill relates to them, see Chapter 7 (Twitch Skill) in *Challenges for Game Designers*.

Emotional Decisions

There is one class of decisions that is useful to consider: decisions that have an emotional impact on the player. The decision of whether to save your buddy (while using some of your precious supplies) or leave him behind to die (potentially denying yourself some AI-assisted help later on) in *Far Cry* is a resource decision, but it is also meant to be an emotional one – and certainly, an identical decision made on a real-life battlefield would come down to more than just an analysis of available resources and probabilities. Likewise, the majority of players do not play through a game with moral choices (such as *Knights of the Old Republic* or *Fable*) as pure evil – not because "evil" is a suboptimal strategy, but because even in a fictional simulated world, a lot of people can't stomach the thought of torturing and killing innocent bystanders.

Or consider a common decision made at the start of many board games: what color are you? Color is usually just a way to uniquely identify player tokens on the board, and has no effect on gameplay. However, many people have a favorite color that they always play, and can become quite emotionally attached to "their" color. It can be rather entertaining when two players who "always" play Green, play together for the first time and start arguing over who gets to be Green. If player color has no effect on gameplay, it is a meaningless decision. It *should* therefore be uninteresting, and yet some players paradoxically find it quite meaningful. The reason is that they are emotionally invested in the outcome. This is not to say that you can cover up a bad game by artificially adding emotions; but rather, as a designer, be aware of what decisions your players seem to respond to on an emotional level.

Flow theory

Let's talk a little bit about this elusive concept of "fun." Games, we are told, are supposed to be fun. The role of a game designer is, in most cases, to take a game and make it fun. I've used the word "fun" a lot in this course without really defining it, and it has understandably made some of you uncomfortable. Notice I usually enclose the word "fun" in quotation marks, on purpose. My reasoning is that "fun" is not a particularly useful word for game designers. We instinctively know what it means, sure, but the word tells us nothing about how to *create* fun. What is fun? Where does it come from? What makes games fun in the first place? We will continue to talk about this on Thursday, but I want to start talking about it now. I'm sure you can agree it has been long enough.

Interesting decisions seem like they might be fun. Is that all there is to it? Not entirely, because it doesn't say anything about *why* these kinds of decisions are fun. Or why *uninteresting* decisions are still fun for children. For this, we turn to Raph Koster.

What a lot of Koster's *Theory of Fun* boils down to is this: **the fun of games comes from skill mastery**. This is a pretty radical statement, because it equates "fun" with "learning"… and at least when I was growing up, we were all accustomed to regard "learning" with "school" which was about as *not* fun as you could get. So it deserves a little explanation.

Theory of Fun draws heavily on the work of psychologist Mihaly Csikszentmihalyi (pronounced just like it's spelled, in case you're wondering), who studied what he called the mental state of "flow" (we sometimes call it being "in the flow" or "in the zone"). This is a state of extreme focus of attention, where you tune out everything except the task you're concentrating on, you become highly productive, and your brain gives you a shot of neurochemicals that is pleasurable – being in a flow state is literally a natural high.

Csikszentmihalyi identified three requirements for a flow state to exist:

- You must be performing a **challenging activity** that requires **skill**.
- The activity must provide **clear goals** and **feedback**.
- The outcome is **uncertain** but can be **influenced** by your actions. (Csikszentmihalyi calls this the "paradox of control": you are in control of your actions which gives you indirect control over the outcome, but you do not have *direct* control over the outcome.)

If you think about it, these requirements make sense. Why would your brain need to enter a flow state to begin with, blocking out all extraneous stimuli and hyper-focusing your attention on one activity? It would only do this if it needs to in order to succeed at the task. What conditions would there have to be for a flow state to make the difference between success and failure? See above – you'd need to be able to influence the activity through your skill towards a known goal.

Csikszentmihalyi also gave five effects of being in a flow state:

- A merging of action and awareness: spontaneous, automatic action/reaction. In other words, you go on autopilot, doing things without thinking about them. (In fact, your brain is moving faster than the speed of thought – think of a time when you played a game like *Tetris* and got into a flow state, and then at some point it occurred to you that you were doing really well, and then you wondered how you could keep up with the blocks falling so fast, and as soon as you started to think about it the blocks *were* moving too fast and you lost. Or maybe that's just me.)
- Concentration on immediate tasks: complete focus, without any mind-wandering. You are not thinking about long-term tradeoffs or other tasks; your mind is in the here-and-now, because it *has* to be.
- Loss of awareness of self, loss of ego. When you are in a flow state, you become one with your surroundings (in a Zen way, I suppose).
- There is a distorted sense of time. Strangely, this can go both ways. In some cases, such as my *Tetris* example, time can seem to slow down and things seem to happen in slow motion. (Actually, what is happening is that your brain is acting so efficiently that it is working *faster*; everything else is still going at the same speed, but you are seeing things from your own point of reference.) Other times, time can seem to speed up; a common example is sitting down to play a game for "just five minutes"… and then six hours later, suddenly becoming aware that you burned away your whole evening.
- The experience of the activity is an end in itself; it is done for its own sake and not for an external reward. Again, this feeds into the whole "here-and-now" thing, as you are not in a mental state where you can think that far ahead.

I find it ironic, when a typical kid is in their "not now, I'm playing a game" mental state, the parent complains that they are "zoned out." In fact, the gamer is in a flow state, and they are "zoned *in*" to the game.

Flow States in Games

Simplifying this a bit, we know that to be in a flow state, an activity must be **challenging**. If it is too easy, then the brain has no reason to waste extraneous mental cycles, as a positive outcome is already assured. If it is too difficult, the brain *still* has no reason to try hard, because it knows it's just going to fail anyway. The goal is to hit that sweet spot where the player can succeed… but only if they try hard. You'll often see a graph that looks like this, to demonstrate:

All this says is that if you have a high skill level and are given an easy task, you're bored; if you have a low skill level and are given a difficult task, you're frustrated; but if the challenge level of an activity is comparable to your current skill level… flow state! And this is good for games, because this is where a lot of the fun of games comes from.

Note that "flow" and "fun" are not synonyms, although they are related. You can be in a flow state without playing a game (and in fact without having fun). For example, an office worker might get into a flow state while filling out a series of forms. They may be operating at the edge of their ability in filling out the forms as efficiently as possible, but there may not be any real learning going on, and the process may not be fun, merely meditative. (Thanks to Raph for clarifying this for me.)

One Slight Problem

When you are faced with a challenging task, you get better at it. It's fun because you are *learning*, remember? So, most people start out with an activity (like a game) with a low skill level, and if the game provides easy tasks, then so far so good. But what happens when the player gains some competency? If they keep getting the same easy tasks, the game becomes boring. This is essentially what happens in *Tic-Tac-Toe* when a child makes the transition to understanding the strategy of the game.

By the way, we can now answer our earlier question: why can children's games get away with a lack of meaningful decision-making? The answer is that young children are still learning valuable skills from these games: how to roll a die, move a token on a board, spin a spinner, take turns, read and follow rules, determine when the game ends and who wins, and so on. These skills are not instinctive and must be taught and learned through repeated play. When the child masters these skills, that is about the time when decision-less games stop holding any lasting appeal.

Ideally, as a game designer, you would like your game to have slightly more lasting playability than *Tic-Tac-Toe*. What can you do? Games offer a number of solutions. Among them:

- Increasing difficulty as the game progresses (we sometimes call this the "pacing" of a game). As the player gets better, they get access to more difficult levels or areas in a game. This is common with level-based video games.
- Difficulty levels or handicaps, where better players can choose to face more difficult challenges.
- Dynamic difficulty adjustment ("DDA"), a special kind of negative feedback loop where the game adjusts its difficulty during play based on the performance of the player.
- Human opponents as opposition. Sure, you can get better at the game… but if your opponent is also getting better, the game can still remain challenging if it has sufficient depth. (This can fail if the skill levels of different players fall out of synch with one another. I like to play games with my wife, and we usually both start out at about the same skill level with any new game that really fascinates us both… but then sometimes,

one of us will play the game a lot and become so much better than the other, that the game is effectively ruined for us. It is no longer a challenge.)

- Player-created expert challenges, such as new levels made by players using level-creation tools.
- Multiple layers of understanding (the whole "minute to learn, lifetime to master" thing that so many strategy games strive for). You can learn *Chess* in minutes, as there are only six different pieces… but then once you master that, you start to learn about which pieces are the most powerful and useful in different situations, and then you start to see the relationship between pieces, time, and area control, and then you can study book openings and famous games, and so on down the rabbit hole.
- Jenova Chen's *flOw* provides a novel solution to this: allow the player to change the difficulty level while playing based on their actions. Are you bored? Dive down a few levels and the action will pick up pretty fast. Are you overwhelmed? Run back to the earlier, easier levels (or the game will kick you back on its own if needed).

You'll notice that when we read in a review that a game has "replayability" or "many hours of gameplay" what we are often *really* saying is that the game is particularly good at keeping us in the flow state by adjusting its difficulty level to continue to challenge us as we get better.

Why Games?

You might wonder, if flow states are so pleasurable and they are where this elusive and mysterious "fun" comes from, why do we design games to do this and not some other medium? Why not design *productive* tasks to induce flow states, for example, so that maybe we could get a few million people working on discovering a cure for cancer instead of playing *World of Warcraft*? Why not design college classes to induce flow states, so that a student could learn a typical 50-hour class in a week (the same way they might play through a 50-hour RPG on their *PlayStation*) instead of having that same class take an entire 10 or 15 weeks?

Games just happen to be naturally good at putting players in a flow state, so it is much easier to design a fun game than a fun course in Calculus. As Koster points out in *A Theory of Fun*, the brain is a great pattern-matching machine, and it is the finding and understanding of patterns that is what is happening when our brain is in a flow state. I think games bring this out really well because you have three levels of patterns: feeling the Aesthetics, discerning the Dynamics, and finally mastering the Mechanics (in the MDA sense). Since every game has these three layers of patterns, games are three times as interesting as most other activities.

"Edutainment" Games

You might think that, if games are so great at teaching and if learning is so darned pleasurable, that educational games would be more fun than anything. In reality, of course, "Edutainment" is a dirty word that we only mention when forced, as the vast majority of games that bill themselves as "fun… *and* educational!" are actually neither. What's going on here?

Many “Edutainment” games work like this: first you’ve got this game, and you play it, and it’s maybe kind of fun. And then the game stops, and tries to give you some kind of gross, icky, disgusting *learning*. And as a reward for doing the learning, you get to play the game again. Gameplay is framed as a *reward* for the *inherently unpleasurable* task of learning something. We have a name for this: **chocolate-covered broccoli**.

I think this design contains an error of thinking, and this infects the design of such games at a fundamental level, invalidating the whole premise. The error is the separation of “learning” and “fun” because, as you now know, these are not separate concepts but rather identical (or at least strongly related). The assumption that learning is not fun and that fun cannot be inherently educational undermines the entire game… and incidentally, also reinforces the extremely damaging notion that education is a chore and not a pleasure.

It would be wonderful if we could stop teaching that “learning is not fun” lesson to our children. It would certainly make my life a lot easier as a teacher, if I did not have to first convince my students that they should be intrinsically motivated to do well in my classes.

What if you want to design a game that has the primary purpose of teaching, then? That is a subject that deserves a course of its own. My short answer: start by isolating the inherently fun aspects of learning the skills you want to teach, and then use those as your core mechanics. By integrating the learning and the gameplay (rather than keeping them as separate concepts or activities), you take a large step towards something truly worthy of the “fun, and educational” label.

A Note for Teachers

If you’ve accepted everything in this lesson so far, you might see a parallel with teaching. If learning is inherently fun, think about what you can do to draw that out of your subject.

- How many interesting decisions do your students make? I once saw a statistic that the average college student raises their hand in class once every ten weeks – that’s three meaningful decisions per year! Can you do better? Consider giving a choice of assignments (with built-in tradeoffs: for example, an easy-but-boring homework or a difficult-but-interesting one). Ask lots of questions in class that get students involved. Have class discussions or debates.
- Are too many of your students bored or overwhelmed, because your class is at a difficulty that is too low or too high? Games have this problem too, and often solve it through including multiple difficulty levels; consider having a tiered grading system where remedial students should be able to pass if they can at least put in the work to grasp the basics, while offering advanced students extra work that is more interesting. Offer the course content in layers, first going over the very basics in a “For Dummies” way that everyone can get, then add the main details that are really important, and finally give

some advanced applications that only some students might understand, but that are interesting enough that students will at least have some incentive to reach a bit.

- The most fun games are designed in a player-centric manner, concentrating first on providing a quality experience. You can tell a game where the designer made a game that *they* wanted to play, because it sold a total of five copies to the designer, the designer's close friends, and the designer's mom. You can also tell a game where the designer started with content rather than gameplay; these are the games that have deep, involving stories and incredible layers of content, but no one sees them because the gameplay is boring and people stop playing after five minutes. What would your class be like if you start your lesson plans by thinking of the student experience, rather than designing a class that *you* find interesting (your students might not share your research interests), or designing a class based around content (which is probably not engaging until *you* bring it to life)?

Lessons Learned

Decisions are the core of what a game is. When critically analyzing a game (yours or someone else's), pay attention to what decisions the players are making, how meaningful those decisions are, and *why*. The more you understand about what makes some decisions more compelling than others, the better a game designer you are likely to be.

Games are unnaturally good at teaching new skills to players. (Whether those skills are *useful* or not, varies from game to game.) Learning a new skill – by being given a challenge that forces you to try hard and increase your skill level – is one of the prerequisites for putting players in a flow state. Flow states are an intensely pleasurable state for the brain to be in, and a lot of the feeling of "fun" that comes from playing games comes from being in the flow.

There it is! Mystery solved! You now know everything there is to know about where "fun" comes from and how to create it. Okay, not really. But this is a start, and we will probe a bit deeper into the nature of "fun" this Thursday.

Feedback

If you have time, before beginning the Homeplay below, please take the time to offer **constructive feedback** to at least three other posts from the art games from Level 6 (posted on the forum). If you posted a game yourself, offer feedback to other posts in the same category as you posted yours. If you were unable to post a game last time, then choose whichever of the four topics most interests you (see the bottom of Level 6 for details).

Try to complete your feedback on or before Thursday, July 23, noon GMT.

Homeplay

This time around, let's practice our ability to provide meaningful decision-making to players. We are going to make a mod of a game. (This homeplay is adapted from Challenge 6-1 in the *Challenges* text.)

Here are the basic rules for the children's card game *War*:

- Players: 2
- Materials: one standard 52-card deck of playing cards (with Jokers removed)
- Setup: Shuffle the deck of cards and deal out half to each player. Players take their cards in a single face-down stack.
- Progression of Play: Simultaneously, players flip over the top card of their deck. If the cards are of unequal rank (number), the higher rank wins the "battle" and that player takes both cards and sets them aside in a discard pile (Aces are high, then King, Queen, Jack, then 10 down to 2). When a player runs out of cards in their deck, they take their discard pile and turn it over to form a new deck.
- Wars: If players flip over cards that are equal rank, it starts a "war." Players each deal three cards face-down, then flip up a new card face-up, and the higher rank of the new face-up card wins all face-down and face-up cards. This process is repeated if the new face-up cards are the same rank, continuing with additional "wars" until eventually one player wins.
- Resolution: The game ends when one player runs out of cards, *or* when one player must play cards for a "war" but they do not have enough cards remaining in their deck and discard pile. That player loses the game, and their opponent wins.

This game has no decisions whatsoever. The game mechanically plays itself, and the players merely act as tools to play out the game to its pre-determined conclusion.

Change the rules so that the game outcome is determined primarily by **skill**, and that the game now has **interesting, meaningful decisions**.

Post in the forum that most closely resembles your skill and experience level as a designer:

Beginner, little or no experience prior to taking this course.

Intermediate, some coursework or exposure to game design but little or no professional experience.

Advanced, at least some professional experience as a published game designer.

Make your post on or before Thursday, July 23, noon GMT. Then, **offer constructive feedback** to at least two other posts in the same forum, *and* at least three posts in one forum "below" yours if you posted in Blue Square or Black Diamond.

Mini-Challenge

Add a rule that gives interesting decisions to *Trivial Pursuit* that can be expressed in 135 characters or less. Post it to Twitter with the #GDCU tag. One rule change per person, please! Tweet before July 23, noon GMT.

Parting Shot

If you're still curious, "Csikszentmihalyi" is pronounced "Chick-sent-me-high". Seriously, I'm not making this up. It's not exact, but it's about as close as you can get with an English-language transliteration.

Level 8: Kinds of Fun, Kinds of Players

By ai864

On Monday, we discovered that “fun” is really just another word for “learning” and that putting players in a flow state is where this elusive “fun” comes from. Today we dig deeper into this concept to learn more about “fun,” digging into LeBlanc *et al.*’s “8 kinds of fun” and relating that back to flow theory and other things.

We currently have an idea of *what* is fun, but it would help to know *why* these things are fun. What if there are new kinds of fun waiting to be discovered?

Course Announcements

I will be at Protospiel this weekend, so I may be a bit slow in responding to email or validating forum accounts. Likewise, next Monday’s lesson may be slightly delayed in posting, depending on what shape I’m in when I return.

Mini-Challenge Results

Here are a small selection of the answers to the mini-challenge from last time (propose a rule change to add interesting decisions to *Trivial Pursuit*):

- Answering player hears all six questions on the card, then predicts the number they’ll get right. If they don’t overestimate how many they’ll get right, they get N points (where N is the number of correct answers); otherwise they get nothing. Presumably, players play to a total score rather than moving around the board. This decision is interesting when the player is not entirely sure whether an answer is correct, and they must choose their level of risk (based on how certain they are and their relative score).
- After earning a wedge, you can choose to keep answering additional questions on the card for additional wedges (or additional turns), but if you miss one then you lose all of the ones you’ve earned that turn. An interesting push-your-luck mechanic.
- Instead of rolling to move, a player can attempt to answer a question of the color of a nearby space (anywhere within 6 spaces) to move there. If they fail, they do not move. Another risk/reward mechanic, where you risk completely wasting your turn in exchange for more precise movement.
- Once per turn, you can force another player to answer a question for you after hearing it. If they get it wrong, your turn continues; if they get it right, your turn ends. Reminiscent of “You Don’t Know Jack.”
- You can get more than one wedge of the same color. You may trade with opponents at any time.

- You read your own question. After looking at the answer, if you are incorrect, you can bluff and claim you were correct anyway. If no one else challenges you, then proceed as if you had answered correctly. If you are challenged… well, the original tweet didn't specify, but presumably the winner of the challenge gains something and the loser loses something. I'd recommend, loser of a challenge loses their next turn, and if the challenger was correct it immediately becomes their turn (possibly skipping other players in the process).

Readings

Read the following:

- *Natural Funativity*, by Noah Falstein. We've talked a lot about what is fun, and from the MDA Framework we know there are different kinds of fun. But why are these things fun in the first place, and not other things? Noah provides a useful theory.
- *Clubs, Diamonds, Hearts, Spades: Players Who Suit MUDs*, by Richard Bartle. If you're too young to know what a MUD is, it is basically a precursor to today's MMO. Replace the word "MUD" with "World of Warcraft" and it will still make perfect sense.
- You may also find it useful to review the *MDA Framework*, specifically the part that talks about the 8 kinds of fun.

Kinds of Fun

You may remember from the MDA Framework that the authors listed 8 kinds of fun. These are:

- **Sensation**. Games can engage the senses directly. Consider the audio and video "eye candy" of video games; the tactile feel of the wooden roads and houses in *Settlers of Catan*; or the physical movement involved in playing sports, *Dance Dance Revolution*, or any game on the Nintendo *Wii*.
- **Fantasy**. Games can provide a make-believe world (some might cynically call it "escapism") that is more interesting than the real world.
- **Narrative**. As we mentioned earlier in passing, games can involve stories, either of the embedded kind that designers put there, or the emergent kind that are created through player action.
- **Challenge**. Some games, particularly retro-arcade games, professional sports, and some highly competitive board games like *Chess* and *Go*, derive their fun largely from the thrill of competition. Even single-player games like *Minesweeper* or activities like mountain climbing are fun mainly from overcoming a difficult challenge.
- **Fellowship**. Many games have a highly social component to them. I think it is this alone that allows many American board games like *Monopoly* to continue to sell many copies per year, in spite of the uninteresting decisions and dull mechanics. It is not the game, but the social interaction with family, that people remember fondly from their childhood.

- **Discovery**. This is rare in board games, but can be found in exploration-type games like *Tikal* and *Entdecker*. It is more commonly found in adventure and role-playing video games, particularly games in the *Zelda* and *Metroid* series.
- **Expression**. By this, I think the MDA authors mean the ability to express yourself through gameplay. Examples include games like *Charades* or *Poker* where the way that you act is at least as important as what other actions you take within a game; *Dungeons & Dragons* where the character you create is largely an expression of your own personal idea; or open-world and sim video games like *The Sims* or *Grand Theft Auto* or *Oblivion* or *Fable*, which are largely concerned with giving the player the tools needed to create their own custom experience.
- **Submission**. A name that often has my students chuckling with their dirty minds, but the intent is games as an ongoing hobby rather than an isolated event. Consider the metagame and the tournament scene in *Magic: the Gathering*, the demands of a guild to show up at regular meetings in *World of Warcraft*, or even the ritualized play of games at a weekly boardgame or tabletop-roleplaying group.

Recall that these are not all-or-nothing propositions. Games can contain several kinds of fun, in varying quantities.

Why not just create a game that has all eight kinds of fun? Wouldn't that be the holy grail of games, the game that's fun for everyone? Unfortunately, no. Just because these are different kinds of fun does not mean that *everyone* finds *all eight* of these things fun at all. Not only do different games provide different combinations and relative quantities of the various kinds of fun, but different players find different combinations more or less fun than others. About half of the people I run into think that *Chess* is fun, and the other half do not; the "fun" Aesthetic arises not from the game alone, but the combination of game and player.

Are these eight the *only* kinds of fun? No; even the authors admit the above list is incomplete. There are many classification schemes out there to identify different kinds of fun, including Nicole Lazzaro's four fun keys, or Pierre-Alexandre Garneau's fourteen forms of fun. Even the 8 kinds of fun from the MDA paper are debatable. Is it meaningful to separate Fantasy and Narrative, or are they just two ways of looking at the same kind of fun? Is submission really a kind of fun, or is it what happens when you have a game compelling enough to earn the status of "hobby" – is it a *cause* or an *effect*? What, exactly, counts as "expression" and what does not?

And where does the whole "fun is learning, learning is fun" thing from last time come into this discussion?

Evolution (sans Pokemon)

Falstein's answer is to take a trip back to early pre-history, when humans were at their hunter-gatherer stage. Primitive humans had to learn many skills in order to survive and reproduce. If we found it fun to learn certain skills, we would be more likely to practice them, and thus more likely to survive, reproduce, and pass on our genes to the next generation. Over time, those things that made us most likely to survive ended up being the things that we find "fun" today. Not all primitive hunter-gatherer skills are necessarily *useful* today, mind you, but our genetics haven't had time to catch up with our technology yet.

In short: **if a caveman found it useful, you'll find it fun.**

Falstein proposes three kinds of fun: "physical fun" (useful for any physical feats that allow us to fight or escape danger), "mental fun" (the problem-solving part of our brain that gave us such useful things as the wheel and fire), and "social fun" (the benefits of banding together in groups for mutual survival… and, of course, reproduction).

When I first saw this, I thought "wow!" Except I spelled it "WoW"… because, what is *World of Warcraft*, but physical fun (combat), mental fun (optimizing your equipment and skills), and social fun (dancing Night Elves)?

But we can apply this evolutionary thought process to any "kinds of fun." Let us look some of the MDA's 8 kinds of fun in this context:

- Sensation includes physical movement (good for building muscle) and looking at and hearing things that are interesting (good for detecting opportunities or dangers).
- Fantasy allows the kind of "what-if" scenario part of our brain to get stronger, allowing us to come up with novel ideas.
- Narrative is useful for passing on vital information and experience to others in your group, increasing the chance that *all* of you will survive.
- Challenge is a convenient way for different humans to show dominance over one another in a relatively safe way – "I can throw this rock further than you" is more useful than "let's fight to the death" if you're trying to build a colony.
- Fellowship opens up the possibility of new food sources (a single one of us might get killed hunting a large beast, but a group of us together can take it down). It's also rather hard to pass on your genetic material to the next generation if you're alone.
- Discovery is what makes us want to explore our nearby territory. The more territory we know, the more potential places for us to find food and shelter.
- Expression probably comes from the same part of us that is hardwired to communicate through language. Language, and communication in general, are pretty useful.
- Submission is… well, I'm not sure about that one. Maybe it *is* an effect of fun rather than the cause.

Discovering New Kinds of Fun

We can do this in reverse. Instead of taking something that's fun and tracing it back to the reptilian parts of our brain, we can isolate skills that our hunter-gatherer ancestors might have needed to survive, and then use that to figure out what we would find fun. For example, here are some activities that are often found in games:

- **Collection**. This is the "gathering" part of hunting-and-gathering, so you would expect it to be fun. And it is. When I was a kid, before video games became ubiquitous, the world's most popular hobby was stamp collecting. In many board games you collect resources or tokens. Trading Card Game players collect cards. In the video game world, we've been collecting things since Mario first started collecting coins.
- **Spatial Reasoning**. Primitive humans needed to figure out spatial relationships in order to build useful tools (for example, if you want to find a big stick to make a crude ladder or bridge, you need to be able to estimate length; if you want to stick two pieces of wood together, you need to be able to figure out how to make them fit). Many games make use of spatial relationships, from *Tetris* to *Pente*.
- **Advancement**. I see this as kind of a meta-skill, the skill of learning new skills, which is obviously useful to a primitive human that needs to learn a lot of skills. We see this formalized in games all the time, from the overt Experience Points and Levels to finding new items or buying new weapons that give us better stats or new capabilities.
- **Finding Shortcuts**. Finding novel, undiscovered ways to work around problems in ways that take less effort than normal helped primitive humans to conserve their energy; in that sense, laziness can be a virtue. Ironically, in games, this often takes the form of deliberate rule-breaking and cheating.
- **Griefing**. Like other forms of competition, putting other people down is a way to show dominance and superiority over your peers. (Yes, some of us find it annoying and immature, but cavemen are not exactly known for their emotional sensitivity.)

Perhaps you can think of other kinds of fun. Feel free to add to the list in the Comments on this blog post.

Games Change Over Time

Play in general, and games in particular, help us to exercise the skills we need for adulthood. While the things we find fun require millions of years of evolution to change, the games we play can change with each generation. As such, you can tell a lot about a society's values by looking at its most popular games. (A few centuries back when most people were farmers, grain harvesting was a big deal to a lot of people. Today it is not, so we do not see a predominance of "grain games" in our contemporary world.)

This gives yet one more potential starting point when designing a game. Think about what kinds of skills are useful in adulthood in your culture. Find a link between those and the skills needed for a primitive hunter-gatherer to survive. Then, design a game that exercises those skills. Most successful learning games do this, by integrating the learning *into* the game. The actions in the game either consist of using the skills that need to be learned, or the learning of a skill is the victory condition of the game. In both cases, the gameplay is aligned with the inherent fun and joy of learning, and you can end up with an "educational" game that is also fun. Note that this is in stark contrast to the typical "Edutainment" title that requires rote learning as a prerequisite for play, or that separates the learning and the gameplay, which has been proven time and again to be *not* fun.

One Problem

So, now it would appear we have all the answers. Flow states are pleasurable. We are driven by our hardwired tendencies to build useful hunter-gatherer skills. Games can exploit these to produce that thing we call "fun."

Is that it?

Well, no.

First, we must question our collective obsession with this "fun" business. Fun is not the only pleasurable emotion. For example, designers often talk of:

- **Fiero**, the triumphant feeling of completing a significant, challenging task. "You rock!"
- **Schadenfreude**, the gloating feeling you get when a rival fails at something. "Tragedy is when I stub my toe; comedy is when you fall off a cliff and die."
- **Naches**, the warm feeling of self-worth that you get when your child, student, or other person you are mentoring succeeds. "I'm so proud of you!"
- **Kvell**, the emotion you feel when bragging about your child, student, etc. "My kid is an honor student at Wherever Elementary."

None of these emotions would be described as "fun" exactly. None of them are directly related to flow states, either. But they are pleasurable. And they could certainly add something to a gameplay experience.

Also, as we discussed when talking about art games, "fun" is not necessarily the only purpose for which games could be made. We may read *War and Peace* and say that it is a good book, but we would not call it fun. We may say that *Schindler's List* and *Saving Private Ryan* are great movies, but people would look at us very strange if we said either one was fun. *Macbeth* is not particularly fun. Viewing the *Mona Lisa* is not fun. The daily news is rarely fun. And yet, these things can all be deeply meaningful.

A game reviewer might say of the *Mona Lisa*: "Great visuals, but only one level, low interactivity, not much replay value. Interesting, but not very fun. 2/10."

The rest of us would not.

So, that premise that I started with last Monday – "a game designer's job is to make a game fun" – is something that you should all be a bit uncomfortable with by now. Fun is certainly a strong component of many games, but games do not have to be limited to that. Our role as game designers goes *beyond* making a game fun. A game designer's job is **to craft a meaningful gameplay experience**.

Fun just happens to be a convenient and easy way to do this. But never forget that it is not the *only* way.

Another Problem

Koster points out in *A Theory of Fun* that players are, at their core, lazy. They tend to seek games similar to those that they're already good at, so they are not learning something that is new, which reduces the amount of learning-pleasure they can receive. They tend to look for loopholes, exploits, and cheats, which likewise circumvent the pleasurable learning process. Players make the game less fun – but they do it anyway.

In fairness, game designers do this too. We probably do this even moreso than most players, since we are so experienced at finding patterns in games and we see the forms so quickly. This leads to lots of derivative work. Personally, the first game I ever worked on was a collectible card game, and even now I instinctively want to add cards, custom decks, cost/benefit decisions, and the concept of rarity to every game I make. Another designer I know sees everything in terms of RPGs. Another one of my colleagues tries to turn everything into a Sim game. Most of us, I think, tend to think in terms of one genre even if we're working in another. In my experience, it's usually the genre of the first game we work on professionally.

Is there something about us that makes us like one kind of game over another? If it is as simple as "personal taste" then why do we see so much overlap among gamers?

Player Types

This brings us to Bartle and his player types. As with kinds of fun (and definitions of games), we find no shortage of people willing to advance their own theory of player types. Why read Bartle, then, and not someone else? First, Bartle's was the first essay of its kind to gain widespread interest and acceptance, so it is important historically; second, because there are certain aspects of it that make for interesting dissection.

Let us look at the four proposed types of players in a MUD (or MMO):

- **Achievers** find it enjoyable to gain power, level up, and generally to "win" the game (to the extent that an ongoing, never-ending game can be "won").
- **Explorers** want to explore the world, build mental maps of the different areas in their heads, and generally figure out what is in their surroundings.
- **Socializers** use the game as a social medium. They play for the interaction with other players. The gameplay systems are just a convenient excuse to get together and play with friends.
- **Killers** (today we call them "griefers") derive their fun from ruining other people's fun.

What is the motivation of each player type? Why do they do what they do? This relates back to the different kinds of fun.

Comparing the lists of Bartle's player types and MDA's 8 kinds of fun, we see parallels. Achievers favor Challenge fun. Explorers seem to like Discovery fun. Socializers are all about Fellowship fun. And Killers… well, they don't map to a specific kind of fun in MDA, but the Griefing fun that I proposed as an addition seems to work well.

Other player type schemes show similar correlations: each "player type" is really a kind of fun, or a combination of several kinds of fun, personified. The two concepts (player types and kinds of fun) are really the same concept expressed in different ways.

This suggests that you can start with a list of kinds of fun, and invent new player types based on some combination of fun types. Car racing games combine Sensation and Challenge fun; I could propose a "Racer" player type as the kind of player who likes these kinds of games. And then I could make a guess that other games, such as "Xtreme Sports," might appeal to the same player type since they have a similar "fun signature."

You could also go the other way. If you manage to isolate a new player type (i.e. a pattern of play that appears in a nontrivial percentage of your playtesters), by studying that type and what the players are doing, you may be able to discover new kinds of fun.

Which Comes First?

If we can go back and forth between player types and kinds of fun, we may wonder if this is a classic chicken-and-egg problem. Is it better to start with players, or fun?

Consider this: as game designers, we create rules (mechanics). The rules create the play dynamics when set in motion, and those cause the aesthetic of fun in the players. The things that we create, are a root cause of fun. Therefore, it is the *kinds of fun* that are of greatest concern to us.

We do not create players. (Well, those of us who are parents could say that they do, but you know what I mean.) As game designers, our rules do not create new players or player types. Therefore, any list of player types is only useful to the extent that it is correlated with kinds of fun.

Let me give an example. There is a book, *21st Century Game Design*, by Chris Bateman and Richard Boon, that proposes player types based on Myers-Briggs personality types. The main idea of doing market research, understanding the players that you are designing for, and designing a game to fit the target market is an idea that has definite applications in game design. But the implementation has a problem. Myers-Briggs types are mapped to player types, which in turn correspond to different kinds of fun. There are *two* levels of abstraction here, which means a higher-than-normal error rate. People do not always fall neatly and precisely into 16 categories, after all.

A more well-trod example is that of classifying players as “casual” or “hardcore.” Now we see why this distinction may be useful to marketing suits, but not so much to game designers. What kinds of fun correspond to these players? What is “casual fun” or “hardcore fun”? This is not clear. We are told that casual gamers want experiences that are short, easy to learn, not very challenging. Yet, some so-called “casual games” are difficult (*Diner Dash*), long (*Puzzle Quest*), or complicated (*Virtual Villagers*). Instead of spending time trying to define a single “casual gamer” archetype, I suspect it would be more fruitful to identify the kinds of fun that help a so-called “casual game” to succeed, and then work from there.

A Note for Teachers

As with last time, there are some direct parallels with teaching. Where I say “player types” and “kinds of fun” an educator might be thinking of “learning styles.” What I call Sensation, Narrative and Expression fun, you might refer to as Audio, Visual, or Kinesthetic learning.

Think of ways to apply this to your classroom:

- How many kinds of fun do you use in your classroom? Do you use a variety, in order to give all students a chance to be engaged and fascinated at least *some* of the time?
- Sensation fun is pretty easy. Bring things to class that are interesting to look at. Bring props that can be felt or passed around. I know one teacher who will get the entire class to stand up and stretch if she sees the students nodding off.
- Narrative is another easy one. Most subjects have stories embedded in them. It is much easier for most people to remember a story than to remember a random factoid. We’re hardwired to tell and to listen to stories.
- Challenge often comes in the form of quiz-show-type games in class. While *Jeopardy!* is still marginally more interesting than the average college lecture, keep in mind that students are not making any interesting decisions. You can do better than this. Formal or informal debates and discussions with students taking sides can also play to this kind of fun.
- Fellowship can happen in class when students are put in groups, or during class discussions.
- Discovery is difficult in most classrooms, as everyone is stuck in their seats and can’t explore the area much. Field trips are an obvious way to work on this. If your classroom is internet-enabled, you can ask students to do Web searches, at least letting them explore a virtual space if not a real one.
- Collection is a kind of fun that is most often seen in elementary school classrooms, giving students stickers or gold stars. It is riskier in higher education (you run the risk of treating your grad students like they were in kindergarten), but it can be done. I know an Economics professor, for example, who printed out a bunch of dollar bills with his face on them, and handed them out to students during in-class exercises, pop quizzes, and the

like. Students could exchange the play money for real cash and prizes at the end of the term.

- Advancement is a kind of fun that is inherent in any course where the later material builds on what was learned earlier. If you created a diagram of skills being taught in the class (with arrows drawn from the prerequisite skills to the new skills being layered on top), you might find that it looks a lot like a "tech tree" or "skill tree" in an RTS or MMO video game. By exposing this kind of skill diagram to the students (and then showing them when they gain new skills and "unlock" access to other more advanced skills) you can create a sense of accomplishment… and also make the connections between the topics easier to see. Incidentally, for department heads out there, you can also do this for an entire curriculum, diagrammatically showing the course requirements and prerequisites.

Lessons Learned

In general, the things we find fun are related to the skills our distant ancestors needed in order to survive. We can exploit this in our game designs to make games that are more fun.

Some people find certain kinds of fun more interesting and engaging than others. Tastes vary. Try looking at your own favorite games (and popular games that you *don't* like) and see if you can discover your own personal "fun signature."

Remember that "fun" is but one of many emotional responses that a game can invoke in a player. Our goal as game designers is to deliver a compelling experience, which may or may not be fun. Most of the art games in Level 6 were not particularly fun… but they were deeply meaningful. Fun is an important part of what we do, but do not seek fun at the exclusion of all else.

As you do your own research, you will undoubtedly run into many articles that purport to classify Fun or Players into types. Do not take any such article as gospel. Instead, analyze it to see if it makes sense. For fun types, can you see *why* we (or our hunter-gatherer ancestors) would find each type fun? For player types, can you see a link between player types and kinds of fun, since it is easier for a game designer to create a custom brand of fun than to create a new type of player?

Feedback

If you have time, before beginning the Homeplay below, please take the time to offer **constructive feedback** to at least two other posts at your skill level from the War challenge from Level 7 (posted on the forum), and also at least three other posts at a skill level below yours (unless you posted in Green Circle).

Try to complete your feedback on or before Monday, July 27, noon GMT.

Homeplay

Let's practice our ability to find mechanics that give rise to a specific kind of fun. The kind of fun we'll be considering here is **griefing** (that is, deriving enjoyment from the act of ruining other people's enjoyment).

Create a concept for a game that is built to appeal specifically to griefers (*i.e.* Bartle's "Killer" player type). Post it in the forums. **You do not need to create the rules for a game.** Just create a concept that includes enough information to communicate what the gameplay would be like

Your post should include the following:

- Target medium or platform. Is this a PC game? Console? Board or card game? Tabletop RPG?
- Number of players
- A paragraph or two describing the game concept
- Another paragraph explaining why this concept will appeal to the target market.

Post in the forum that most closely resembles your skill and experience level as a designer:

Beginner, little or no experience prior to taking this course.

Intermediate, some coursework or exposure to game design but little or no professional experience.

Advanced, at least some professional experience as a published game designer.

Make your post on or before Monday, July 27, noon GMT. Then, **offer constructive feedback** to at least two other posts in the same forum, *and* at least three posts in one forum "below" yours if you posted in Blue Square or Black Diamond, before Thursday, July 30.

Mini-Challenge

Can you think of a new kind of fun? Describe it and justify why it is fun in 135 characters or less. Post it to Twitter with the #GDCU tag. Offer as many as you can think of. Tweet before July 27, noon GMT.

Level 9: Stories and Games

By ai864

Up until this point, we have talked about games from a purely *ludological* viewpoint. That is, we have looked at games as a system of rules, with the implicit assumption that the rules *are* the game, and that a narrative of any kind is just window dressing. (Any word with the root *lud-* or *ludo-* is referring to games; the root is Latin for *play*. We use words like *ludology* and *ludography* and *ludic* because everything sounds more distinguished if you say it in Latin.)

But this is not entirely true. As mentioned when we talked about kinds of decisions, some player choices may have absolutely no meaning within the game system and yet they are still meaningful because they are emotionally charged.

Those of you who play tabletop Role-Playing Games are probably more keenly aware of this. Think of the most interesting play session you've ever had. You're probably not thinking of a die roll, or an interesting strategic decision that a player made in combat. You're remembering something dramatic, emotional. You remember the story, not the die-rolling.

What makes for good stories? Game designers often reference three works in particular that tell us how to make useful stories that apply directly to games. If you're curious, these works are:

- *Poetics*, by Aristotle
- *The Hero with a Thousand Faces*, by Joseph Campbell
- *Story: Substance, Structure, Style and the Principles of Screenwriting*, by Robert McKee

Today we will look at these works and their effect on game design. We will build up a set of guidelines for how to tell a good story within a game. And then, at the end, we will tear it all down again.

Course Announcements

As I've been out of town and offline since early Friday morning, I have not had time to validate new user accounts for the forums, read email, moderate comments on this blog, etc.

I will catch up on these things later today, or tomorrow morning at the latest, after I have fully recovered from a nearly-sleepless (in a good way) weekend. I'll say this: playtesting your games with skilled game designers is very different from playtesting with typical gamers.

Mini-Challenge Results

Here are some new kinds of fun that were proposed from last time:

- "Servant": opposite of a griefer, someone who gets pleasure out of making sure other people have a *good* time. (I would actually call this something else, like "Party Host"… and yes, it makes sense that this would be valuable to a hunter-gatherer. You are showing value to your tribe. I understand that Disney has identified this as a customer archetype, usually associated with the mom in the family.)
- "Maker": someone who gets joy out of constructing and building things. (The example given was user-generated content for video games, but you sometimes see this in standalone board games as well, like *Settlers of Catan*. Certainly, building and construction are useful to survival, even if all you're making is a tool or a crude hut.)

Readings

Read the following:

- *Into the Woods: a Practical Guide to the Hero's Journey* by Bob Bates. This article summarizes Joseph Campbell's work, as it is relevant to game design.
- *What Every Game Developer Needs to Know about Story*, by John Sutherland. This article summarizes the book *Story* by Robert McKee (which itself is essentially a practical guide to Aristotle's *Poetics*), rounding out the Holy Trinity of storytelling for game designers.
- *Understanding Comics*, Chapters 2 and 3, if you have a copy of that book. As you read, pay particular attention to how any of this might apply to games. Scott McCloud isn't going to come out and say it, so you will have to connect the dots yourself.

Aristotle

A lot of words were different in Aristotle's time than how we use them today. *Poetics* is not about poetry, but about how to write tragedy. *Tragedy*, as Aristotle used the term, did not mean "a story with a sad ending" but rather a story that is serious and lifelike – a story devoid of the supernatural or fantastical (which he referred to as *comedy*).

However, one thing that hasn't changed in all this time is that there is still a lot of really bad writing.

Aristotle may not have been the first to notice, but he was certainly one of the first to actually do something about it. He wrote about how to write a decent story. If a lot of his advice sounds familiar, it is because it is often repeated in writing classes, even at the elementary school level – although Aristotle may or may not be credited for the idea in any given class.

For example, have you ever heard that stories should have a beginning, a middle, and an end? That was from *Poetics*. It is a reminder that there are different parts to a story, and that the writer should be aware of how it all fits together.

Poetics also defined what is known as the *three-Act structure* for stories, basically a division of a story into three parts. In the first part, something happens to set the events of the story in motion. In the second part (which tends to be the longest), the protagonist tries to deal with the events as they happen. In the final part, a resolution is reached. (I've heard it described thus: in the first act, get the hero up a tree; in the second act, throw rocks at him; in the third act, get him down.)

One important thing that Aristotle really hammered on is that each scene should follow the previous ones with a logical cause-and-effect relationship. Weak writing goes like this: "X happens, then Y happens, then Z happens." Stronger writing is more like this: "X happens, and because of that Y happens, and because of that Z happens."

This cause-and-effect rule is even more restrictive when it comes to the protagonist. When bad things happen to the main character, it should not be random; it should be caused by that character's understandable human action, and it should follow as a plausible and inevitable effect of that action. This makes the audience pity and empathize with the hero, because we can see the human weakness, we can understand *why* the character did what he did, and yet we also see that it causes his undoing. This explains why Aristotle really hated what was called *deus ex machina* (that is, an ending where everything is suddenly made better through no fault of the main character – for example, "…and just as the main character was about to die, he woke up, and realized it was all just a bad dream, The End"). In a *deus ex machina*, the hero is not the cause of the ending. The main character is not in control of the story.

Applying this to games, it becomes clear why it is sometimes so frustrating when, for example, a character in a video game dies during a cut scene. The one time the player doesn't have any choice – the one time when the main character is *not* in control – is the one time when the plot advances.

Lastly, it is worth mentioning that Aristotle defined a stage play as being comprised of six elements. We have similar elements in games with a strong story component:

- **Plot**. The narrative that describes what actually happens.
- **Theme**. What does it all mean? *Why* does it happen?
- **Character**. As in, a single role within the story.
- **Diction**. The dialogue, and also the actor's delivery of that dialogue.
- **Rhythm**. This does include "rhythm" in the sense of music, but also the natural rhythm of human speech.
- **Spectacle**. This is what Aristotle called the "eye candy" or special effects of his day. He often complained that too many plays contained all spectacle and nothing else – sound familiar?

McKee

I'm not sure if Robert McKee ever actually wrote a screenplay that was made into a movie. Mostly, he teaches screenwriting. If you've ever come out of a movie saying "wow, that was a really great *story*," the screenplay was probably written by one of McKee's students. (I would love to be considered the "McKee of games" some day. Note to my former students: go out there and make me look good!)

Story is essentially a re-telling of *Poetics*, but made specific to screenwriting for movies. I found *Story* to also be a lot more accessible to read; it is written in a conversational style (not to mention that it is written in contemporary English and not ancient Greek). To paraphrase a few of the many lessons from McKee's book:

Story is not about formulas, it is about forms. You do not create a story by following a template. However, by understanding the common links between different stories, you can make one that is unique. (I would add that the same is true for everything in this course.)

All stories have this form:

- The protagonist has a goal, which is created by an *inciting incident*.
- The protagonist tries to reach the goal, but a *gap* (that is, some kind of obstacle, not necessarily a literal gap) opens up and prevents the immediate achievement of the goal.
- The protagonist attempts to cross the gap. Either the gap widens and they are unable to cross, or they *do* cross the gap but a new gap appears.
- This cycle of gap-crossing continues until the protagonist either finally completes the goal, or is prevented from completing the goal in an irreversible manner.
- In a typical three-Act structure, there are two *reversals* (new gaps) that happen between the Acts.

Stories are, at their heart, about change. Every scene should change something, or have something unexpected happen. If a scene has the characters in the same state at the end as it was in the beginning, that's a sign that you should remove that scene. Think of it this way – if you were to convert your life into a two-hour movie, would you waste any screen time on your day-to-day maintenance tasks? Or would you only show the times when something big changes in your life, and allow the audience to assume that things are carrying on normally in between?

Notice how nicely this dovetails with games. Games are about decision-making, which causes a change to the game state. Games rely on having an uncertain outcome, and it is only at the very end that a goal is attained or lost in an irreversible manner. It is not surprising, then, that some games have very strong emergent stories that arise from a particular play experience.

Another interesting thing McKee talks about is the difference between what he calls *character* and *characterization*. The things we normally think of when we define a "character" are superficial data: favorite food, blood type, hair color, and so on. McKee calls these *characterization*. Character is what defines the person – used in the sense of "this activity builds character" or "she has a strong moral character." What McKee says is that *character* can only be revealed by putting a person in opposition. For example, we may say that someone is "selfless"… but until they're in a burning building and have to make the choice between trying to save a total stranger or saving themselves, it's all just talk.

What is the implication of character and characterization in games? First, that linear stories have the best opportunity to show character through cut scenes, not gameplay. Having the player make moral choices for the main character is hard, because the choices often don't involve real consequences. Because this is play ("only a game," the "Magic Circle"), the player is safe, and therefore has nothing in their own real world to lose. The player is therefore not making choices that reflect their own character, because their character is not being tested by extreme opposition. Taking a bullet for a friend in the real world is not quite the same as deciding in a menu whether or not to gain Light Side Points. It is certainly not impossible to embed moral dilemmas in a game, but it is a lot harder to make the emotional consequences of those choices felt by the player, because the player is making those decisions and not the protagonist. It is therefore much easier to show strong *character* when the player is not in control of the story.

But of course, that also makes it less interactive and thus less like a game. And this is one reason why storytelling in games is hard.

Joseph Campbell

Joseph Campbell spent a lot of his time studying myths, legends, and hero stories, and finding the similarities and differences between them. He found that most myths follow a common structure, which he called the *Monomyth* or the *Hero's Journey*. It is a specific kind of story and therefore more specific than McKee's story description. Because many games put the player in the role of a hero, this is obviously useful to know.

The Hero's Journey goes something like this:

- The hero starts off a commoner in a common world, and this "normal" world is established.
- The hero receives a call to adventure.
- The hero may decide to follow the call, or to ignore it. In the latter case, new events then force the hero to follow the call anyway.
- The hero starts their journey and encounters the first barrier. There is often a guardian that must be overcome to proceed.
- The hero then moves through the barrier into a new, darker world. They follow a trail of trials, each more difficult than the last. Along the way, the hero grows – not just in the "experience points" and "levels" sense, but in the "coming of age" sense. The hero becomes a better person. They become, well, a real *hero*.
- Eventually, the hero encounters the final evil, and is able to overcome it.
- The hero claims the prize.
- The hero starts returning to their world. Along the way they encounter the final barrier.

- Finally, the hero returns to their common world. The world may be the same, but the hero has changed.

You may recognize this structure in many hero stories, and Campbell's book goes into detail about *why* each of these things happens, what it symbolizes, and what it says about our values as a society. In short, hero stories are about what a particular culture sees as the ideal set of ethics and values, and the hero character embodies and demonstrates these things.

Now, you might be tempted to use this as a formula. Get a list of archetypes with a checkbox next to each, and *presto*, you now have a suitable story! Unfortunately, it's not that easy. As McKee says, stories (and hero stories are included in this) are not about formulas, but forms. The purpose here is not to follow the Monomyth blindly.

What use is it, then, if we cannot use this to make a story? I think the most important thing to take from this is to be aware of what the common story forms *are*, so that you call follow each step *or not* as appropriate to your own story. But, it is important to do so *deliberately* and not just "because Campbell said so." Note that not all games follow this structure – especially games where you play an anti-hero.

Bob Bates comments on the structure in his article:

- When writing, start with a core premise or vision first. Choose a hero and villain that embody your premise.
- Show the hero's common world, then disrupt that world through an inciting incident. This is typically what happens at the beginning of a game.
- Enter the "woods" – the game itself.
- "Encountering the evil" is essentially a description of a boss fight – suggesting why we see so many boss fights in games!
- "Claiming the prize" can be thought of as the hero realizing the Premise of your story. It does not have to be finding a literal "prize" like a bag of gold or a princess or an ancient magical artifact.
- During the game, the hero character should grow. Again, it is easy for us as designers to fall into the trap of *only* having the main character "grow" in terms of power level (and it is convenient that the *player* is growing in their skill at the game as they play). Still, it can often make a better story if the hero's *character* grows during the story as well. They don't have to start out as a god. It can be more interesting if they start out as a peasant and *become* a god. Remember, it's the *hero* that must grow, not just the *player*.

Scott McCloud

Understanding Comics doesn't say a lot about telling stories in Chapters 2 and 3, but it does give some useful advice on creating strong characters and dramatic moments.

On pages 44-45, McCloud notes that art styles can vary between iconic (like a smiley face) and photo-realistic, with many potential steps in between. He points out that the more iconic something is, the more we project ourselves onto it; the more detailed and realistic, the more we see it as something *other* than ourselves. (Taking it back to Koster, we can say that this is because our brains are wonderful pattern-recognition machines, and we will fill in the blanks with what we already know from the vast library of patterns we've built up.)

What are the implications of this in games?

- Consider the main characters in many video games – Master Chief, Samus Aran, Gordon Freeman, Chell. You do not typically see your own character much at all, nor do you hear them speak much. This is not an accident. It is done deliberately to allow the player to project their own personality onto the character. The character becomes an extension of you as the player, and you feel an emotional connection to the character specifically because they are *not* very well defined.
- On the other hand, you can also have a strong character that is *very* defined – Duke Nukem or Lara Croft, for example. In this case, we immediately recognize the main character as *not* ourselves. To compensate, they must show a strong personality.
- In general, then, I would say that you can go one of two ways with the main character. Make it iconic and do not define its personality (to allow the player to create one for themselves), or make it realistic and define its as a very strong character. Any other combination makes it harder for the player to connect emotionally with their avatar.
- Also, consider the enemies and opposition within the game. Since realistic visuals impart a sense of otherness, enemies that are highly detailed will seem very alien, while enemies that are cartoony or iconic feel more familiar. The monsters in the video game *DOOM* are drawn in a realistic style, making them seem more alien and thus more dangerous. By contrast, the monsters in *Pokemon* are cartoonized, making them seem more friendly, which is fitting for a game where you can recruit enemies and turn them into allies. In board games, we would expect that games with iconic tokens (like colored pawns) that represent players make the pawn into an extension of the player (a sense of familiarity), and also that other players' pawns have a sense of the familiar – it promotes togetherness. By contrast, games with highly detailed tokens (realistic miniatures, or detailed art or photographs of player characters with in-depth character descriptions) gives a sense of separation between player and character, and also would cause players to regard each other as opposition.
- This also has applications when dealing with environments. If the environment (whether a 3d computer level or a 2d physical game board) is photorealistic, it is a reminder to the player that this is an *other* world. This is more suitable for games that wish to make the players feel like they are in an exotic or unsettling location. For example, suspense and horror games would do well to include highly photorealistic environments.

Another point that McCloud makes (on page 38) is that we are made to use tools, and we see those as an extension of ourselves. Our sense of self extends not just to our own bodies, but to everything under direct control. As he points out, when you are in a car accident, you are more likely to say “hey, they hit me” than “their car hit my car.” It becomes personal.

What does this have to do with games?

- For video games, a console controller (or mouse/keyboard) becomes an extension of the human body. The player thinks of the controller as part of themselves. This explains why play control and a good user interface is so important for video games – if you have trouble figuring out how to use the controller, it is just as frustrating as if you tried to pick something up with your hands but your hands didn’t respond.
- For both video games and tabletop games, the avatar (that is, the representation of the player within the game) acts as an extension of the player as well. As with an auto accident, if your opponent lands on your pawn and sends it back to start, you are likely to say “hey, they just sent me backwards.” As a designer, be aware of the player’s emotional attachment to their avatar within the game.

The last thing I’d like to draw your attention to is McCloud’s concept of the “blood in the gutter” (pages 66-69). In the book, there are two panels, one with a murderer swinging an axe at a victim and then the next that just shows a scream. When did the guy die? Between the panels… and it was *you* as the reader, with your imagination, that killed him. Nothing was actually shown.

This has implications in all other kinds of storytelling media. Alfred Hitchcock was a master of *not* showing anything. As an example, in the famous *Psycho* shower scene, you actually never see anything. There is one shot of a guy making a stabbing motion with a knife (but not showing any victim), juxtaposed with another shot of a woman screaming (but not showing her being stabbed), back and forth, and eventually a shot of fake blood running down a drain (without showing either the murderer or victim).

How do we apply this to telling stories in games?

- Some storytellers have a strong desire to give every last technical detail of how everything works and every last bit of backstory in their fantasy world. But this is not necessary. Players will fill in the blanks on their own. You don’t actually have to tell them anything.
- In fact, it is often more effective if you don’t! A player’s imagination is infinitely more vivid than the artwork in your game.
- Think of the player as an active participant in your story. They will be thinking about it anyway; write a story that rewards them for using their imaginations.
- This also has an economic advantage. We tend to pour a lot of money into detailed art and long, drawn-out cut scenes, but if we economize and show less, the net effect can actually be more powerful if we do it right.
- In other words… less can be more. Finding the balance between “enough information to understand what is going on” and “not so much information that nothing is left to the imagination” is one of the trickiest jobs of a story writer, and is another reason why storytelling in games is hard.

- Think of some examples of stories you've seen (from games or otherwise) where there was too little information, or too much, and the story suffered from it. Think of other examples where you were not told everything, but was fine, and the audience was able to still have an enjoyable experience.

Ernest Adams

Game designer Ernest Adams gave an inspiring talk at GDC 2006 called "A New Vision for Interactive Stories." First he briefly summarized much of what I have written above, and then he proceeded to challenge all of our basic assumptions, and then he tried to take things one step beyond. What follows are my notes and personal commentary from the session.

- Aristotle's *Poetics* is great, but never forget that it was written for stage plays and not games. Stories may have a beginning, middle and end… but in games, they can often have *multiple* beginnings, and middles, and ends. The three-Act structure works great for a two or three hour play (or movie), but is not necessarily appropriate for a 30-minute board game, a month-long RPG campaign, or a 100-hour console game.
- Campbell's *Hero's Journey* is limited to hero stories. What if your story isn't about a hero? Also, as Campbell admits, the Monomyth is not a template, so we cannot use it as a tool to build our stories.
- McKee's *Story* is focused on screenplays, so it may or may not be applicable to every game. Games are a different medium than movies. While there are some similarities, it is important to be aware of the differences, so any advice on screenwriting must be used with caution when applied to games.

So, if none of this stuff is useful, are we back to square one? (I don't think so. We still have to start somewhere, and starting by studying what makes a great story in other media is still a useful starting point. We will get into the unique forms of interactive stories this Thursday.)

Adams then stated three assumptions that we often make when trying to tell stories in games:

1. The "holy grail" of interactive stories is a complete sandbox, a "Holodeck," a perfect world simulation that responds believably to all player input.
2. Interactive stories aren't games.
3. When a player is involved in an interactive narrative, they should be thinking about story and not game mechanics.

He then challenges these assumptions.

First, what could be wrong with having a perfect world simulation? There is always the practical reason that it would be infinitely expensive. And then there's the argument from Koster that we already have one of those, it's called the Real World, and it's not always fun. But mostly, the problem here is that even in the most "open-world" games, players do not get their enjoyment from complete freedom… but rather, from having freedom within a constrained environment.

Ernest proposed a rule from another designer, which her referred to as “Ken Perlin’s Law”: **the cost of an event in an interactive story must be directly proportional to its improbability**. What does he mean by “cost”? He explains that every writer has a “credibility budget” – and if too many incredible things happen, you violate suspension of disbelief. The cumulative sum of all improbable things that happen during your story need to not exceed a certain amount, or the players will call foul. (Naturally, some games have a higher credibility budget than others, based on their setting – chickens appearing out of thin air may be mundane in a high-magic world, but would be considered out of place in a realistic modern-day setting.)

As a designer of an interactive story, you are essentially making a pact with the player: if you (the player) act believably, you will get a believable story. This is important – both the designer *and* the player share the credibility budget. The player must accept the premise of the story as part of stepping into the Magic Circle to play. If the player acts in a manner that is inconsistent with the world of the story, and gets an unbelievable story back, that is not the fault of the story writer; it is the fault of the player. As such, it is *not* the goal of the story writer to create a 100% believable story in all instances; it must merely respond believably to a player who acts in a believable manner.

As we saw from Doug Church’s *Formal Abstract Design Tools*, there is a balance between player intentionality and narrative. However, we can extend this through the social contract of “role-playing” (in the sense of actually *playing a role*, not crawling through dungeons) between the player and the designer.

Of course, in order for the player to accept this contract, they must be aware of the rules of the game, and they must agree to play by those rules. In this sense, the rules are an important component of the game, but the interactive story and game are also linked together in a way that makes the experience both game *and* story.

A way to merge games and stories. That is what many of us are looking for, is it not?

Lessons Learned

Aristotle, Campbell and McKee provide some of the most often-cited advice for storytellers in general, so it is natural that we apply their advice to games. For those of you who are primarily interested in this aspect of games, I would highly recommend reading their books on your own time (after this course is complete, of course). You can find them here: Aristotle, Campbell, McKee. **I provide these links for convenience only; they are not required for this course.**

In games, identification between the player and their characters, avatars, tokens and so on is a common way to get players to be emotionally engaged with the game. As you are designing a game, think about this and how else you can get emotional investment from players.

Remember that there is a difference between the embedded narrative that the story writer creates, and the emergent narrative that arises from gameplay. Think about which is more important in each game that you make, and how you can make it stronger.

Feedback

If you have time, before beginning the Homeplay below, please take the time to offer **constructive feedback** to at least two other posts at your skill level from the Griefing challenge from Level 8 (posted on the forum), and also at least three other posts at a skill level below yours (unless you posted in Green Circle).

Try to complete your feedback on or before Thursday, July 30, noon GMT.

Homeplay

The point of this homeplay is to give you some experience understanding the relationship of story to game mechanics, and what happens when the two are or are not aligned.

We start with a simplified version of the abstract game of *Pente*:

- Players: 2
- Objective: place your stones to either create five-in-a-row, or to capture five pairs of your opponent's pieces.
- Setup: place a grid-shaped board (you can use a Pente board or a Go board, or make one of your own – try making it 19×19) on a table between the players. Choose a player to go first.
- Progression of play: on your turn, choose a blank square and place your marker in that square. (You can use colored glass stones, or you can just write "X" and "O" on a piece of paper as with *Tic-Tac-Toe*.) If there are *exactly two* opposing markers in a straight line (orthogonally or diagonally) adjacent to where you just placed, *and* on the other side of the two opposing markers in the same line there is one piece of your own, then the two enemy markers are captured. Remove them from the board (erase the symbols if playing with pencil and paper), and put them off to the side to denote that you have made a capture. It is possible to make several captures on a single turn if there are several sequences of two-enemy-one-friendly radiating out from your placement in multiple directions.
- Limitations to capturing: captures *only* take place when a piece is placed. It is legal to move into a place that causes an "X-O-O-X" or "O-X-X-O" line on the board, by placing in the middle. In such a case, the inner pieces are not captured.

- Resolution: If a player ever gets five of their own pieces adjacent in a straight line (orthogonally or diagonally), they win. If a player makes a total of five captures, they also win.

If you have not played this game, you may want to play a couple of times (either with a friend or just against yourself) to get a feel for it.

When you are familiar with the game, **create an embedded backstory** for the game. What is the setting? What do the pieces represent? Why are you placing them? Try to come up with a story that fits the mechanics. **Do not change any rules.**

Next, **play the game** (with unmodified rules, but with your narrative) with a friend. Note their reaction to the game.

Post the following to the forum:

- Your backstory, as written.
- Your experience when playing the game with the backstory. Did your story make a difference? Did it affect the play experience? Or was it exactly the same as if there were no story at all?
- Why do you think you got the reaction you did? Do you think it would have been different if you had chosen a different story?

Post in the forum that most closely resembles your skill and experience level as a designer:

Beginner, little or no experience prior to taking this course.

Intermediate, some coursework or exposure to game design but little or no professional experience.

Advanced, at least some professional experience as a published game designer.

Make your post on or before Thursday, July 30, noon GMT. Then, **read at least five other posts** in the same forum, and five more in the skill level **above** yours (unless you posted in Black Diamond). **You do not have to respond**. What I want you to see is the variety of responses that people will have. Do this reading before Monday, August 3.

Level 10: Nonlinear Storytelling

By ai864

Last time, we learned some basic linear storytelling principles, as told to us by people that worked with books, plays, and movies. And this is fine and good for games that have a linear story. Many video games work this way, where the story is essentially told as a movie broken up into small parts, and the player has to complete each section of the game to see the next bit of movie. I do not mean this in any kind of derogative way; many popular games work like this, and many players find these games quite compelling. Even personally, I have had times when I would be messing about in the subscreens optimizing my adventuring party, only to have my wife call from across the room: "stop doing that and go fight the next boss so you can advance the plot, already!"

However, not all games are like this. As we've seen, player decisions are the core of what makes a game. Some games have a strong narrative intertwined with gameplay. For these games, it would make sense for player decisions to not only affect the mechanical outcome of the game, but to affect the narrative as well.

For some game designers, a true "interactive story" is something of the Holy Grail of games. In reality, we often fall short of giving the player the feeling that they are actually the starring role in a compelling story of their own creation. How have we told interactive stories in the past, and how can we make better ones in the future? It's largely an unsolved problem, but we can at least cover the basics of what is already known, and that is our focus today.

Note that most board games do not have strong embedded narrative, so this entire discussion is *mostly* relevant to video game narrative, as well as tabletop RPGs. However, some modern board games are in fact attempting to combine the tabletop board game experience with that of the RPG, including story elements that the players must interact with as part of the gameplay.

Course Announcements

I will be at SIGGRAPH all next week. While I will make every effort to post on time, I may not be able to respond quickly to email and I may be slow in validating blog comments or new forum accounts, so please be patient. Naturally, if you are attending there yourself, feel free to say hello in person – I'll be speaking on Monday morning about the results of the Global Game Jam.

Readings

Read the following:

- *A Point of View*, by Noah Falstein. We talk of the point of view of a story as "first person" or "third person" – and we also talk of the point of view of a video game in similar terms

(First Person shooter, Third Person stealth, etc.). It is easy to assume that the two are equivalent, but they are not. This article makes the differences clear.

- *Diversity in Game Narrative*, by Chris Bateman. This is a brief overview of the different kinds of story structures encountered in games.
- *Challenges for Game Designers*, Chapter 13 (Designing a Game to Tell a Story). This short chapter covers today's topic and can also serve as a review of some topics from last time.

Kinds of Stories

We can roughly classify different stories by their overall structure. The structure is determined by what kinds of choices are available to the player, how open or constrained those choices are, and what effect those choices have on both the ongoing story and the final ending. Each structure has its advantages and disadvantages, which we will discuss below.

Linear

Linear stories are the traditional narrative, with some gameplay elements thrown in that do not affect the story. In this case, the story and gameplay must be separate entities, because the story has no choices and the gameplay must include decision-making of some kind (else it is just a story and not a game). They can be thematically linked, and the story may influence gameplay (perhaps when a certain pre-scripted story element happens, it causes a new gameplay effect to come into play), but the gameplay cannot influence the story because there is only one story.

Linear stories have one major advantage over all other story structures: it is easy to apply traditional storytelling techniques, which have been developed over thousands of years. Such a story can have a powerful emotional impact – witness that we *still* talk of Floyd's death in *Planetfall* and Aerith's death in *Final Fantasy 7* as key moments in the history of game narrative… even though neither events was under player control.

Linear stories have the obvious disadvantage that, due to a lack of decisions, they are not very game-like. As stated above, there is a natural barrier between linear stories and game mechanics, which limits the effect the story can have.

Branching

The first and most obvious thing to do to a linear story, if we want to add decisions, is to add choice points at various places. When the player reaches a certain point, they decide what to do, and then the story goes down one of several continuing paths until another choice point is reached. An example is the old-school *Phantasy Star III* for Sega Genesis; twice in the game, the main character is given a choice of which of two girls to marry (and then the story continues to the next generation of characters), leading to a total of four branches, each with its own story and its own ending.

Branching stories have the advantage of being interactive. If you include a sufficiently large number of choices and your choices cover all of the things that a player would want to do, the game can respond believably to any number of player decisions. At first, this would appear to be the ultimate solution to game narrative since it can handle just about anything.

However, there is one major drawback of using a branching story: it is expensive! With only two choices (which is not very many), the story writers of *Phantasy Star III* still needed to write four stories. A third choice would have had them writing eight stories, and including a mere ten choices would require writing 1024 stories! Consider the number of decisions you make as a player in a typical strategy game, and you'll see that the amount of work to write a branching story can quickly explode into something unmanageable.

To make things worse, note that a player that goes through the game once will never even see the vast majority of content. It requires multiple replays just to see every path through the story… and even then, the player must decide which is the "real" story and which are the alternate timelines that didn't actually happen. (If the developers ever make a sequel to the game, they must deal with this problem as well.)

Parallel Paths

This is Bateman's word for a branching story that collapses in on itself, allowing the player to make choices but then collapsing all of them eventually into several mandatory events. In *Silent Hill*, for example, there are several choices the player can make that may advance the story or reveal some additional story elements along the way, and these will influence the ending. However, there are certain events that the player is forced to encounter no matter what else they have or haven't done.

Parallel paths solve the problem of branching narrative by keeping the advantage of player decisions while still keeping the total amount of story manageable. So, at first, *this* would appear to be the ultimate story structure.

As you might guess, there is still a problem: since the player is forced into certain events, the entire plot arc is now essentially linear again. We have lost the player's feeling that they are directing the story, because no matter what they do there will be certain parts of the story that are the same no matter what.

One potential solution is to have the player decisions alter the game's ending. The player may still encounter the same plot arc, but the final outcome is determined by the choices the player made. Unfortunately, that means the relationship between cause and effect can be easily lost –

the player's decisions are (by definition) not seen until the very end, and it is often unclear what the player did to cause a certain ending.

And, we still have the problem that the player must replay through the entire game multiple times just to see all the endings.

Threaded

This is Bateman's term to describe stories that are divided into small pieces, perhaps with several plot arcs going on at once that may or may not intersect. The player then chooses which paths to follow and in what order. One example of this is *World of Warcraft*, where the player can accept any of several quests to advance different elements of the story. Another example are the *Elder Scrolls* series of games (like *Morrowind* and *Oblivion*), where the player may follow certain storylines based on how they want to advance their character, and the player may find other quests or subplots that they choose to pursue (or not) along the way, and these subplots may or may not have their own effects on the main plot line.

The advantage of a threaded narrative is that it is extremely expressive. You can have multiple storylines happening concurrently, as with nonlinear films like *Pulp Fiction* or *Love, Actually* (except, unlike those movies, have the plot lines advance according to player intent).

Also, the story may have multiple beginnings and middles and endings, but the player has access to all of them and can advance any combination of them in any order, so we have finally solved the problem of forced replays. The player can see everything there is to see with a single play-through, if they are thorough enough.

As you might guess, there is still a problem here. First, since some events may affect others, testing out all possible paths through the story can get even more complicated than with a branching narrative. (For programming geeks, a branching story with two choices per choice point is O(N^2), while a threaded narrative is potentially O(N!). Yes, *factorial*.)

Writing a threaded narrative is hard, because events can happen in any order, leading to the potential for the player to do things in an order that doesn't make sense (for example, perhaps they are given a quest to assassinate a rival leader in the middle of a war, *before* the war actually breaks out, or *after* the war is concluded). The story writer must be careful to allow access to certain story events only when it would make sense to do so. Keeping track of all the variables that determine when a story event is or isn't active can get very complicated very quickly.

Lastly, a threaded narrative runs the risk of confusing the player, if there are too many concurrent plots happening at a single time and the player does not immediately see the relationships between them. This is also a danger with books and movies that try to tell several stories at once.

Dynamic Object-Oriented Narrative

This last structure is Bateman's term that, as far as I can tell, he invented to describe the game *Façade* (which you should absolutely download and play if you have not yet seen it). The idea is that there are several mini-stories, each with potentially several entry points and exit points. A single mini-story's exit point may lead to a final ending, *or* to another mini-story. The mini-stories may be thought of as "chapters" in a book or "acts" in a play (except that you may not "read" all of the chapters or may read them in a different order, depending on the choices you make and how you exit each chapter).

This kind of story has the advantages of parallel paths, but without a linear story arc. Each mini-story has its own choices, and the overall collection of mini-stories itself acts like a larger branching or parallel path story. Each individual mini-story is self-contained, which reduces the required time to write the complete story.

This kind of story has two disadvantages. The first is that there's still the forced-replay problem: a player must play many times to see all of the story paths (which is perhaps why *Façade* needs to last about ten or twenty minutes, and not ten or twenty hours). It is also a highly experimental structure, so we do not yet have enough games to really analyze what does and doesn't work in this form. *Façade* itself took a couple of guys with PhDs in Computer Science to develop, so this is not the kind of story structure that is easily accessible to a traditional story writer.

Points of View

The camera in video games is generally either first-person or third-person.

Camera Views

A first-person view means the camera is, essentially, glued to the main character's forehead looking forward. The player sees what the character sees. This can lead to a greater sense of connection with the main character, because the player is inside the character's head, so to speak. The disadvantage is that this is not *truly* a first-person view, because a typical human's peripheral vision is wider than the screen (and a typical human can turn their head faster than a typical first-person video game camera), which can break immersion at times for players who are not used to the limitations.

In a third-person camera view, the player is instead looking *at* the main character, usually from a behind-the-shoulder perspective so that you usually see the character's backside. This gives a wider view of the area and is more realistic in terms of giving the player the visual information that the character would have. Of course, by looking *at* the main character all the time, it brings a sense of otherness – you can't look at your *own* backside without the aid of mirrors, so this camera view is a reminder that the main character is not you.

Combining this with McCloud's point on icons versus realism (from last time), we might guess that a first-person character is closer to an icon and tends to do better as a "blank slate" character that the player can project their own personality onto, while a third-person character that is drawn realistically should have strong characterization.

A second-person camera view, if it existed, would involve the camera that spends the entire time looking at the main character from the front. Needless to say, this would be confusing in the vast majority of games. The closest I've seen is an obscure console title, *Robot Alchemic Drive*, where you control a human character (in third person) that is in turn controlling a giant robot by remote control (in second person). To the extent that the robot is the main character, this game uses a second-person camera for control.

Story Views

We also use the words "first person" and "third person" when discussing literature. There, it works a bit differently.

A first-person story is when the story is told from the narrator's own point of view. The main character is addressing the reader directly, telling you a story of what happened to them.

A third-person story, which is more common, is when the story is told from an outside perspective (much like a typical movie camera, where the view just represents a "fly on the wall" and not an actual character's viewpoint). This may be "third-person omniscient" where the reader can see things that happen and even hear several characters' thoughts, or "third-person limited" where some information (such as character thoughts) is concealed from the reader.

A rare kind of story in literature is that of second-person, where the story is told from the *reader's* point of view. This kind of story is rare because it is very hard to write in a way that is believable.

You might be realizing about this time that **almost every game narrative is told in second person**. This is the case whenever the player is controlling the main character, which is most of the time.

And this is another reason why storytelling in games is so hard.

Interactive Characters

Sometimes, the overall plot arc of the game is fixed and linear, but there are a number of characters (specifically, "non-player characters," referred to as NPCs) that the player can interact with. In video games, where interpersonal relationships must be quantified, there are two common ways to treat NPCs and their relations with the player.

Flags

A “flag” in this context is just a binary (yes-or-no) value. An action did or did not happen. The player did or did not talk to a certain NPC, and if they did, they either did or did not choose a particular conversation path. As a result, certain new NPCs or conversation options may appear or disappear.

The advantage of flags is that it is very simple to do. Everything either happens or it doesn’t, making the logic that drives the characters pretty easy to follow. It’s also easy to implement in code; programmers can do binary logic easily.

The disadvantage is that there is not a lot of depth to these characters, if there is only black and white and no shades of gray in between. You *can* add more complexity by combining binary values (“only make Aragorn appear *if* Frodo chose to travel to the Prancing Pony *and* he successfully avoided capture by the Ringwraith *and* he puts on the Ring”) but at this point it can get complex to follow, reducing the benefit of simplicity mentioned earlier.

Affinity

Instead of a yes/no dichotomy, use numeric values to measure things like how strongly a character feels affection or hatred towards another character. If you have to decide whether an NPC behaves in a certain way, check to see if its affinity value is past a certain threshold value. (In tabletop RPGs, it is common to also add a die-roll to the mix, so that things may or may not happen based partly on chance.)

The advantage of an affinity system is that it is still fairly simple, it can handle more complexity than a flag system, and it is much more expressive.

However, you must be careful with affinity systems. There is a danger of having a very muddy system (where the player can’t tell why a character behaved in a certain way), especially if several affinity variables or a random die-roll are included in determining the outcome. In short, it is hard to get this kind of system to “feel right” without a *lot* of playtesting.

Character Conversations

Some games include a conversation system, where the player is given a choice of things to say, and then the NPC they’re talking to responds.

Conversation systems can be thought of in the same way as game narrative systems. A conversation can be:

- Linear. I walk up to an NPC and choose the “Talk” command. They tell me something. That is all.

- Branching. I initiate a conversation. The NPC says something, and I am then given a choice of responses. Based on my response, the NPC may say something else and then give me a brand new set of responses.
- Parallel. I initiate a conversation. The NPC says something, and I am given a choice of responses, then they respond, and so on. There are several combinations of responses that can get me to the same outcomes, however.
- Threaded. I initiate a conversation, and am given a set of different ways to begin. The NPC responds to what I say, and each action I take within the conversation may open up new paths of conversation and close older, no-longer relevant conversation threads.
- Dynamic. I go through a branching or parallel conversation. Depending on the outcome, I proceed to a new conversation (or a new area of exploration within the current conversation).

Writing dialogue in each of these methods is proportional in difficulty to writing a story of each form.

Writing Good Characters

Some characters in stories are "round" – they have many facets to them, a deep character with many layers, and are highly detailed and developed over the course of the story.

Other characters are "flat" (or "shallow") – they are very simple, don't have a very deep personality (at least, not that the audience can see), and do not develop or change much (if at all).

Normally if we say a character is "flat" it sounds a bit derogatory. Are flat characters always bad in stories? I don't think so; not all characters need to be complicated, deep, or well-defined. Even in Shakespearean plays, some minor characters are flat (I'm looking at *you*, Rosencrantz and Guildenstern), but the main protagonist and villain at least should be round.

A rule of thumb is that a character should develop over time as we seen them through the story. As a corollary, minor characters with very little "screen time" can be flat since they don't have much time to develop anyway; the characters who we see the most often (generally the main characters) should develop a great deal. A flat protagonist is probably not a good idea if you are trying to make a strong character.

Archetypes versus Stereotypes

Here's a question I was once asked in a job interview: describe the terms "archetype" and "stereotype" and the difference between them.

McKee says that story is about forms, not formulas. Archetypes are forms; stereotypes are formulas.

“Villain” is an archetype. The Snidely Whiplash-esque, mustache-twirling, top-hat-and-black-cape-wearing, purely-mean-for-its-own-sake bad guy is a stereotype.

Archetypes are useful. They allow us to tell a story with believable characters that fit familiar forms. Stereotypes are overused, and typically make a story *less* believable because the audience has seen these same characters before from other stories, so they do not seem unique. Generally, avoid stereotypes, unless there is very good reason (such as if your story is a parody that is making fun of a particular character stereotype).

Freytag’s Triangle

You might have seen this before in a grade school literature class. The idea is that stories start with a small amount of exposition to set the tone, then they have rising action where things get more intense, then they reach some kind of climax (a final battle or confrontation, etc.), then there is falling action and resolution at the end as the story reaches its conclusion.

Note that this is not just true for story, but for gameplay as well. Games have exposition (we call it the “opening cinematic” and “tutorial level”), rising action (most of the game), climax (final boss fight), falling action (occasionally some post-fight sequence like a timed escape from the building before it blows up), and resolution (end cinematic).

You get a more dramatic experience if the dramatic arcs of story and gameplay are aligned and in synch with each other. This is a lot harder than it first appears; in many games, the most difficult part of gameplay occurs somewhere in the middle of the game, when the enemies and challenges are getting harder but you haven’t yet found new weapons and abilities to power yourself up to compensate. As the player gets more powerful and better at playing, the challenge of a game often *decreases* over time, even as the intensity of the story is *increasing*. You can tell these games when the final boss fight is introduced with this high amount of drama and foreboding, and then the player wins after swinging their sword around for a few seconds – it feels anticlimactic and not very believable.

In short, as you balance your game, pay attention to the story and whether the story drama is proportional to the dramatic moments in gameplay.

It’s Still a Game

Remember that, as a game designer, you are still making a *game* and not a *story*. In most cases, the story should not preclude gameplay. At the very least, you should be extremely careful when you are tempted to make a concession in gameplay for the sake of the story, or when you plan to overload the player with non-interactive story bits.

When writing the narrative for a game, return frequently to your core aesthetic – your overall vision of the optimal experience your game will offer – and make sure that the story supports it!

Lessons Learned

There are lots of ways to take a traditional linear story and make it more interactive. All of these have advantages and drawbacks. If the Holy Grail of a full interactive story is even achievable, we have still not discovered how.

But we should keep trying, and exploring the various non-linear story forms that are uniquely suited to games.

Reading

If you have time, before beginning the Homeplay below, please take the time to read at least five other posts at your skill level from the *Pente* challenge from Level 9 (posted on the forum), and also at least five other posts at a skill level *above* yours (unless you posted in Black Diamond). Do this on or before Monday, August 3, noon GMT.

As you read, pay attention to the variety of responses you see. Do you see any recurring themes, or are people's experiences different from each other? Why do you think that is? Reflect on this.

Midterm Review

You get a break this weekend: no homeplay. If this were taught in a classroom, you would have a take-home midterm exam at this time. But I do not have the time or the desire to grade 1400 exams, nor to generate a series of unique questions. Instead, I provide this brief review of what we have covered. Next Monday, we will transition from the mostly-theoretical discussions on what makes compelling games, to the mostly-practical business of actually designing an original non-digital game from scratch and taking it through the process from concept to completion.

We have come a long way in only five weeks, and for those of you who have kept up, I salute you. We've covered most of the basics of how to design a game. In particular, we have discussed:

- The importance of having a critical vocabulary to discuss games, along with the acknowledgement that we do not have one that is strongly developed yet.
- There are many definitions of the word "game"… but they are generally too broad, too narrow, or both. Most games include rules, goals, conflict, decision-making, the "Magic

Circle,” a lack of net material gain, and an uncertain outcome. Most games are activities, they are voluntary, they are make believe, they are closed systems, they are representations or simulations, and they are a form of art.

- Player intentionality (the ability of the player to form a plan and carry it out through the game’s systems) requires clear feedback, and competes with linear narrative.
- There are many formal elements of games: players, goals, rules, resources, and so on. Each element must be intentionally designed, and it is also useful to start with the formal elements when analyzing an existing game.
- Games are systems. They must be set in motion (i.e. played and experienced) to be fully understood.
- Games require rules for setup, progression of play, and resolution.
- Mechanics are the rules of play. When set in motion during play, these lead to the Dynamics, or the actual play of the game. The dynamics have a mental and emotional impact on the players, or the Aesthetics of the game. Designers create the Mechanics only, designing from the “inside out”; players experience the Aesthetics first, learning the game from the “outside in.”
- Most games have a single strong core, the gameplay experience that is fun or engaging or meaningful or whatever the overall vision is. When designing the game, return to this core vision constantly and ask if all of the mechanics support the core.
- Emergence is a special kind of dynamic that is experienced as complexity that arises from the interaction of relatively simple dynamics. This can be useful in that the mechanics are what must be designed (and programmed, in the case of video games) so it can ostensibly deliver a deep gameplay experience for relatively little effort. In reality, much of the cost is simply shifted to playtesting, as emergence is not always a positive effect on gameplay, so it must be tested heavily.
- Feedback loops are another special kind of dynamic. Positive feedback destabilizes the game by causing the winners to win faster and the losers to lose faster in a “snowball” effect. Negative feedback stabilizes the game by disadvantaging the winners and providing extra help to the losers. Feedback loops are not always desired or intentional, though they do have their uses; designers must be sensitive to detecting feedback loops and deciding whether they are good or bad for the game.
- There are many different kinds of fun. There are also many systems that define player types. Designers create fun, not players, so “player types” are mainly useful in understanding the kinds of experiences we must create. Many things that we find “fun” can be traced directly to our distant ancestors and the skills they needed to survive.
- There are many kinds of decisions players can make in games. Meaningless decisions have no effect on the game’s outcome. Blind decisions have an effect, but give the player no information on which to base their choice, so any decision is as good as any other. Obvious decisions have an effect, and offer *too much* information, so that one decision is clearly better than all the others. More interesting decisions involve some kind of tradeoff, where the player gains one thing but loses another.
- When people enter a deep state of concentration, it is called being “in the flow.” One way to achieve this state is by being challenged by a task that is not too easy but also not too

hard. It is a pleasurable state to be in, and games are particularly good at putting their players in a flow state. Flow is pleasurable because the player is learning a new skill (or strengthening an existing skill).

- When writing linear stories, we can learn a lot from the last couple of centuries. Aristotle tells us that stories should be a believable chain of causes and effects brought about by the main character, and that stories should have a beginning, middle and end. McKee tells us that story is about forms, that story is about change, and that character is more important than characterization. Campbell gives us the Hero's Journey story form, which is useful when writing stories about heros.
- There are several different non-linear story structures that allow for player choices to influence the story. They all have advantages and disadvantages.

Have a restful weekend, and I'll see you back here on Monday, August 3 at noon GMT.

Level 11: Design Project Overview

By ai864

You made a game on the first day of this course. It took you all of 15 minutes. It probably wasn't very good. At this point, with your current understanding of flow states, feedback loops and kinds of decisions, you can probably isolate the reasons *why* it wasn't very good.

You made a few other games after that one. You might be proud of some of them, and embarrassed of others. Looking back, you might find one that you *were* proud of that you now realize could have been better. Or maybe not.

At any rate, you know how to make games.

So, let's make a *good* game. You have all the knowledge and theory you need.

We will spend the next month making a game. If you're a student, that may sound like an incredibly long time to you, and you will be surprised at how fast it flies by. If you're a little more experienced, it may sound like an unreasonably *short* time, but I promise you will manage. Do not fear; we will take things one step at a time, through the entire process. You will not have to complete everything all at once. (Nor should you. That is not how games are made.)

Readings

Read the following:

- *Challenges for Game Designers*, Chapter 11 (Targeting a Market). We have discussed the importance designing for the player (as opposed to the designer) earlier in this course. This chapter goes into more detail, giving some considerations when designing a game for target-demographic or mass-market appeal.

Design Project: an Overview

In my classroom courses, I call this a **portfolio project** – a game that will ultimately go into the student's game design portfolio as a way of showing their skill at game design. You may consider doing this as well, depending on your situation and your career goals.

The purpose of the Design Project is to gain some experience in taking a game through the entire process from concept to completion. Because of this, **do not** simply start with an existing design (such as an earlier game you created in this course, or an idea you've had floating around in your head for awhile). You have plenty of time – the entire rest of your life! – to take your existing projects further. For now, get some practice at *all* of the stages of designing a game.

The Process

As you might guess from the syllabus, the process we will follow is going to go something like this:

- First, generate some core ideas for games. These do not have to be fleshed out in any meaningful way, they are just "seeds" that can serve as starting points. You will choose one to serve as the basis for your Design Project.
- Next, you will create the core mechanics of the game. The game does not yet have to be complete with all details fleshed out, but it does have to be at the point where you can start playing it with yourself (even if you have to make up a lot of the rules as you go along). You'll play your own game in private, working on it until the point where you have a complete set of rules.
- After that you will bring in some close friends, family, confidantes, or other participants of this course. Share your project with them, play the game with them, and get feedback. The key here is to figure out if the core of the game is fun at all (if it is not, you can start over with one of your other ideas or else modify your current one and try again). If it doesn't start out feeling like there is some magical fun quality to the play, that feeling is unlikely to materialize later – it is far better to abandon an idea early and try again than to waste a large amount of time on something that is just not going to work. Ideas are cheap, implementation is expensive; act accordingly.
- When you have the core of the game working and it is meeting its design goals, it will be time to get into the details. Make sure the game can be played to completion, without the designer being present to answer questions or make on-the-fly rulings. Get to the point where the game has a complete set of rules, with no dead-ends or holes that cause the game to stop when the players can't figure out what happens next. You'll playtest with new players who have not seen the game before, and observe them from a distance to see what they do.
- Once you are confident that your game is solid, you'll explore "blindtesting" – a playtest where you are not present at all. You'll give your game to some other people who will agree to test it and provide feedback. This most closely simulates actual market conditions, where a person buying a game does not have direct contact with the game's designer, and they must figure out how to play it for themselves.
- After all of the details are complete in your game, it is time to tweak the small things. Make sure the game is balanced – that is, that there are no strategy exploits that are too powerful, and that all players feel like they have a reasonable chance of success.
- Lastly, as the game nears completion and the mechanics become solidified, you'll consider the "user interface" of your game – the visual design of the physical components that will make the game as pleasant, easy to learn and easy to play as possible.
- Once everything is set, you'll spend a short amount of time on the craft of the physical components, making the artwork and assembling the components in their final form.

Keep in mind that game design is an iterative process, and that at any point in the process you may find a reason to return to earlier steps to redo something. This is fine, and it is to be expected. This is also the reason why it is better to kill an idea early than to abandon it late. If you find that you have to start over from scratch, you'll have more time remaining if you start over in the first week (as opposed to restarting the project in the *last* week).

Idea Generation

Recall from Level 4 that there are many ways to start conceiving of ideas. Start with core aesthetics, or a core mechanic. Start with materials from other sources. Start with a narrative. And so on.

Today, start generating some ideas. Look in the world around you. What systems do you see that would make great games? Carry a notebook with you wherever you go in the next few days, and write down every idea that occurs to you, no matter how silly it may seem.

The more you generate ideas, the easier it gets.

Design Project Constraints

I could leave this entire project open-ended, but in order to get you started I'm going to give you some constraints. Remember, constraints are your friends.

If you've never designed a complete game before this course, follow this set of constraints. Create a board game, card game, or tile-laying game (that is, it must either have a board, cards, or tiles as physical components). It may have more than one of these components, and it may involve additional components beyond these (such as dice or pawns). You may choose any theme you want, as long as it is original – do not use an existing IP (intellectual property). In short, if your work would violate someone else's trademark or copyright, don't do it. You will undoubtedly work with other people's IP at various points in your own career; take the opportunity now to do something original with *your own* IP.

I'm going to place two more restrictions to help you. First, you **may not** make a trivia game, or any other game that relies on large amounts of content (such as *Trivial Pursuit*, *Pictionary*, *Apples to Apples*, or *Cranium*). This is purely for the purpose of keeping your scope limited; if you have to generate 250 cards with unique trivia questions on them, it will leave you far less time for playtesting the game mechanics. I would put Trading Card Games (like *Magic: the Gathering* and *Pokemon TCG*) in this category as well, since it requires so much time to create a large number of cards.

Second, you **may not** use "roll-and-move" mechanics in any form. Do not throw dice and then move a pawn around the track. Do not use a spinner or a teetotum or card draws or any other random-number-generating device to determine what a player does on their turn. There are several reasons for this prohibition. First, the mechanic is highly overused, and it is practically impossible for you to make a game that will not feel like a clone of *Monopoly*, *Trouble*, *Sorry!*, *Chutes & Ladders*, or any of the other myriad games that rely on this as their core mechanic. Second, the mechanic essentially makes the key decision each turn for the player, so the *game* is making interesting decisions but the *player* is not. By divorcing player intentionality from the game's outcome, you usually end up with a game that is not particularly fun to play (no matter how fun it is to design).

If you have designed one or more complete games before but still do not feel like you are a strong game designer, follow this set of constraints. Follow all of the Green Circle constraints above. In addition, add one of the following constraints. This is your choice, based entirely on your area of interest within game design:

- Design your game such that it has a strong embedded narrative that is interactive in some way. You will have to think of ways to tell a story through the player actions of a board game, and how to integrate narrative and game mechanics. If you are interested primarily in RPGs or other forms of storytelling, do this.
- Create a purely cooperative board game for two or more players, so that everyone wins or loses as a team. This is challenging for several reasons. The game must provide systems that are the opposition, since the players do not provide opposition to each other. Cooperative games generally have a problem where a single skilled player can direct all of the other players (since everyone is cooperating, after all), leading to an MDA Aesthetic where most of the players are bored because they are just being told what to do by another player. If you are interested in the social dynamics of games, choose this.
- Make a two-player head-to-head game with *asymmetry*: the players start with unequal resources, positions, capabilities, and so on… and yet they are balanced even though they are quite different. These games are not so hard to design the core rules for, but they are very difficult to balance. If you are interested in the technical and mathematical side of game design and game balance, try this.
- Create a game to teach any topic that is normally taught at the high school (pre-college) level. It is up to you whether to teach a narrow, specific fact or a broad concept. The challenge here, of course, is to start with a fun game and not have the focus on education get in the way of that. If you're interested in "serious games" (games that have a purpose other than pure entertainment), then do this project.

If you have designed multiple games professionally and you consider yourself highly experienced, follow this set of constraints. Ignore everything above. You must create a board game that uses a "roll-and-move" mechanic as the primary gameplay activity. **But make it good.**

This mechanic is highly overused in games. It also creates a separation between the player's decisions and the actions that the player takes on the board. It is therefore extremely challenging to design a game that uses this mechanic in a way that feels fresh, original, and compelling. But I'm sure if you have reached this point in your career, you are up to the challenge.

What If I Don’t Want To Make a Board Game?

Some of you expressed a strong interest in board games and are excited to get started. Don’t let me keep you. Realize that you are in the lucky minority.

Some of you are still more interested in making video games. I’ll remind you that the vast majority of your time making a video game will be spent creating art assets and writing programming code, and if you want to learn *game design* then you should choose an activity where the bulk of your time is spent *designing the game*. The principles and concepts of game design are mostly the same, whether you work in cardboard or code, so if you’ve got the skills to design video games you should be able to use those same skills to make a board game.

Some of you expressed interest in creating tabletop role-playing games. I’ll remind you that evaluating the design of an RPG is tricky, since a sufficiently skilled GM and players can salvage a weak system (or, sufficiently inexperienced players can ruin a perfectly good system). This will make playtesting far more difficult to evaluate, so you will find it useful to practice on a board game project first. Note that the line between board game and RPG has blurred in the past few years, given narrative-heavy board games like *Android* and mechanics-heavy RPGs like *D&D 4th Edition*.

Some of you might have additional real-world constraints. You might be on a budget, and so you can’t spend more than a certain amount of money on your prototype. You might live in a remote location where prototyping materials are scarce, and you’ll have to make do with what you have. You might have less time than usual to devote to your project, in which case you’ll need to design a game that has a short play time (so that you can playtest and iterate more frequently in less time). If you have constraints from your life that affect this project, consider those to be part of the project. A designer should not complain that they lack the resources to make the game they want; rather, they should find a way to make the *best game possible* with the resources they have.

Homeplay

I ask three things of you:

- Start generating ideas for your Design Project now, based on the constraints above. As I mentioned earlier, keep them in a notebook or some other place where they will not get lost, and that you keep with you constantly so that you can write down your ideas as you think of them.
- By **Wednesday, August 5, noon GMT**: look over your ideas and post your three favorites on the forum. Note that this is a day earlier than usual, in order to give time for feedback.
- By **Thursday, August 6, noon GMT**: read the posts of two other people at your same skill level, and provide constructive comments on their ideas. If you posted in Blue

Square or Black Diamond, also critique three others at a skill level below yours. **If you see posts with no responses, reply to those first, so that everyone can have at least some feedback.**

You may also use Twitter (with the #GDCU tag) to ask for immediate feedback of ideas as they occur to you.

Level 12: Solo Testing

By ai864

At this point you have some ideas, and you have some feedback. As with many things in life and in game design, you could sit here forever contemplating which is the best choice… but at some point you'll need to start working towards your goal. Choose a direction and go with it, even if it might not be the best one. Trust that you will be able to use the iterative process to fix any mistakes you make along the way.

Today I'd like to cover the general concept of playtesting. As we will see, there are many different kinds of playtesting, and it is important to be able to differentiate between them in order to get maximum value from our time.

Readings

There are no readings for today, other than this blog post. Take the time that you would have spend reading, and use it to work on your Design Project.

Different Kinds of Playtesting

The word "playtesting," like the word "game," is overused and can mean different things to different people. In general, the term covers any activity where you are playing a game in progress for the purpose of improving it. But different playtests may have different goals, and it is important to know what your goals are before you do anything.

I'll be playing a bit fast and loose with terminology here, so in this case the concepts are more important than the labels I'm giving them.

Bug Testing (or Quality Assurance)

The purpose of QA is to find errors in the game's behavior relative to its design. "Fun" does not enter the equation. If the designer says that the game should do one thing and it actually does another (even if what the game *is* doing may be superior), that is a bug that needs to be identified.

Normally, we think of bug testing as specific to video games. Board games do have a corresponding kind of testing, where the purpose is to find holes in the rules and dead ends in gameplay – gaps in the game that the designer did not cover.

Focus Testing

In a focus test, you bring together players that are part of the target audience's demographic in order to determine how well a game serves their needs. This is normally done for marketing purposes, but if game designers are involved it can also help to make the game more enjoyable for that particular demographic.

Usability Testing

In a usability test, players are given specific tasks to accomplish in an attempt to see whether they understand how to control the game. This is done frequently in the greater software industry to make sure that a piece of software is easy to learn and easy to use. Video games can take advantage of this as well, and results from a usability test can be used to either change the controls or modify the early levels to teach those controls more effectively.

In board games, usability is doubly important, because there is no computer to respond to player input for you. If you misunderstand how houses work in *Monopoly* and place them on Community Chest spaces, the game will not stop you. By observing players who are trying to play your game, you can learn a lot about how to design the various game bits so that they are easy and intuitive to use.

Balance Testing

A fun game can quickly become boring if some kind of play exploit exists that lets a player bypass most of the interesting choices in the game. If only one strategy can win and it is just a matter of which player follows that strategy the best, it is not as interesting as if there are multiple paths to victory. Likewise, if one player has a clear advantage over the others, it is important to identify that so that players do not feel the game is being unfair. The purpose of this kind of test is to identify imbalances in the game so that the designer can fix them.

Fun Testing

A game can be usable, balanced and functional and still be uninteresting. That elusive "fun factor" may be hard to design intentionally, but when people are playing the game it is pretty obvious whether they are having fun or not. Certain aspects of the game may be more fun than others, so it is also important to figure out what parts of the game need to stay the same… not just what to change.

All of these forms of testing have some elements in common. Best practices are similar if not identical. All are important to the success of a project. So why make a distinction?

The reason is that each is appropriate at different stages of completion in a project. Each kind of testing has different goals, and you need to know what your goal is before you can achieve it.

Order of Effects

When should you do which kind of playtesting? What order do you do them in? A lot depends on your particular project, so some of this will be up to your judgment as the designer. However, there are some rules of thumb.

- Very early on in the project, you need to make sure your project will meet its design goals (usually the "design goal" is to make a game that's fun to play). Testing for fun is necessary to make sure you do not spend a lot of time building on the wrong foundation. If you are making a game for a specific market, focus testing may be involved at an early stage as well, simply to ask the target audience if a game with a particular concept sounds interesting to them at all.
- Once you know that you have something, you need to solidify the mechanics. Design the whole game, making sure that all the details are taken care of. Test for "bugs." (Note that bug testing in software projects is often done continually throughout the project, increasing in intensity toward the end. Non-digital games are easier to "debug" though, and a "bug" can stop a playtest in its tracks, so it is important for us to have a complete set of rules early in the process.)
- Once the game is fun and the design is complete, gradually shift from testing for fun to testing for game balance. Make sure that all the numeric values and player abilities are where you want them to be.
- When the game is working and balanced, towards the end, you'll want to think more about the usability of the game. When you change usability you are not changing any mechanics, merely the way those mechanics are presented visually to the players. This is an important step that is often neglected. If you've ever encountered a game that you could only learn by being taught by another player (as opposed to reading the rules yourself), that is the kind of usability failure you want to avoid in your own projects. You may also do additional focus testing at this time, to make sure that the theme and visual elements of the game appeal to the target audience.

As I said, these are just guidelines. If it is incredibly important that your game be well received by a particular demographic, for example, you may be doing focus testing throughout the project at all stages. Do not let this order of things be your master.

Different Kinds of Playtesters

As there are different kinds of testing, there are also different kinds of testers. Each kind of tester has their own strengths and weaknesses, and some are more important for some kinds of testing than others.

- Yourself. You are your own most valuable playtester. Do not forget your ability to play your game on your own. You know your game better than anyone.

- Other game designers. If you are lucky enough to personally know some other skilled game designers, you can get some very useful testing done through them. They are able to critically analyze your game and propose design solutions. (If you do not know any professional designers, perhaps you can at least make contact with other participants of this course.)
- Close friends, family, and confidantes. People close to you who are willing to provide their time to test your game are very useful. They are approachable and can make themselves available as a favor to you. Take good care of them, and do not abuse their kindness. Note that these people may not fall into any of the other categories, so while they are good for early tests, they may not be appropriate in more focused testing for bugs or balance since they may not know what to look for.
- Experienced gamers. Skilled game players are great at finding exploits and dominant strategies in a game, and are appropriate for balance testing.
- Complete strangers. People in your target audience are appropriate for focus testing and usability testing, and they are absolutely critical when testing for fun. Finding them can be tricky, though, because it is not in most of our natures to just walk up to someone we've never met and ask them to play a game. We will talk more about this in the coming weeks.

Order of Familiarity

In general, you will want to go through testers in order from more to less familiar. Test with yourself first, then with close friends, then with acquaintances that are useful (because they are designers, gamers, or part of the target market), and then with strangers.

If you show your work to other people too early, it will likely be in such a rough state with multiple design flaws and holes in the rules that it will waste their time and frustrate them, and you want to treat your playtesters better than that. Also, if you start playtesting with strangers too early in the process, you may not get useful feedback – if your game prototype is in a rough state with only crude art and components, for example, the playtesters may be so busy commenting on the poor quality of the pieces that they will not be able to concentrate on the gameplay.

At this point you might be tempted to just do *all* of the playtesting by yourself, so that you don't need to rely on other people or keep track of them. In practice, the designer eventually gets too close to their own project and is *so* familiar with the game's systems that they can miss some really obvious flaws. If you keep the same set of playtesters for long enough, they will suffer from this problem as well. You need to bring in fresh sets of eyes to look at your game on a continuing basis throughout the project.

Playing By Yourself

In the early part of playtesting, when you are playing the game on your own, here are some things you should be looking for:

Does the game meet your design goals?

Is it fun, at least for you? While you are not the ideal playtester to judge effectiveness most of the time, if you are not having fun then most other people will probably not either.

Are there any holes in the rules?

A "hole" is a situation where the rules simply do not say how to proceed. For example, perhaps one of your rules is that a player's army can attack another player's army, but you don't yet have rules for resolving the attack. What happens in this case? In practice, what happens is that the players sit around and wait while the designer figures out what to do!

As an example, consider these rules for Tic-Tac-Toe played on a 4×4 grid:

- Players: 2
- Objective: Get a straight line of symbols.
- Setup: Draw a 4×4 square grid.
- Progression of play: On your turn, place your symbol ("X" or "O") on an empty square.
- Resolution: If either player on their turn has a set of four of their symbol in a straight line (across, down, or diagonally), they win.

If you try to play this game just following the rules, you'll quickly realize that you can't even start – nowhere does it say which player is X or O, or who takes the first turn! To fix this, you would add a situation to handle this. For example:

Setup: Draw a 4×4 square grid. Choose a player to go first, who is assigned the symbol "X". The other player is given the symbol "O".

Are there any dead ends?

A "dead end" is a game state where there is no way to proceed further, but the game is not resolved. Consider our revised *4×4 Tic-Tac-Toe* rules above. Suppose that both players fill up all squares on the board without anyone winning. At this point the game cannot proceed, because the rules say a player must place their symbol on an empty square. There is no empty square, so the player cannot take a turn. But there is also no resolution, because neither player has won. In this case, a new rule would have to be added (such as: in the resolution, if neither player can make a legal move and no one has won, then the game ends in a tie).

Are any of the rules unclear?

It is natural for us to assume things that are in our head, to the point that we often forget to write them down in our rules. Try to look at your rules and see if there is anything you are assuming that your players might not.

Are there any really obvious rules exploits?

Is there a single strategy that wins the game easily? Try to find it. It's much less embarrassing if you find and fix it yourself, as opposed to having it discovered by your playtesters (or worse, your players *after* you release the game). Clarity and exploits are often hard to find in your own game; you tried to design this game to not have any problems, after all. Still, make an honest effort, and sometimes you will be rewarded by finding and fixing errors early (which saves a *lot* of time in the long run, leaving you more time to iterate on other parts of your design).

You might think that looking for exploits is something to do later in the project when balancing the game. Sometimes it is. It is a matter of degree. If an exploit is *so* powerful and *so* obvious that it prevents your playtests from giving you real information about your game, fix it now.

Solo Test Difficulties

There are a few things that are hard to test alone:

- Realtime multiplayer games, such as games where you must slap a card or say an answer faster than your opponent.
- Hidden information games, where each player has information that only they know and that is important to keep secret from the opponent.
- Trading, negotiation, and auction games, where each player must place a value on an item, and different players may value things differently (and especially when players can artificially extort high prices or drive up the cost of an item at auction just to make their opponent pay more).

For the latter two, it is possible to play anyway, by simply limiting your actions to what you think you would do if you were in each player's situation, knowing only what they would reasonably know. Some people find this more difficult than others.

The simplest answer here is, for the purposes of this project, to not use mechanics that you can't test yourself. The alternative is to bring in another player or two early on in this case only, after you take things as far as you can on your own.

Let It Grow

Experienced designers often talk of a game "making itself" – as if the game has a life of its own, and the designer is merely guiding it rather than creating it. On the surface, this seems strange because really, the game is just sitting there and doing nothing unless the designer is playing it or changing it. What's going on here?

I think that what is *really* happening is that the creation of a game is a learning process. You may have some idea of where you want your game to end up, but the final version may be very different from what you originally envisioned. The reason why it changes is that at the beginning, you don't know very much about your game. You have some basic ideas, but you don't actually know how the mechanics will interact, or what the actual dynamics and aesthetics will be. As you playtest, you learn more about how your game's systems are working. As you learn, you become more able to predict the effects that changes will have on the system.

Right now, though, you don't have that experience… at least not with *this* game. Playtesting on your own is your first act of discovery. As you discover, it may seem as though your game *wants* to grow in a new direction, as if it has a life of its own. If you feel that, go ahead and listen to your game. See where the process of discovery takes you.

Homeplay

As the nature of the work at this stage is solo, you do not have to post anything on the forums. Work alone until Monday, at which point we will start involving others.

Before Monday, August 10, noon GMT, **create a playable prototype** of your game. You may find it useful to review the section of Level 4 on prototyping. Remember that you can make a prototype in about 15 minutes – it doesn't have to be pretty and it doesn't even have to be complete (yet), but it does need to be at the point where you can sit down and play it by yourself.

Then, **play the game by yourself**, at least once. In your idea notebook, write down any problems you encountered or questions that you ran into. Trust me, no matter how obvious the things are that you need to fix, you will forget them if you don't write them down.

Finally, **write down the rules** once your game is at least somewhat playable. This is also for your own reference, so that you do not forget.

Level 13: Playing With Designers

By ai864

As the designer, it is an important skill to be able to playtest your own creations (which you've already done at least once). It is also important to be able to set up conditions for *other people* to playtest your games, so that you can get useful information from the precious time you have (which we will cover over the next week).

There is another side to playtesting: the ability to playtest *other* people's games and provide quality feedback. This is a skill in and of itself, and a surprisingly rare one to find. This scarcity makes it a valuable skill. Personally, I have received many freelance opportunities through colleagues, simply because they know that I am good at finding the flaws in their designs. This is what we cover today: the ability to critically analyze a fellow designer's work-in-progress.

Readings

Read the following: "Giving Criticism – the good, the bad, and the ugly!"

This may not be a class on giving constructive criticism, but it is something I'm going to ask everyone to be doing. Far too often in classes, students are asked to give peer feedback and review, and yet not given the tools to do so in a useful way. I think many teachers either take the stance that simply giving feedback enough times will make people better at it ("practice makes perfect") or else that feedback techniques should be taught in some other class ("I can't waste precious class time on that").

This article may not be particularly comprehensive, but it is short, and after doing a Google search for "constructive criticism" it is the one that I found that fits best with the advice I give in my classes.

The Time Barter System

At Protospiel, an annual gathering of non-digital game designers, participants are encouraged to give as much playtesting time as they take. For example, if your prototype takes two hours of play and discussion and it requires four players (other than yourself), a single playtest consumes eight person-hours of time; in exchange for that playtest, you are then expected to playtest other people's prototypes for eight hours of your own time. This system prevents there from being an extreme shortage (or surplus) of testers relative to games, and it gives people incentive to respect each other's time.

Notice that this means you tend to spend far more time playing other people's games than you do playing your own. You could even say that, given the time difference, the ability to be a good playtester is *more* important than being able to design your own games.

Keep this in mind as you proceed through the rest of this course. You will be consuming large amounts of other people's time as you iterate through your own designs. Accordingly, treat your testers with respect. (It wouldn't be out of the question to give them food or some other compensation, as well, if it is in your means to provide.)

If you are in a group, playtesting with other designers should be relatively easy. Just meet up with your fellow designers. Keep in mind that you should be giving more of your time to other people's games than your own.

Next Steps After Solo Testing

At this point in the project, you should have a playable prototype of your game, and a set of rules. You should have playtested on your own at least once, identified any really obvious problems, and iterated on your design. You should continue to do this until your design is at a point where you are confident that you can play all the way through without having to make major changes.

Once you reach that point, your goal shifts from "make this game work" to "make sure the core mechanics are fun" (or whatever your design goal happens to be, if not "fun"). Who would make the best playtesters to help with this?

Normal players (such as friends and family, or even complete strangers) are marginally useful here. By watching them, you can determine if they are having a good time and if your game is meeting its design goals. However, if there is a problem, a typical gamer will not be able to give you useful feedback other than "it's great" or "it sucks." It will be up to you as the designer to identify and fix the problems. Therefore, normal testers can be used if necessary, but their usefulness is limited.

Far better is to playtest with other game designers. Game designers can also let you know if the game is fun, *and* they can offer suggestions on where the problem points are and what can be changed to make your game better. You can often have wonderful discussion following the play of the game, on the design of your game and sometimes on game design in general. These kinds of discussions are important, and your game can get better much faster with them.

Finding Designer Playtesters

There are a few places to find other game designers.

If you are lucky enough to already work at a game company (or know someone who does), you probably already know some designers who you work with regularly. In this case, finding skilled testers is the easy part. The difficulty is that professional designers are often busy with work, and simply do not have the time to help you. You must work around their schedules. Also be prepared to offer something of value in exchange. You are essentially asking for a professional game design consult. Your colleague could spend the same amount of time freelancing and make anywhere from US$40 to $250 an hour, depending on their experience and the nature of the project. (I get those figures from personal experience and what I can piece together from what my colleagues say on the subject.) If you are asking for their time for free, be prepared to give a comparable amount of your own time to their projects in the future. *At least* be prepared to be extremely grateful.

What if you don't know any professional designers? Perhaps you signed up for this course in a group with your friends. This is where that group of yours will come in handy. Get in touch with your friends, and arrange a time to meet in the near future for a playtest session. Play through each of your games.

If you took this course alone and don't know anyone else, the next best thing is to check the forums. The bottom section of forums, "Local Communities," was set up specifically so that you could find other people in your local area. If you are not yet registered on the forums (but you did sign up for the course ahead of time), do that now – just be sure to sign up with the same email address that you registered with for the course (or at least drop enough clues in your forum account info to let me figure it out on my own). Arrange through the forums to meet at a neutral location. Who knows, this may be the start of a long-term professional relationship.

If you're having trouble finding others in your area on the forums, as a last resort, post your work on the course wiki and beg on the forums for playtesters. There may be other people in similar situations. Again, if others are willing to take a look at your work and provide feedback, return the favor and playtest *their* work. When playtesting another's work in this way, be sure to give them instructions for assembling a playable prototype. The easiest way to playtest in this way is to solo test each others' games. Another option is to arrange a meeting over the internet (using a chat tool such as IRC or Skype) and attempt to play remotely in real time (some games are easier to do in this way than others).

Being a Great Designer

As other people playtest your game, keep in mind the following:

- Your game is not perfect. If your game were perfect, you wouldn't need to playtest.
- There will be problems. The goal of playtesting is to find and eliminate those problems. If all your playtest did was confirm that your game is perfect, you have just wasted your own time and everyone else's.
- It is far better to identify problems in a small playtest, than for them to be found after the game is printed and ships to millions of players.
- If one of your playtesters finds a major problem in your game, they have given you a great gift. Do not be hostile or defensive; be gracious.
- When a problem is identified by a playtester, your goal is not to verbally defend your game or to explain why the playtester is wrong. First, even if your playtester *is* "wrong," it probably means a lot of other players will also be "wrong" in the same way, and you can't ship yourself in a game box in order to explain your Grand Vision to everyone.

Second, the playtester is probably right – they are seeing your game through fresh eyes, and are more likely to have an unbiased view of the game.

- If your playtesters do identify problems, the correct response is to write the issue down in your notebook… and then discuss your design goals with the playtesters so that you can get some ideas of how to preserve your goals while changing the game.
- Not all people are tactful. Sometimes people will say things about your game (or even about you, personally) that are downright hateful. Sometimes people will make fun of your game, or will taunt or berate you for a problem with your design. Keep in mind that, no matter how it is delivered, this is still extremely useful content.
- It takes a strong person to hear a statement like "your game sucks, it is the worst game I've ever played, and by extension *you* suck and you are nothing better than a waste of space" and to genuinely reply: "You have just helped me identify some major flaws in my game. **Thank you.**" Getting to the point in your life where you are emotionally strong enough to have an exchange like this should be one of your long-term goals as a game designer. You do not have to be like this right now. I'm not. But I have seen an exchange like this before from a great designer, and it made me realize how far I have to go.
- If it sounds like I'm repeating myself here, it's because I've seen this go horribly wrong so many times, that I think it is worth repeating.

Running a Great Playtest Session

If you want your playtesters to keep coming back for your future designs, be as respectful of their time as possible. Here are some things to consider:

- Before you show your game to other players, make sure the rules are fresh in your mind so that you do not need to look them up. Try explaining all of the rules to yourself in the mirror to make sure you can do it. This will save time, if it only takes you a couple minutes to explain rather than half an hour.
- If you already know there are problems (and you just don't have the solutions) or if you have specific design goals other than "make a fun game," let your playtesters know this up front. It will help them to be more aware of potential solutions.
- End your playtest as soon as you can. If you have received as much useful information as you are likely to after a half hour of play, stop there (even if the full game would last three hours). Remember that the purpose of the playtest is to identify problems, not to "play games." If you're not identifying problems, you are wasting everyone's time.
- Bring your playtest notebook and take good notes. You *will* forget everything that takes place, no matter how obvious your playtest results seem at the time, so make sure you write down every piece of information that you don't want to lose.

Being a Great Playtester

Here are some of the things you should keep in mind when testing other people's games:

- When testing, give the designer and the game your undivided attention. You would want others to extend the same courtesy to your game, after all.
- Don't leave in the middle of a test. Aside from being rude, it can throw off the results (not all games can gracefully handle it when a player leaves). At minimum, if you know you have limited time or that you may get called away in mid-game, let others know this up front so they can handle it accordingly.
- Be as detailed as possible. Don't just say that the game is "fun" or "boring," try to analyze why. You should have enough of a background at this point to give meaningful feedback. Make use of your design skills!
- Allow some time after the game for discussion with the other testers and the designer. Talk about your play experience, and how it was related to the mechanics.
- Remember that there are many possible playtest goals. Are you playing to see if the game is fun? Are you playing to win? Are you playing to find holes in the rules? Play accordingly. We are so used to playing games in our own personal style, that it can be difficult to remember that there are other ways to play. Keep the goals of the playtest in mind.
- Be polite. Attack the *game* mercilessly, but do not attack the *designer*.

Homeplay

Continue working on your game from last time. If your game is not already at the point where it is ready for playtesting with other designers, continue testing on your own until you are at that point.

You have two additional tasks.

First, before Thursday, August 13, noon GMT, **arrange a playtest session with other designers**. The session itself should take place on or before next Monday (August 20).

Second, **playtest other people's games**. Keep track of the number of person-hours spent in the playtest of your own game (not including yourself), and give at least that many hours of your own time towards helping your fellow participants.

Feedback

Do you know of any great articles on giving constructive criticism, or playtesting games as a designer? Post them in the comments on this blog, or on Twitter with the #GDCU tag.

Level 14: Playing with Non-Designers

By ai864

In 2006 at GDC, game designer and academic Jesse Schell said that the most important skill of a designer is to listen:

- Listen to your playtesters. They may not be professional designers and their suggestions may seem to make no sense. But if they react in a certain way to your game, it's up to you to figure out why.
- Listen to your game. Games often seem to take on a life of their own once they reach a certain complexity, and it's more important to make a *great* game than to make the game that you *originally intended*.
- Listen to yourself. Every time you follow your instincts as a designer, whether you're right or wrong, your instincts get better. (This is, incidentally, why a 20-year industry veteran is going to be a better game designer than a beginning college freshman, no matter how much "natural talent" either one possesses. There are no shortcuts. This is also why I'm having you make so many games over this summer, to get you to a higher level as fast as possible.)

Today, we cover the first of these: listening to playtesters. Once you are at a certain point in your game, you will want to playtest with some new people – preferably, the people in your target market, the ones who are representative of those you ultimately want to be playing your game.

Playtesting with gamers is very different from playing with other game designers. Done right, the feedback you can get from players in your target audience is even more meaningful than getting feedback from game designers, because you are seeing firsthand how your game will be experienced by the very kinds of people who will eventually be playing the final version. However, it is a very different skill. Non-designer playtesters will give a different kind of feedback, and it takes a bit more effort to find the root cause of problems that are identified. You have to be much more observant.

Readings

No readings for today. As with last time, if you know of any relevant readings you have encountered before, post it as comments to this blog post, or on Twitter with the #GDCU tag.

Street Performers

Have you ever seen a street performer? This is a musician, magician, juggler, mime, or other person who is performing for an audience of passers-by. These people rely on donations from observers; they do not get any pay other than what these strangers on the street choose to give them. Because of this, they tend to be very good at pleasing a crowd – if they aren't, they don't get to eat.

During the act, the audience is obviously paying attention to the performer. But what is the *performer* paying attention to? Next time you see one of these people, don't watch the act, but instead watch the performer. They aren't concentrating on themselves or their act, the way the audience is (the performer knows their own act inside and out, after all). Instead, they are watching the *audience*. They are looking for interest and excitement in the crowd. If they see a positive reaction or a negative one, they will adjust their act accordingly, on the fly. Maybe this particular crowd likes magic tricks with coins but not cards, or they seem to like blues more than jazz, or they're more excited by juggling pins than balls. The performer's most important skill is being able to read the audience.

Note that they do not ever stop their act to ask whether people are having a good time. They know by observing. They don't have to ask.

What Does This Have to do with Game Design?

When you are playtesting with non-designers, your role is similar to that of a street performer. Don't simply ask your playtesters if your game is fun; they may not be able to tell you, and if they do, they may not give you an accurate or precise answer. Instead, *watch* your testers as they play, and take notes:

- What is everyone's body posture? Are they leaning forward with interest? Are they leaning back in boredom? Are they standing up from excitement?
- Where are everyone's eyes going? Are they scanning the board constantly? Are the players looking at each other? Are they looking at you? Or are they looking around at the rest of the room, or counting the dots on the ceiling tiles?
- What kinds of moves are people making? Are they playing aggressively or defensively? Are players cooperating and negotiating, or are they backstabbing each other?
- Are your testers playing the game by the correct rules, or are they playing the "wrong" way, breaking rules or forgetting restrictions accidentally? Do your testers ever get stuck and need to look something up (or ask a rules question), or are they following the game flow smoothly?
- How do your observations compare to the design goals of your game? Is your game meeting its goals, or is it falling short?

Note that all of these things may vary during a single play session. You may find that certain parts of your game generate more engagement than others. Your goal during the playtest session is to observe these things.

Preparing for a Playtest Session with Non-Designers

People who are not fellow designers are sometimes (not always) less tolerant of extremely rough prototypes. A typical gamer, when handed a stack of hand-written index cards and instructed to move pennies around on a game board that's hand-drawn on notebook paper, may be concentrating so much on the poor quality of components that they have trouble thinking about the game mechanics. You may get a lot of comments about missing art or board layout, which are a waste of your time – after all, at this stage you just want to get feedback on the play experience, not the final artwork. We haven't even started talking about the appearance of the game yet!

If you are lucky enough to have some playtesters lined up who can handle a rough prototype, then you might not need to do anything. For everyone else, it may be worth a little bit of time at this point to create some components that, while not necessarily *high-quality*, are at least close enough to fake it.

As with your rough prototype, you do not want to put *too* much time into revising your components here. The more time you put in, the harder it will be for you (emotionally, at least) to make massive changes.

How do you make a prototype that looks better than hand-drawn, without taking too much time? Here are a few quick tips:

- Google image search is your friend. If you want art for some cards or a game board, type in some relevant search terms. Copy and paste. You can do this in minutes. Steal other people's artwork liberally.
- For basic components like pawns or tokens, use game bits from other existing games you might already own (if you do not already have a selection of these). It gives the game a slightly more professional look than using bottle caps or pennies or pieces of lint.
- For cards, you can create nice-looking ones in a program like Powerpoint or Visio without too much trouble. Standard-size cards are 2 ½ inches wide and 3 ½ inches tall. On a standard 8.5×11 sheet of paper, you can fit eight cards in a 4×2 grid with landscape orientation, or nine cards in a 3×3 grid if you use portrait orientation. Print out on standard paper and just cut with scissors.
- If you want your cards to be easier to shuffle and hold, use plastic card sleeves (normally sold in game and hobby shops to protect collectible cards like *Magic: the Gathering* cards). Insert a standard card of some kind (either a *Magic* card or just cards from a standard Poker deck) so that there are uniform card backs, then add your slip of paper in front.
- For a game board, printing it out on one or more pieces of paper is sufficient at this stage. You can create a board in Powerpoint or even in something as simple as MS Paint, using basic lines to make squares, the text tool to write text or numbers on the board, and copying/pasting art from other sources where needed or desired.
- You may find other tools that you like to use. Feel free to post them here!

Running a Playtest Session with Non-Designers

Since you are going to spend so much time taking notes and observing, you will probably find it easiest if you do not actually *play* the game. You may be able to take the role of a player when testing with other designers, and you're obviously taking on the role of *all* players when solo testing, but in the kind of testing we're talking about today you should avoid playing so that you can focus all of your attention on how your testers are interacting with the game and with each other.

If you didn’t before, you should formally write out a set of rules now. Hand the rules and components to your testers, stand back, and get out of their way. Let them know that you are there merely as an observer, *not* as a player and *not* as a resource. Instruct them to pretend you are not there, and to proceed as if the designer of the game were not in the room.

Your playtesters will probably forget this often. They will run into a place in the rules that is unclear, and they’ll have to ask clarification from you. **Do not answer immediately.** Instead, first answer their question with a question of your own: “If I weren’t here, and you had to make a judgment call on your own, what would you think?” Their answer may be the correct one… or it may be incorrect but enlightening. Either way, it will tell you how players are likely to perceive your game by default. After your players answer you, *then* you may give them the answer they were seeking. But don’t lose the opportunity to get a valuable bit of information in the process.

Sometimes your playtesters may not ask you, and they’ll simply start playing “wrong.” Maybe each player is supposed to draw two cards on their turn, but they only draw one. Or they skip the first step of every turn. Or they forget to apply the effects of some tiles in play. **Resist the temptation to stop them.** You will find this excruciating. There are few things as painful as *watching people play your game as it was not designed*. And yet, this is likely how people would play if you released your game at this moment, and this is something that is important for you to see, so that you can clarify the rules and game components later.

There is one other useful aspect to letting people play the game “incorrectly.” Sometimes you will find, quite by accident, that the way your testers are playing is actually *better* than your original rules. Most people, even non-designers, have a strong instinct towards play. Sometimes, people will violate the rules of a game because at an instinctive level, they are playing in a way that they believe will be more fun.

Finding Non-Designer Playtesters

Here’s the good news: finding non-designer playtesters is much easier than finding other game designers. There are more of the former than the latter in the general population.

This is where friends, family, and colleagues can become useful. They are often easy to ask for a favor. For many of us, they are local and available. If you somehow know no one in your local area (maybe you just moved), consider this just one more incentive to get out there and meet people – as if you didn’t already want to.

Do keep in mind that the people that know you are far less likely to give strongly negative criticism. They may tell you it is the best game they ever played, even if it isn’t, because there is an interpersonal relationship at stake that is likely more important to them than the outcome of some game project. In other words, expect some of these people to be big stinking liars. This is where observing them closely comes in; it is up to you to figure out what parts of the game are *actually* fun for these people.

Homeplay

Your homeplay this past Monday was to arrange for a playtest session with other designers. You may have already performed this playtest, or you may have just scheduled it to take place over the weekend, but that playtest session should be concluded before next Monday, August 17, noon GMT.

In addition, over the weekend, you should **arrange a playtest session with non-designers, to take place *after* the designer playtest.** This session can take place at any time on or before next Thursday (August 20), but it should be *arranged* (that is, you should have made plans with specific people) on or before next Monday (August 17).

Time permitting, you may continue to run additional playtest sessions, either with designers or non-designers.

Feedback

Do you know of any great articles on running playtests? Do you have any favorite tools using a computer to generate quality game components quickly and easily? Post them in the comments on this blog, or on Twitter with the #GDCU tag.

Level 15: Blindtesting

By ai864

When your game is playtested without you (or another of the game's designers) being personally present to observe, it is called **blindtesting**. This is the subject we cover today.

Readings

No readings for today. As before, if you know of any relevant readings you have encountered, post it as comments to this blog post, or on Twitter with the #GDCU tag.

The Challenges with Blindtesting

As you might imagine, blindtesting has a lot of problems compared with a playtest that you run in person:

- Without you there, any issue the players run into is a show-stopper that prevents them from finishing.
- Perhaps worse, the players may interpret the rules incorrectly and play anyway. If they are unaware they are playing "wrong" your test results may be skewed, and the testers won't even know it.
- Feedback is highly variable; as you have no doubt seen by now, some playtest groups are better than others. This problem is made worse when you are not present and cannot ask targeted questions.
- You cannot observe in real-time. The only information you get is what the group can report back to you in retrospect. The fidelity of information is much lower than when you are present.
- Setting up a blindtest takes a bit more time. You cannot simply bring your prototype with you to a friend's house and play right then. You have to find a way to get a copy of your game into the hands of your testers, and then *you have to leave*. Then you have to wait while your testers play your game, *on their schedule*, and then finally get back to you with some results. Even in the best case, blindtest results may take a day or two to get back to you, which is far more than an in-person playtest which you can conduct in a matter of minutes or hours.

Blindtesting is therefore limited in the type of information it can provide, and the schedule on which it can provide it.

Why Blindtesting?

If blindtesting is so limited, why bother with it at all? What use is it?

This is a technique we borrow from science. Awhile ago, some researchers noticed that the results of an experiment would be different if the researchers are in the room. Sometimes, test subjects would say what they thought the researchers wanted to hear rather than what they actually thought. Sometimes, the researchers would give subtle non-verbal cues that neither they nor the subjects would consciously notice, but that would bias the results of the test. For example, if the experiment is a taste test between two leading soft drinks, if the researchers know which drink is which and they know which one they *want* the test subjects to pick, they might suddenly find that all of the test subjects are choosing the "right" drink… but only because the researchers are (accidentally) telling them what to choose!

The solution to this is the so-called **double-blind** experiment, where neither the researchers nor the test subjects know what the "right" answer is. This eliminates many sources of accidental bias, making the results of an experiment more valid.

Playtests share a lot in common with science experiments. In both cases there is a hypothesis ("I think this game is fun"), an experiment is designed (building a prototype), the experiment is run (the playtest), results are analyzed. The purpose of both is to find out more information about the inner workings of the system that you are studying.

Blindtests, then, are the only way to get a true idea of what your game is like "in the wild" – that is, how players will react to it if they have just purchased it from a store and are playing it for the first time, without a designer present at their table to answer questions.

When to Blindtest?

Because you can get so little information from blindtesting, it is not suitable for an early test when your game is in a rough state (the testers would likely run into problems, be unable to continue, and you would have spent a lot of time waiting for something that you could have figured out much faster with an in-person playtest. **Blindtesting is most suitable near the end of development, when you already have a high degree of confidence in your game**.

The purpose of blindtesting is to catch the non-obvious problems that you may not be catching in your in-person playtests, because you accidentally bias the results of the test by helpfully being available to answer questions. Even if your playtesters never have to ask you anything, the very knowledge that you *could* can sometimes make people relaxed enough to get through the rules, where they might otherwise get flustered and give up. Or, you might say some key piece of information when introducing the players to the game ("this is an auction/bidding game") that is not written anywhere and clarifies a lot by creating some preconceived notions in the minds of your testers. Without you present, you'll see just how accurate your in-person playtests are, and you may catch some surprising errors that you did not notice before.

Who to Blindtest With?

Most board games require multiple players. For practical reasons, most blindtests done with professional games are done with regular game groups. Some companies keep a list of volunteer groups and put out calls to their private list for playtesters; they will then send out advance copies of their game in exchange for feedback. For our purposes, this is unhelpful, because most of you do not work at such a company and do not have a database of volunteers to call at a moment's notice.

Ideally, your blindtest should be with people who are in the target audience for your game. If you're making a children's game, your blindtest should involve kids in your target age range (and perhaps their parents). If your game is made for people who are already avid Eurogame players, you'd do well to find a group that plays these kinds of games regularly. And so on.

There are a few ways to find blindtesters for your Design Project in this course:

- Your social network of friends, family, and colleagues. These people may be local, or they may live in another city or even another country from you. Since you don't have to be present, at least this time, geography is not a barrier.
- Other people who are taking this course. If you already have a local playtest group that you've used before, you can put out a call for help on Twitter or the forums. Offer a blindtest exchange: you'll blindtest their game if they will return the favor with yours.
- As a last resort, you can look outside of this course for other discussion forums or other online hangouts for board game designers and/or playtesters, and see if there is a method for recruiting blindtesters.

How to Blindtest?

With in-person tests you can afford to be a little bit sloppy. Maybe you accidentally left some of your game components at home, or some of your rules are a little unclear and need clarification. It's not ideal, but a playtest session that runs into problems can be salvaged when you are there. With blindtesting, you do not have this luxury, so you must be extra careful to make sure that your testers have everything they need to give your game a proper playtest. This includes:

- A complete set of game components. Double and triple check to make sure that you have everything the players need, together, in one package.
- A list of everything that your package should contain, so that the testers can verify for themselves that nothing was left out (and if it was, it will at least give them a clue of what they need to supply as replacements).
- A complete set of rules describing how to play. (The components list can simply be part of the rules.)
- A separate set of instructions on how to conduct the blindtest. Is there anything in particular you are looking for the players to do? Do you just want them to see if they can play through the game? Do you want to know if they find the game enjoyable? Do you want them to try to find rules exploits and imbalances?
- A final set of instructions on what should be done on conclusion of the playtest. How should the testers contact you (phone, email, etc.)? What should they tell you when they

contact you – if you don't give a list of questions for them to answer, you will just get whatever they happen to feel like telling you, making your results a lot less focused than they could be. Give some thought to what information would be the most useful to have, what kinds of feedback are most important to you… and ask for it!

How do you get your package into the hands of the playtesters? If they are local to you, it is as simple as taking your playable prototype and handing it off. If your blindtesters are not local, this presents additional challenges. You have two options here.

First, you can mail your prototype to them through the postal service. Depending on where they are, this step alone can slow things down considerably (not to mention making it more expensive), so plan your schedule accordingly. If you want your prototype back, consider including a return package inside the main one, already addressed to you and with postage already paid.

Second, you can handle everything over email. Send documents that can be printed out and assembled to make a playable prototype, and include instructions on how to print everything. Do everything you can to reduce the amount of work that must be done at the playtesters' end of things. After all, you are already asking for their time in the form of a playtest; asking for another hour or two to print and cut sheets of cards is adding insult to injury. Along the same lines, try to keep the cost of materials down and availability high if you're expecting your blindtesters to provide their own. For example, do not insist on printing on heavy card stock (which may require a special trip to a print shop) when printing on plain paper will do.

Contingency Planning

In the field, game companies do not rely on a single blindtest group, but several. Aside from the obvious reason that more tests give more data, there is also the problem that blindtest groups are unreliable. Without you in the same room, they may take awhile to organize a playtest, and they may take even longer to get back to you. Some groups may lose the components, or forget about their obligation, or they may simply be so busy with other things in their life that your game takes lower priority. Or maybe you'll send your game through the post and it will get lost. In this course, when you set up a blindtest, you may find yourself in the frustrating situation of waiting for test results that are simply not coming… or at least, they may not arrive within the schedule you set for yourself.

You have three options for dealing with this potential hazard. It is up to you which method best fits your personal situation.

1. You can set up multiple blindtest groups. As a rule of thumb, three is a fairly safe number. That way, if one or two groups don't show, you'll at least have *some* results. (Note that if you are doing a "blindtest exchange" with other participants of this course, that means you'll be testing three other projects, so make sure you have time for this.)

2. You can choose a blindtester that you know and trust to be reliable. If you are sending your game to a friend who is highly organized and always keeps their promises, you may have confidence that they will get back to you when they say they will.
3. You can cross your fingers and hope that something bad won't happen to you. This third method is not recommended for professional projects.

Homeplay

Your homeplay this past Thursday was to arrange for a playtest session with non-designers. You may have already performed this playtest, or you may have just scheduled it to take place over the first part of this week, but that playtest session should be concluded before this Thursday, August 20, noon GMT.

In addition, you should **find a blindtest group and arrange for a blindtesting session, to take place *after* the non-designer playtest.** If possible, you should plan to have blindtest results on or before Thursday, August 27.

Feedback

Do you know of any great articles on blindtesting? As you conduct your own session, or you're your own personal experience if you've done this before, have you found any helpful tips or tricks that you'd like to share? Post in the comments on this blog, or on Twitter with the #GDCU tag.

Level 16: Game Balance

By ai864

When veteran gamers or game designers are playing a game, if they are doing too well or too poorly, they will often comment on the game's **balance**. This word is important, but I fear it is often overused. Like the word "fun," there are different kinds of balance, and understanding what game balance is and why it's important is what we cover today.

Why are we only covering this now and not earlier (like, say, at the *start* of the Design Project)? As mentioned earlier, balancing the game is something that is best left until *after* you have a good set of core mechanics. Balancing a game that is simply not meeting its design goals is a waste of time, and when you change the core mechanics you'll just have to balance the game again. So here we are, with a work-in-progress that has survived multiple rounds of playtesting, and it is time to take it to the next level.

About the Pace of this Course

At this point, I realize from the comments and some forum posts that many of you are falling behind. As a reminder, this course is going at a brisk pace, and I recognize that many participants may not have the time to devote to this one hundred percent. On the bright side, the playtesting part of designing a game can go at a fast or slow pace; for your own hobby projects, time may not be of the essence, so you can continue to follow along at your own pace.

If you can still keep on schedule, that is great. If you are finding that setting up playtest sessions and modifying your game is taking too much time, my suggestion would be to do things as soon as you are able, and then follow along through the rest of this course a little behind schedule. I'm not taking this content down any time soon, so it will patiently wait for you. The rest of the course will concentrate on the Design Project and there will be no other homeplays to distract you.

I would recommend keeping up with the readings on this blog, however, if you have the time. This will give you an idea of what to look for in your project as you look ahead, and I may make announcements here that I'll want you to see.

Readings

No additional readings for today. There is plenty of material here in this blog post.

What is Game Balance?

In a two-player game, saying it is "balanced" usually means that one player does not have an unfair advantage over the other. But we also hear the term used in relation to single-player games, where there is no opponent other than the game itself, and any "unfair advantage" the game may have could just be considered a challenge. And then we may talk of individual cards in a game like *Magic: the Gathering* as being "balanced" even when all players have access to that card, so it does not give an advantage to any individual. What's going on here?

In my experience, when we talk of "game balance" we are generally talking about one of four things. Context usually makes it clear which one we are talking about:

1. In single-player games, we use "balance" to describe whether the challenge level is appropriate to the audience;
2. In multi-player games where there is **asymmetry** (that is, where players do not start with exactly equal positions and resources), we use "balance" to describe whether one starting position is easier to win with than another.
3. Within a game, if there are multiple strategies or paths to victory that can be followed within the game, we use "balance" to describe whether following one strategy is better or worse than following another.
4. Within a system that has several similar game objects (such as cards in a trading-card game, weapons in a role-playing game, and so on), we use "balance" to describe the objects themselves, specifically whether different objects have the same cost/benefit ratio.

Let us examine each of these more closely, and then we will go over some practical techniques for balancing your game.

Balance in Single-Player Games

In single-player games, we use "balance" to describe whether the challenge level is appropriate to the audience.

Note that, by simply playing the game and getting experience with it, your audience will eventually become more skilled at the game. This is one reason why the later levels of video games are usually harder than the earlier levels. (Recall that another reason is so that the gameplay matches the dramatic tension in the narrative.) The change in difficulty over time in a single game has a name: we call it **pacing**.

There is one obvious problem here that we face as game designers: how do we know what an "appropriate" challenge level is? Sure, we can say that a logic/puzzle game for adults is probably going to be harder than a similar game for young children, but beyond that… how are we supposed to know what is too easy or too hard? The obvious answer: playtest!

There is another problem, however: not all players are exactly the same. Even within a narrow target audience, players fall along a bell curve, with a few that will be highly skilled and a few that are the opposite. In your playtests, how do you know where your testers fall on that bell curve? If you are just starting out, the ideal thing to do is to use *lots and lots* of playtesters. When you have dozens or hundreds of people giving feedback on your game, you can get a pretty good idea of what the overall ranges are. As you gain experience as a game designer, you will get a better feel for your audience, and you will need progressively fewer and fewer playtesters to give you the same good results. (If you're starting out but you don't have the time or resources to do lots of playtests, you can sometimes fake it if you have some idea of where your own playtesters fall on the curve. Do they consider themselves above-average or below-average skill level, compared to the other kinds of people you're making your game for?)

Even if you have a pretty good idea of how to modify the difficulty of your game and where your target audience falls, what do you choose as your challenge level when the audience is diverse? No matter what you do, your game will be too hard for some people and too easy for others, so this appears to be a no-win situation. If you must choose a single level of challenge, a rule of thumb is to aim for the middle of the curve, as you will get the most people (the widest possible audience) that way. Another way around this is to provide support for those at the ends of the curve, using multiple difficulty levels, handicaps, or alternate rule sets to make things easier or harder.

Balance in Asymmetric Games

In multi-player games where there is asymmetry (that is, where players do not start with exactly equal positions and resources), we use "balance" to describe whether one starting position is easier to win with than another.

Truly symmetric games are rare. We think of classic games like *Chess* and *Go* as symmetric, since each player starts with the same set of pieces and plays by the same rules, but there is one asymmetry: one player goes first! If you modify *Chess* so that both players secretly write a move on a piece of paper and then the moves are performed simultaneously, it becomes fully symmetric (and plays very differently).

This brings up an interesting question: if your game *is* symmetric, do you need to worry about game balance at all? After all, both players start with exactly identical resources and starting positions and so on, so by definition no player can have an unfair advantage. This is true, but the designer must still consider other types of balance, particularly whether there is a dominant strategy. Simply making all players equal does not get you off the hook.

Even if your game is asymmetric, why bother balancing it? The simple answer is that it is a typical player expectation that a game will not give an automatic advantage or disadvantage to certain players, other things being equal. (You can play around with this. The card game *The Great Dalmutti*, for example, intentionally casts players in unequal roles as a way of showing how life isn't fair; but that is part of the game, and the instructions and mechanics go to great lengths to set player expectations accordingly. But if your game is not breaking this rule with specific intent, you should be thinking about how to make it as balanced as possible.)

Asymmetric games are, naturally, harder to balance. The more asymmetric, the more carefully the game must be playtested carefully. One of the easiest ways to confirm this kind of balance is to find ways of relating each players' resources to one another. If you determine that in gameplay, one Apple is always worth exactly two Oranges, then a player who starts the game with an Apple will be balanced against a player starting with two Oranges.

Sometimes, players are so different that direct comparisons are impossible. Some games not only give players different starting resources and positions, but also different rules to play by. Some players may have exclusive access to certain resources or abilities. One common asymmetry in games is to give players different and conflicting objectives (for example, one team's objective is to survive for some number of turns and the other team's objective is to eliminate the first team before that many turns). The more difficult it is to make direct comparisons between players, the more you will have to playtest to compensate.

Balance between Strategies in a Game

Within a game, if there are multiple strategies or paths to victory that can be followed within the game, we use "balance" to describe whether following one strategy is better or worse than following another.

One might wonder, why bother with this? If a game allows for multiple strategies but one is more powerful than the others, doesn't exploiting the best strategy just equate to players trying to win? As long as no individual *player* has an unfair advantage, isn't it okay for your game to simply be "whoever finds the most powerful strategy wins"?

The problem here is that, once a dominant strategy is discovered, astute players will ignore all suboptimal strategies. Everything in the game that is not part of the dominant strategy becomes extraneous noise. There is nothing inherently wrong with a game that has a single winning strategy, but in this case the suboptimal ones should be removed to make the game more streamlined. If you include options for players that are suboptimal, these become false decisions, because really there is only one decision (follow the dominant strategy).

If it is worth including several potential winning strategies in a game, then, it becomes much more interesting if those strategies are balanced. Again, much of this comes down to playtesting. In this case, when players are playing your game, make note of whether certain strategies seem to be used more often than others, and which ones seem to win. If several items are available for players to purchase in a game, is there one that seems to always get bought early, while others seem to be used rarely if ever? If players have a choice of actions each turn, does the winner of each playtest always seem to be the one that chose one particular action more often than everyone else?

Playtesting alone is not automatic proof that a particular strategy is unbalanced, but it should give you strong signals that certain aspects of the game need closer inspection. Sometimes, players will use a particular strategy because it is the most obvious or the easiest and not because it is the most optimal. Some players will avoid anything that seems too complicated or requires finesse, even if it is ultimately better in the long run.

Balance Between Game Objects

Within a system that has several similar game objects (such as cards in a trading-card game, weapons in a role-playing game, and so on), we use "balance" to describe the objects themselves, specifically whether different objects have the same cost/benefit ratio.

This kind of balance is specific to games that give players a choice between different game objects. Some examples:

- Cards in a trading-card game. Players build a deck with a set number of cards from their collections. The choice of *which* cards to add is one of the key factors in the game's outcome, and designers try to make the cards balanced with one another.
- Units in some war games and real-time strategy games. Players have the ability to purchase units during play, and different kinds of units may have different abilities, movement rates and combat strengths. The designer will try to make the units balanced with one another.
- Weapons, items, magic spells, etc. in a role-playing game, either tabletop or computer/console. Players may purchase any of these for use in combat, and they have different costs and different stats and abilities. The designer will try to make these objects balanced with each other.

In all of these cases, there are two goals. The first is to prevent any game object from being so weak that it is useless in comparison with other objects. This again becomes a false choice for the player, because they might *be able* to gain or purchase a certain object but they will quickly find that it is not worth using; the object is therefore a waste of the player's (and designer's) time.

The second goal is to prevent a game object from being too powerful. Any single game object that becomes a dominant strategy makes *all other objects in the game* useless in comparison. In general, if you absolutely must choose between making an object too weak or too powerful, err on the side of making it too weak.

Two objects are balanced if they have the same **cost/benefit ratio**. That is, what you give up to gain access to an object (this includes explicit costs like in-game money or resources, and also opportunity costs like drawbacks, limitations, or exceptions to the object's capabilities) should be in some proportion to the in-game benefits you get from the object. The costs and benefits do not have to be exactly the same (in fact, usually the benefits are greater, or else you would simply ignore the object). However, when comparing two *different* objects, the proportion of costs to benefits should be roughly the same for each.

Three Ways to Balance Game Objects: Transitive, Intransitive, and Fruity

I have encountered three general methods for balancing game objects. The first is technically referred to as a **transitive relationship**. In more colloquial (but still geeky) circles, it is called a **cost curve**. This is the most direct way to balance objects. The general idea is to find some desired proportion of costs to benefits. This may be a linear proportion (something twice as costly is exactly twice as powerful) or it may be curved in some way (perhaps there is a law of diminishing returns, where you have to pay progressively more for each additional bit of benefit; or perhaps there are increasing returns, where you essentially get a "bulk discount" for paying a lot at once). It all depends on your particular game, but playtesting, experimentation and instinct will help you to figure out what kind of relationship there should be.

The next step is to reduce every cost and every benefit to a single number that can be compared. Take all the costs of an object and add them together; also sum the benefits. Compare the two, and see if the object is giving the correct numerical benefit for the cost.

This method is often used in trading-card games. If the game has an established cost curve, it makes it much easier to create new cards with combinations of existing effects. In *Magic: the Gathering*, if you want to create a new creature with a given color, power, toughness, and set of standard abilities (say, a White 4/3 with Flying and First Strike), there are already several costs it can be, and designers working on this game (and sufficiently informed players) could tell you exactly what those equivalent costs are. Adding more abilities comes with an increase in cost, and decreasing the cost would necessitate a removal of stats or abilities.

The second method is an **intransitive relationship** between game objects, better known as a **rock-paper-scissors** relationship. In this case, there may not be a direct relationship between costs and benefits, but there is a relationship between the game objects themselves: some objects are inherently superior to others and inferior to still others. The game *Rock-Paper-Scissors* is the canonical example; none of the three throws is dominant, because each throw will draw with itself, beat one of the other throws and lose to the third one.

This can be seen in some strategy games as well. In many real-time strategy games, there is some kind of intransitive relationship between units. For example, one common relationship is that infantry are strong against archers, archers are strong against flying units, and fliers are strong against infantry. Part of the game is managing your particular allocation and positioning of units (in real time) in comparison to your opponent.

Note that transitive and intransitive relationships can be combined, as in the previous example. In typical real-time strategy games, units also have different costs, so a weak (but cheap) archer may still be defeated by a strong (and expensive) flying unit. Within a single class of units, there may be transitive relationships, but the different classes have intransitive relationships with one another.

Intransitive relationships can actually be solved, using matrices and some basic linear algebra. For example, the solution to rock-paper-scissors is that you expect the proportion of each throw to be equal to the others: there should be a 1:1:1 ratio. Now, suppose you modify the game slightly, so that each win with Rock scores 3 points, a win with Paper scores 2 points, and a win with Scissors scores 1 point. What is the expected ratio now? (It turns out the ratio is not what you'd expect; with optimal play on both sides, you would see 1 Rock for every 3 Paper and every 2 Scissors. The math required to do this is outside the scope of this course.) If you are looking for players to use objects in a certain proportion (with some being more commonly used than others), a well-balanced intransitive relationship is a good way to guarantee this.

A third method of balancing game objects is to make each one so different and unique from the others, that direct comparisons are impossible. (I call this "fruity" in the sense that the designer, and later the players, can only compare apples to oranges.) Since formal and numerical comparisons between objects cannot work, the only way to balance this is through excessive playtesting.

There are challenges associated with all three of these methods. For transitive relationships, everything relies on the designer finding the correct cost curve. If your math is wrong, it will be wrong for every object in the game; if you find one thing that is unbalanced, you'll probably have to change *everything*. Transitive relationships are much easier to develop in retrospect after playtesting, than developing them ahead of time. Since so much relies on getting the math right, it also tends to take a lot of trial-and-error and therefore a lot of time.

Intransitive relationships, as noted above, take some tricky math to solve. Another drawback is that, unless done very carefully, their presence can make the entire game feel like glorified *Rock-Paper-Scissors*, which some players find to be a turnoff – many have the perception that intransitive relationships are nothing but guessing games, where every decision is based not on strategy but on luck and randomness. (A full discussion of whether this is or is not the case is also outside the scope of this course.)

"Fruity" relationships are really hard to balance, because one of the most important tools in doing so – mathematics – is no longer available.

Three General Game Balance Techniques

In general, there are three ways to balance games:

- Use math. Create transitive or intransitive relationships in your game, and make sure that everything is in line with the cost.
- Use your instincts as a game designer. Change the balance in the game until it "feels right" to you.
- Use playtesting. Adjust the game based on the results of playtests, where the players are experienced gamers who have been instructed to play to exploit and win.

There are challenges with each of these ways:

- Math is hard, and it can be incorrect. If your formulas are wrong, everything in the game may be off a little bit, which is inconvenient for rapid prototypes. Some really strange abilities or game objects may not *have* any math to them if they are too unique, requiring other ways of balancing.

- Instinct is vulnerable to human error. It is also not absolute or reproducible; different designers may disagree on what is best for the game. This is particularly dangerous on large team projects, where one designer may leave in mid-project and another cannot take over (or rather, they *can*, but they will not be able to finish the game in the same way that the original designer would have).
- Playtesting relies on the quality of your testers. Testers may not find every balance issue with the game; some problems will go undiscovered for months or years (even after public release of the game). Worse, some testers may *intentionally* avoid showing you rules exploits, because they plan to use them after the game is released!

What is a designer to do? Do the best you can, and understand both the strengths and limitations of the balance techniques that you are using. And as a game *player*, the next time you run into a game that seems horribly unbalanced, have some appreciation for how difficult it can be to get things perfect.

More Game Balance Techniques

Here are a few other random bits of advice I've picked up, in no particular order.

Be aware of the different objects and systems in your game and their relationships. You should already have done this during your initial design of the game, of course, but it is easy to forget the big picture when you start focusing on small details. There are two things in particular that you should return to first, whenever you make changes to your game:

1. What is the core aesthetic of your game? Does this change support the core?
2. Look at the interconnections between systems. If you change one thing, you should know what other things will be affected. Individual game elements rarely exist in a vacuum, and changing one thing can have ripple effects throughout the game. By being aware of the relationships between systems and objects, it becomes easier to predict the second-order effects of a mechanics change.

Make one change at a time. We've said this before, but it bears repeating. If something breaks after making a change, you know exactly why. If something breaks after making ten changes, you don't know which change (or combination of changes) caused it.

Learn to love Excel. It can be any computer spreadsheet program, though Microsoft Excel is the most popular among game designers. Often, students look at me like I'm crazy when I suggest that a spreadsheet is useful in game design. (Like, aren't those things only used by corporate finance dudes or something?) Here are some examples of how spreadsheets are used:

- Excel makes it easy to keep lists of things and organize them. List all of your game objects and their stats. In a role-playing game, list all weapons, items and monsters; in a tabletop war game, list all units and their stats. Anything that you'd find in a reference chart in the instructions (or a strategy guide) probably started off its life in a designer's Excel sheet.

- Excel is great for keeping track of tasks and status, which is useful for a complicated game with lots of systems and components. If you've got a table with a couple hundred monsters and all their stats, it might also have an entry for whether the art for that monster is done, or whether the stats for that monster have been balanced or playtested yet.
- Spreadsheets are good for collecting and manipulating statistics in your game. In a sports game where each player has a list of stats, are all of the teams balanced with one another? Sum or average all of the stats on the team, and you can get some idea of the overall strengths and weaknesses of each team. In a game with transitive relationships, is each game object balanced? Add up the costs and benefits in a spreadsheet.
- You can use spreadsheets to run statistical simulations. By generating random numbers (in Excel you can use the *RAND()* function and press F9 to reroll), you can generate random die-rolls for things like damage within a range, many times, to see the overall range and distribution of outcomes. (Statisticians call this a "Monte Carlo" simulation, in case you were wondering.)
- Spreadsheets help you to see causes and effects of changes in the game. By creating formulas based on specific values that you want to change, you can change one value and see what happens to the *other* values that depend on that one. For example, if you're working on a Massively-Multiplayer Online RPG, you could use Excel to compute the damage-per-second of a weapon and then instantly see how that changes when you modify the base damage, accuracy, or attack speed.

Use the Rule of 2. Suppose you have some number in your game that you know is too high, but you don't know how much. Maybe it's *just a little bit* too high, or maybe it's quite a bit off. In either case, cut it in half. Likewise, if you have a value that you know is too low, regardless of *how much* too low it is, double it. If you aren't 100% sure of what the correct value is, double it or cut it in half. This is the "Rule of 2."

At face value, this sounds rather ridiculous. If the cost of a gemstone is only 10 to 20 percent too low, what could be gained by taking the drastic measure of *doubling* it? In practice, there are some reasons why this works. First, you might *think* that it's only slightly off, and you might be wrong; if you only make minor adjustments and the value really did need to be doubled, it will take you much iteration to get to where you needed to be in the first place.

There is a more powerful force at work, though, when applying the Rule of 2. Game design is a process of discovery. The fact is, *you don't know* what the correct numbers are to balance your game; if you did, the game would be balanced already! If one of the values in your game is off, you need to discover what the correct value is, and you do this by changing the value and seeing what happens. By making a major adjustment, you will learn a great deal about the effect of this value on the game. Maybe it *did* only need a minor adjustment, but by doubling or halving the value, you will learn so much more about your game.

Occasionally, you will also find that by making such a large change to your game, it changes the dynamics in a way that was unexpected but (accidentally) superior to your original design.

Balancing the first-turn advantage. In turn-based games in particular, it is common for there to be a very slight advantage (or disadvantage) to going first. This is not always the case, but when it is, there are a few common techniques for compensating:

- Rotate who the first player is. In a four-player game, for example, after each complete round (where every player has a turn), rotate the starting player to the left for the next round. In this way, the player who goes first on this round will go *last* on the next round. (When I was growing up, my game group used a pencil to mark the first player, so we dubbed this the "Pencil of Power" technique.)
- Give the disadvantaged players some extra resources. For example, if the objective of the game is to score the most points by the end of the game, give each player a different number of points to start the game, with the last player having slightly more points to compensate for the disadvantage of going last.
- Reduce the effectiveness of early turns for the first players. In a card game, maybe players typically draw four cards at the start of their turn. You could modify this so that the first player only gets to draw one card, the next player draws two, and so on until all players are drawing four.
- For very short games, play a series of games where each player gets to go first once. This is common with card games, where a complete game is played in a series of hands.

Write down your own rules as you learn them. As you design games, you will have successes and mistakes. You will learn from both (especially your mistakes). When you find a "law" of game design or a new game balance technique, record it and review your notes periodically. Tragically, very few designers actually do this; as a result, they cannot pass on their experience to other designers, and they sometimes end up making the same mistakes on multiple games because they forget the lessons learned from earlier ones.

The Role of Balance

In my early days as a designer, I was very passionate about game balance. "A fun game is a balanced game, and a balanced game is a fun game" was my mantra. I'm sure I annoyed many of my seniors with my rantings on the imbalances in our games and why we needed to fix them immediately. Such is the privilege of youth.

While I still believe balance is important and a key skill for any game designer, I now take a more moderate line. I have seen some games that are fun in spite of being unbalanced. I've seen other games that are fun *specifically because* they are *intentionally* unbalanced. I have seen some games that have done astoundingly well in the marketplace, despite having egregious imbalances. These are rare and special cases, to be sure. But let this serve as a reminder to you that the techniques and concepts discussed today should never take priority over the ultimate design goals for your game. Let these techniques be your tools, not your master.

Homeplay

Your homeplay this past Monday was to arrange for a blindtest session, to be completed on or before next Thursday (August 27). Continue working on this if you have not completed it already.

Your other task, **before next Monday (August 24)**, is to critically analyze your game **with respect to game balance**. Playtest the game once (either on your own or with others) with the purpose of finding balance issues – instruct everyone to intentionally seek out optimal strategies and exploit them. Play to win. Be vicious.

Then, from there, think of what systems and game objects are in need of modification (sometimes this is called **tuning**). Think about what techniques best fit your game. Are there transitive or intransitive relationships? Is it more suitable to balance primarily with playtesting, math, your own instincts, or some combination of the three?

Lastly, decide on the most serious game balance issue you identified in your earlier playtest. M**ake at least one change** and playtest again (you can probably do this in the same test session). Did you fix the problem, or at least reduce its severity? Then, think about what other changes you might make, and what you will look for in future testing.

Level 17: User Interfaces

By ai864

If you have made it to this point, your game is hopefully coming along nicely and you are approaching the final stages of design. You may still have temptations to mess around with the mechanics, especially given the short time period allotted for the Design Project, but at this point I want you to settle on something that is at least "good enough" so that you can experience the later stages of design.

Once your mechanics are complete and the game is playable, balanced, and meets your design goals, the last thing to do is figure out how to construct the final version. This is not simply a matter of drawing up some art for your cards and board and sending it off to the printer. There are some considerations to be made concerning the user interface of your game, and those considerations are what we will talk about today.

Readings / Viewings

Read the following:

- *Challenges for Game Designers*, Chapter 16 (Creating a User Interface)
- *Cooperation and Engagement*, a Google tech talk by game designer Matt Leacock. This video actually ties together a lot of the concepts we've talked about in this course, from difficulty levels and flow states to the iterative process to game narrative, and it should serve to solidify those concepts using the concrete example of *Pandemic*, one of the best-selling hobby games of last year. But what I really want you to pay attention to is how Matt presents his iterative work on the design of the game components themselves, such as how he determined the shape, orientation and color of the cards. The actual talk is only 30 minutes long, although there are an extra 20 minutes at the end from audience Q&A.

What is a User Interface?

Normally, we think of the term **user interface** (or **UI**) as it applies to software applications. This term refers to the parts of the software that interact directly with a human. It covers things like what options are available to the user at any given time, how those options are presented on the computer screen, as well as the physical interactions (mouse/keyboard, game pad, etc.). In general, with video games, the UI is divided into two parts: the **input** (that is, how the player gives commands to the game) and **output** (how the game communicates the results of those actions and other aspects of the game state to the player).

What if you're making a non-digital game? Is there a UI? Of course there is – and if anything, it is more important to get it right, since you don't have a computer automatically enforcing the game rules. If the players don't understand the rules of a non-digital game, the only recourse is often to stop playing. As the game's designer, the last thing you want is for your brilliant mechanics and carefully-crafted play experience to be ruined because of an interface issue.

In non-digital games, the UI is the game components themselves, since they are both how you interact with the game (through manipulating the game bits) and receive feedback (by viewing the game state). So what we are really talking about today is designing your final components.

User Interface Design

What makes one UI better than another? We generally talk of two things:

- Ease-of-use. If you already know what you want to do, how fast and easy is it to perform your desired task correctly?
- Ease-of-learning. If you are new to the game, how easy is it to figure out what you are allowed to do and what information is available to you?

In practice, there is often a tradeoff between these two. For example, on computers, the presence of special "hotkeys" (Shift/Alt/Ctrl/Cmd key combinations and the like) saves time by making it fast and easy to perform common tasks like saving a file or switching between applications. But if these are the *only* way to perform that task (as is the case with some older word processors that lacked menus), it makes the applications difficult to learn for the first time.

In board games, you often see this tradeoff with how information is presented. Charts, tables, keywords, special symbols and icons – all of these make it much easier for an experienced player to get a quick summary of the game state, but they can be confusing and intimidating to the novice player who does not know what these things mean. The alternative, writing everything out in longhand, makes it much easier for a new player to see immediately what everything does… but it also makes the game take longer for players who already know the rules and do not need them repeated in full on every card.

Sometimes you can include both. Modern software applications include hotkeys *and* menu items, and some even have a "beginner" mode that hides advanced functionality to keep the menus simple, making the software much easier to learn. Card games like *Magic: the Gathering* include keywords, but then explain those keywords in parentheses for players who do not know what the keyword means.

Look at all the mechanics in your game, and what players must do in order to follow them. Are there any cases where your wording could be clearer for first-time players? Are there situations where experienced players of your game feel like they are doing excessive book-keeping or note-taking, or performing multiple automatic steps, where it would be possible to streamline the experience?

The Two Models of Usability

In usability for computer applications, there are two related concepts: the **user model** and the **program model**. The user model is how the user (i.e. the player) thinks that things should work. The program model is how the software *actually* works. (In non-digital games, you can think of the "program model" as the actual rules of the game as intended by the designer, and the user model is what the players *think* the rules are.)

Here's the thing. **The program model is always right.** If the user model and program model match one another, there is no problem. If the two are different, the player is going to do something, they'll expect one thing to happen, but another thing happens instead. With computer games, this leads to frustration, and reviewers will say the game has "poor play control" (often without fully understanding why).

In board games, if the player model and the "program" model are different, the players will just play the game incorrectly. Sometimes, this means the players will have a suboptimal experience, because some aspects of the game will feel unbalanced. Other times, the players will have a perfectly good time, but when they later play the game with *other* players who are playing the game the "right" way there will be rules arguments.

Changing the User Model

It is common to see a user/program model mismatch during playtesting. Here is what it looks like: with every playtest group, the players always do one particular thing *wrong* the first time around. This suggests an ease-of-learning problem.

A more serious case is an ease-of-use problem with the UI of your game. It looks like this: one or more players accidentally do something wrong in the rules. You point it out to them, and they correct their behavior. But then they forget, and they *do it wrong again* on the next turn. And the next. And the next. And your players apologize to you for continuing to get it wrong, and they feel like idiots.

In cases like this, it would be ideal to change the user model. That is, you'd like to change your players' expectations or actions in order to match the "correct" model of the game itself. Unfortunately, in practice, this is effectively impossible to do. People are creatures of habit. We build up elaborate mental models of the world around us, and we rely on those models greatly. Changing a model is a slow process that requires great effort on the part of a person, and most people are not going to put that kind of effort into your game.

To illustrate this in my classroom courses, I tell the story of the design of a fighter jet. Once upon a time, the military was noticing that one particular type of aircraft had a much higher-than-average number of accidental pilot ejections (that is, the pilot's ejection seat was activated when the pilot did not intend for that to happen). Given the cost of military aircraft, it gets very expensive when something like this happens, so they had a lot of engineers study the problem to figure out why the seat was ejecting, but no mechanical or electrical problems could be identified. Eventually, someone got the bright idea to look at the aircraft that the accidentallyejected pilots had trained on. It turned out that in all cases, they had trained on an aircraft where the positions of the throttle and the ejection seat controls were reversed! So, when the pilots got in this new aircraft, they had already formed a mental model of where all the controls were, and no amount of training on the new aircraft could change it.

Identifying the User Model

Okay, so we can't change the user model. That means if you find a mismatch between the user model and the game, you should change the game's interface so that it either conforms to the user model, or change the interface so that it is different enough that it triggers a *different* user model. But how do you know what the user model is in the first place?

Thankfully, this is pretty easy to do. The easiest way is to ask. Find people who play games similar to the one you're making. Set up a situation for them, and ask them how they think the game would work (or how they would accomplish some task, or whatever you're trying to figure out) if they had to guess. Once you ask a few people, a clear consensus will emerge pretty quickly.

Another pretty easy way to identify the user model is to playtest. Watch when people are playing the game, and make note when they do something wrong.

Lastly, if your game's model is nontrivial, it is probably wrong. Other things being equal, people will assume simplicity.

Whose Responsibility Is It, Anyway?

You might wonder, in some cases, if usability issues are really your problem as the game designer. After all, if you create a good game and you give a complete and correct set of rules, isn't it the responsibility of the players to actually, you know, *read the rules and follow them*? If they play your game wrong, how is that *your* fault? Some people are just stupid or careless, and certainly the brilliant designer should not be held accountable for these people. Right?

Well, first off, it may not be the player's fault. They might have been taught by someone else. They might be in a distracting environment, so they can't give the rules their undivided attention. They might not *have* the rules; maybe they purchased the game second-hand and the rules were missing. The language the rules were written in might not be their first language. There are any number of reasons why a perfectly intelligent and rational person might have difficulty. So, don't just write these people off as not worth your time.

Second, most usability failures *feel* like a mistake on the part of the user (player), but they are *actually* a UI issue that could be fixed. Consider: if your game were easier to use, it wouldn't *let* the players make mistakes. As the designer, take responsibility for the usability of your game, and you will find that players are learning it faster, making fewer mistakes, and generally having a better time.

Building a Good UI

Now that we know how to identify a bad UI, how do you design a good one? In general, a good UI does two things:

- It does what the user thinks it will do; and
- It gives the user feedback so they *know* what it did.

If the game doesn't do what the player thinks it will, that is a problem with the user model not matching the game model, as we've already discussed. But there is one other aspect to designing the UI: giving the player immediate feedback so that they know what they did is correct (and, in the unlikely event that they did something wrong, they will immediately *see* that it is wrong and understand why).

Here is another way of looking at what a good UI does:

- Make it easy to do things correctly; and
- Make it hard to do things the wrong way.

Here's an example: suppose you have a board game with several kinds of tokens. Maybe you have one set of markers that keeps track of player score, using a scoring track around the edge of the board. Maybe the game board has a map divided into provinces, and players have military units that are placed in the provinces. Maybe there's also a global market with goods for purchase and sale, and a separate track for each trade good that lists its current price.

It would be easy to get all these different game bits confused. But what if each token was a different size and shape, and each space on the board matched the shape of the token that was used there? All of a sudden, it becomes a lot easier to figure out that the small square tokens must go on the small squares of the scoring track, the star-shaped goods markers go on the star shapes of the trade good price tracks, and so on.

How will the players remember how to adjust the trade good values on each track? A rules summary printed on the board right next to the tracks makes it easy to remember. What about how combat is resolved? Unit strengths, stats and abilities can be printed on the military units themselves, and the remaining rules can either be summarized on the board or on a quick-reference card or other player aid given to each player at the start of the game.

As you go about designing the UI, here is a process you can follow:

- First, make a list of tasks that players need to accomplish in the game. Make it easy to do those tasks.
- Second, pay special attention to the most *common* tasks, the ones that players are doing over and over. Difficulty and complexity of a task should be in proportion to how rarely it is used.
- Iterate through playtesting.

Use of Color

Color is a great way to convey information to players if done well. It is efficient: color takes up no additional space on a game component, because the component is already there; all you're doing is coloring it. Here are a few quick-and-dirty tips for thinking about color use in your game:

- The colors that normal human eyes detect most easily are red and green, followed by blue. These colors will tend to stand out more… which can be good for drawing attention, but bad for eye strain if you use fully-saturated bright colors.
- Don't rely on color alone; a nontrivial portion of your audience has some degree of colorblindness. Vary the intensity of colors as well (so that if, for example, photographed with a black-and-white camera, you would still see a difference), and use different shapes in addition to different colors. By having things differentiated in multiple ways (different shape, different color, etc.) it makes it *really obvious* that those things are distinct from each other.
- Use color consistently. If you use one color for several things in a game, those things should be related. Some board games I've played, for example, have (say) five different resources, and each one has a color; but each player *also* has a color, and some player colors and resource colors are the same, even though there is no relationship between the green player and green resources. This kind of thing can confuse players who might think that a particular resource is owned by their opponent when it isn't.

More UI Design Tips

In no particular order:

- Automate or eliminate tasks that don't involve interesting decisions, if possible. Every click or keypress in a video game, or die-roll or card flip in a board game, should involve the player doing something that is interesting to them. If the player first has to do a bunch of upkeep tasks just to have the privilege of making interesting decisions later, see what you can do to streamline the boring stuff.
- Use visual metaphors. They make it obvious what something represents. If your players control a bunch of pieces that represent people, using people-shaped pawns is clearer than using wooden cubes. Compare the pawn images below. Each has a very different effect on how the player views it.

- Likewise, if you use icons in the game to represent certain abilities, choose icons that look like the concept they're representing (if you can).
- Be consistent with similar games. In an RPG, red hearts probably mean life or hit points, while blue drops probably represent mana or magic power. Why? Because that's how everyone else does it, and your players will assume by default that you do it that way too.
- **Don't** just say "well, this is confusing, so we'll explain it in the manual." Remember, your players may not have the manual and they may not read it. Try to make your UI intuitive enough that your game doesn't even *need* a manual.

Lessons Learned

Giving your game a good UI is a skill that is separate from core systems design, but it is an important skill to learn. Keep in mind that, as with most things in this course, UI design is a huge field and we are only going to cover the very basics. There are entire courses (and even college majors) devoted specifically to UI, not to mention hundreds of books, and I encourage you to seek out these resources after this course is over.

Further Reading

There are many great books on UI design. If this topic interests you, I would recommend Donald Norman's *Design of Everyday Things*, which gets into the details of how the design of something as simple as a door or a stove can go horribly wrong… with lessons that can be applied directly to games, both digital and non. Also, for ways to show game data to players in efficient and innovative ways, I would recommend any of Edward Tufte's books: *The Visual Display of Quantitative Information*, *Envisioning Information*, and *Visual Explanations*.

If you are primarily interested in video games, there are so many great sources on computer UI that it would be hard to list them all here. If you have any personal favorites, leave as a comment on this blog post, or post on Twitter using #GDCU.

Homeplay

The ongoing homeplay from a week ago was to arrange for a blindtest session, which should be completed on or before this Thursday (August 27). Continue working on this if you have not completed it already.

Your other task, also before this Thursday, is to critically analyze your game **with respect to user interface**. Think about your playtests so far, and what rules your players have had trouble with. What kinds of components or player aids would make it easier for them to remember?

Come up with a **user interface plan** for the final version of your game. This plan should involve the following:

- A complete list of all game components that will be included in the final game.
- For each component, a detailed description of what you are planning to use. If you have, say, "one pawn to represent each player", how many pawns are included? What colors will they be? What shapes? Will they be made of metal, plastic, wood, or something else?
- If there are cards, describe a sample card. What information will be displayed on each card? Will the card be oriented as "portrait" or "landscape"? How will the information be laid out – where on the card will each item go? How will it be displayed – what colors, shapes, and keywords do you plan to use (if any)? If there are several *kinds* of cards, do this for each.
- If there is a board, describe the layout of the board. As with cards, consider what information will be displayed on the board, how it is laid out, and how it is displayed.
- If there are other components, give similar information to that listed here.

This list is mainly for your own reference, and the purpose is to make it as easy as possible for you to assemble your final game (which you will be doing this next week). The list also serves as a sanity check: do you have access to all the components you need? If you do not own some of them, think about how and where you plan to create or purchase them.

Post your UI plan to the forums no later than this Thursday (August 27), noon GMT.

By next Monday (August 31), **read and respond to at least three other posts at the same level as you**, giving preference to those that have not had any responses. By reading, it may give you some ideas to use on your own project. You may also be able to share some ideas you already have with others, helping them to improve their projects.

Level 18: The Final Iteration

By ai864

We are nearing the end of this adventure, so if you are still here, you deserve some congratulations.

At this point you have created a game. You have playtested your game multiple times, with multiple purposes (from fun to balance to usability). You have a list of materials that you need to assemble, and a list of things you must do to assemble them. All that is left in the Design Project is to complete it.

Readings / Viewings

None.

On Craftsmanship

One might wonder, why bother with assembling high-quality components for a game? The game is all about the mechanics, after all, and the game pieces are just a physical manifestation of the mechanics. Therefore, any pieces should be as good as any other. UI issues aside, why not just use the prototype you've been using and call *that* the final project… handwritten cards and all?

I have several responses to this.

First, if the pieces are a physical representation of the rules, you as the designer should give them the same care, so that the outer beauty of the game matches the inner beauty of the mechanics.

Second, you should take pride in your game. The physical pieces are as much a design statement as the rules. What the pieces say to a player is that the game designer felt that their game was high enough quality to deserve high-quality components. To use an analogy, if you are a chef and you're making a simple peanut-butter sandwich, maybe you don't mind eating off a flimsy paper plate; but if you've put together an elaborate nine-course gourmet dinner, find some better-quality plates to serve it on.

Third, if you plan to eventually commercialize and sell your game, the quality of the components is one of the first things a prospective customer will see. Many Eurogames actually print a nice picture of the game components on the back of the box as a selling point. They are in essence saying, "if you buy this game, you will get these fine components." Even if you aren't planning on selling the game you're making *right now*, going through that process is good practice for future projects.

Lastly, keep in mind that your game represents you as a designer. If you ignore the one thing that every player is going to see, what does that say about you? Does it suggest that you don't care about your projects enough to put in some extra effort? Does it mean you don't take pride in your own work? Does it mean you lack confidence? If you are planning to use this game (or another one) as part of your professional design portfolio, think about how it will appear to prospective schools or employers. For the same reason that you might dress up for a job interview, dress your game up before you release it.

An Anti-Craftsmanship Aesthetic

You might press this even further. Some *commercial* games were made with *intentionally* poor production values. The first printing of *Kill Doctor Lucky* (and the rest of the games from the original publisher) had cards printed in one color on flimsy paper with just simple text and no art. The game didn't even include common components like pawns or dice, and it was sold in a thin paper envelope with the rules printed right on the bag – there wasn't even a separate printed manual. In the digital realm, *Kingdom of Loathing* is an online browser game that uses badly-drawn stick figures for all of the game art. It's as if these people aren't even trying, and yet their games see many players. Why not just do that with your own game?

I would argue that these games actually have very *high* production values. Each offers a consistent visual aesthetic and a strong impact that is consistent with the values of the creator and the game mechanics.

In the case of *Kill Doctor Lucky*, the whole *point* of the entire line of games, in fact James Ernest's *mission statement*, was that games were getting too expensive because they included all these extraneous components. Why bother paying an extra dollar for two dice when you've probably got dozens of dice lying around from other games you own? And sure, the cards are flimsy… but if you accidentally ruin them, oh well, go out and buy the game again. The game cost you less than that mocha latte you had at Starbucks this morning.

With *Kingdom of Loathing*, the game is a parody of similar games. As a parody, the intentionally low-quality art style is a deliberate choice, standing in contrast to the focus on detailed visuals in other games. And any time that didn't go into drawing stick figures went into the writing of the game, which is (frankly) of higher quality than many of the visually stunning games out there.

So, if you have a game where "low" production values make sense – maybe you're making a game about living in a trailer park, or a game about magpies where the object is to build a nest out of the shiniest junk, or something – then by all means, craft your game accordingly. But even in this case (or I should say, *especially* in this case), make the quality and appearance of each component a *deliberate* design decision. You'll find that it may take more skill to pull this off in a way that looks genuine than it would to just make a game with standard high-quality components.

Quick Tips

For printing cards:

- Print on heavy card stock, not standard printer paper.
- If you’re not printing in color, consider at least printing on colored card stock, choosing a color that matches the rest of your game.
- Make sure the cards are readable when printed. Sometimes they don’t print out exactly as they look on a computer screen.
- Use rounded rectangles to mark your cards in a print program, and then carefully cut them out on your own with scissors. Or, if you have a corner-cutter (available at most craft/hobby stores), use that and save yourself a lot of work. Rounded corners are gentler to hold; square corners can poke you uncomfortably.
- If you want to get really fancy, get the cards laminated (this means cutting the corners of the lamination instead of the card itself). This will make the card virtually indestructible and waterproof, although it can be expensive. As a less expensive alternative, some hobby stores sell spray-on plastic coating that will give your cards a similar feel to standard plastic-coated playing cards. Take care to apply the spray evenly so that your cards don’t have lumps or irregularities.
- Print cards double-sided if you can, with a standard card back. Note that the back is a mirror image of the front, so things that print on the left side of the back of the sheet will be printing over the right side of the front. Also, be very careful about double-sided printing, as even a minor offset on one side may make the entire sheet unusable. Try to get the two sides lined up exactly.

For printing a board:

- Don’t just print on card stock alone. It feels flimsier than most game boards you’re used to. Try mounting it on flat cardboard, poster board, or foam core.
- If you have to cut cardboard or foam core to a specific size, try a test cut in a corner first. Some tools will not cut cleanly, but instead will leave the entire edge torn and jagged and ugly.
- Getting a large board printed on a single sheet of paper is expensive. Getting it printed on a vinyl banner is even more expensive. One low-cost alternative is to break up a large board on several standard sheets, then mount them on a firm backing. If you want to get fancier, instead of printing on normal paper, print on a full-sheet adhesive label. Then, you can just peel off the sheet and stick it on somewhere without having to worry about glue… but be careful when sticking it so that there are no creases.
- To make a foldable board, put two smaller boards near each other and tape them together with clear tape (such as mailing tape). Leave a little bit of space between them where there is just tape (this is where it will fold). Once you mount the actual game board on top, you might not even notice the tape.

For making custom dice:

- Print on adhesive labels, cut them to size, and stick them over the faces of existing dice.

- Leave a little room at the end of each die face (in other words, cut the adhesive labels a little smaller than the actual dimensions of the die). This will avoid having little bits of paper sticking out around the edges.
- For standard dice (white background, black recessed dots), the black dots may show through even after you put a label over them. If this is the case, put an extra layer or two of blank adhesive label over the face first, before applying the final label on the outside. This will reduce or eliminate the visual effect.
- If you really want to get fancy, there are some game manufacturers that sell blank dice with recessed sides. These are ideal, because you can put adhesive stickers on them and not worry about the stickers rubbing off through repeated use. These can be hard to find, though.

Coloring your pieces:

- If you have a lot of blank wooden pieces (such as cubes or pawns) that you want to differentiate by color, the easiest way is to fill a bowl with water and a little bit of food dye, then soak the pieces and let them dry. I suggest doing this in two stages. First, do a test on a single piece per color, to see what they'll look like (some colors may not show up as well as others, or you may find you need to soak them for more or less time; better to find this out *before* you ruin your entire stock of pieces). Then, once you have something that works, repeat for all of the rest of the pieces at once.
- You can also paint wooden pieces by hand, but it is far more time-consuming and tedious and the paint is a bit more expensive.
- If you have plastic pieces, the easiest thing to do is check your supplier to see if they naturally come in different colors already. Doing this by hand is not easy. If you find you must differentiate plastic pieces by hand, consider using small adhesive bands or dots instead of trying to color the plastic itself.
- If you have metal pieces and need to color them, paint is probably your best option. The same hobby stores that sell metal miniatures will probably also sell paints and other materials for you to paint them. Note that this can get very expensive and time-consuming, but if you're using metal parts you were probably not concerned about cost in the first place.

For printing your rules:

- The rules *are* the game, so don't neglect these! Print on good-quality paper at the very least.
- Consider if it is appropriate to "theme" your rules according to your game. For example, a game about trains might have rules laid out like a train schedule. A game about running a restaurant might have rules that are inserted into a plastic-shielded restaurant menu. A game about collecting classical artwork might have some art pieces displayed in the pages of the rules.
- Make sure the font of the rules is readable. Don't get too cute with custom fonts – doing this for headings is fine, but for the main text of the rules, remember that your players actually need to *read* them.

- And for the love of all that is holy, double- and triple-check your rules for spelling, grammar, and clarity of writing. You did this before, but do it again – this time it counts.
- Also check your written rules to make sure they are *correct*. You've likely made a lot of changes to the mechanics of the game over the course of the project, and the last thing you'd want is to accidentally print out an old copy of the rules before you made several key changes!

Everything is a Design Decision

If you just throw together a bunch of random components, that is a decision you made. If you hand-carved each component out of the branches of a tree in your backyard, that is also a deliberate decision. In short, everything you do for your game is a decision that you, as the designer, are responsible for.

So, make your decisions… but make them with both eyes open. Consider each component and how it fits into the overall visual and tactile aesthetic of the play experience.

Now, get out there and build the final version of your game.

Homeplay

I generally give my students in my classroom courses a single weekend to build everything. They do it, and they usually come to class on Monday very tired, reporting that they stayed up until 3am or later the night before just finishing everything.

What they don't say (although they *do* grudgingly admit to it when pressed) is that the reason they stayed up so late was that they procrastinated. Looking at what has to be done to build your game, it probably seems like a pretty small task. Be warned that it will probably take a lot longer than you think to craft the final version.

But you can still finish it in a weekend.

You have but one goal this weekend: **finish your Design Project**. Create a final version, using the best-quality components you can find.

On Monday, if you have the means, **post your game to the course wiki**. I advise use of the wiki and not the forums, because the forums do not support images or file attachments. Here is what you should post:

- A brief introduction: who you are, title of your game, what your game is about (briefly).
- A short creator statement: why did you choose to make *this* game, what was your inspiration, where did you get your ideas?
- A file with the final rule set, including a list of all required components.
- If some of the components are custom-made (such as custom cards, dice or a board), include the computer files you used to print them (if any), and include printing instructions for any who wish to print their own copy of your game.

- If you have the means, take a picture of your final game components (perhaps set up as a game in progress, or perhaps just all lined up on a table together) and post the photo.

Remember, you’re almost to the end! One final push of effort and you can stand back and examine something glorious that you made yourself.

Level 19: Game Criticism and Analysis

By ai864

If you have made it this far and kept up with the course in real time… then you have a strong will, and a lot of free time.

Take a step back and look at what you have done. And realize that this is not the end of your journey, but the beginning.

Today, we switch gears for a bit and turn our attention to game criticism. Why do we care the least about game reviews or critiques or game journalism or any other form of game writing? Because without the ability to analyze a game, we might make games and then be unable to discuss them amongst ourselves, and thus be unable to improve on them.

A game designer does not have to be an expert critic (nor vice versa), but an understanding of how to critically analyze games is a useful skill to have. With this ability, a designer can learn more by playing other people's games, figuring out what works and what doesn't (and why), and applying those lessons to their own designs. It's far cheaper to learn from *other* people's mistakes than your own.

Readings / Viewings

Read the following:

- *Game Criticism, Why We Need It, and Why Reviews Aren't It*, by Greg Costikyan. I could write at length about the difference between a "game review" and a "game critique" but Greg has already done so, and far more succinctly and clearly than I could.

Critical Analysis

I have mentioned before that an important game design skill to have is the ability to critically analyze *other* people's games. I think about half of the reason why I am as far along as I am in my career, personally, is that I have the ability to play a game and offer direct constructive feedback that is useful to another designer. My usefulness to designers makes me friends in a lot of places, and I want you to have the same opportunity.

Consider your own Design Project to be, if not finished, then at least "handed in" (if this were a normal class). Put it aside, and let us examine some other projects. It's good practice, and it may even make you some friends that extend beyond this course.

This time, we will follow a process that is a bit different than before. We are going to take off our game designer hats for a moment, and put on our *game critic* hats.

Review and Critique

We don't hear the words "game critic" very often. Especially with video games, more often we see "game reviewer." Review. Critique. Reviewer. Critic. Are these synonyms? What is the difference?

As Costikyan points out, the difference is in the purpose of the writing. In short, a *reviewer* is aiding the consumer in making a buy-or-don't-buy decision. A *critic* is writing about a game and why it is or is not important, or valuable, or what have you. Since we often hear the term "game journalism" as well, I would add that a *journalist* is writing about news in the field. Thus, a reviewer writing about *Settlers of Catan* might say that it is an excellent game that you should buy right now; a critic might write of its historical importance in bringing Eurogames to the US, or an analysis of its mechanics and how the game deals with minimizing player downtime, balancing luck and skill to make it accessible to all ages, and so on; and a journalist might write about a new game in the series getting released, or a new publisher getting the rights to the game, or the designer getting married to a famous celebrity, if any of those things ever happened.

For game designers, it is critique that is the most important, because critique can directly inform our present and future designs.

Not all critique falls into this category. A critique of *Chess* as a patriarchal game that pushes an anti-Feminist agenda would not be particularly useful if you're just trying to design a compelling strategy game, *sans* social commentary. But another critique of the same game might speak of the mechanics and dynamics, what is effective and what is not, and how it all fits together to make the kind of game that has persisted for centuries. And *that* would be quite useful, wouldn't you say?

Critical Analysis of a Game: the Process

First, go to the course wiki and select any one game (other than yours). Feel free to browse the projects that are posted, looking at the brief descriptions and photos at the beginning, and pick one that looks like it might be interesting. **Do this even if you did not complete your own project yet.**

Next, download the components. There should be enough there for you to build the game. You can make a quick-and-dirty paper prototype that contains all of the information, or you can print it out on simple paper. You can even follow the included instructions to assemble your own high-quality version, although this may take more time and money than you're willing to spend right now.

Next, read the rules and then play the game. You can try playing on your own if the mechanics allow it, or play with one or more friends if you get them together for one final playtest session. Play as a *designer*, paying attention to the mechanics, dynamics and aesthetics (in the MDA sense).

Lastly, reflect on your play experience. What were the game's apparent design goals? Did it succeed at those goals? Why or why not? What were the mechanics? What was the play experience? What is the relationship between the two? Did you find any strategies that were exploitable, or did the game seem well-balanced? What kinds of interesting decisions (and uninteresting ones) were you making throughout the game? What do you feel was the core of the game?

Write this up. Include the following in your analysis:

- Name of the game and its designer (partly to get in the habit of giving credit where due, partly so that if you accidentally post in the wrong place it can be rerouted easily).
- Description of the core mechanics of the game. You do not have to reproduce the rules, but you should describe the basic play of the game and the main decisions players are making, as if the reader had not ever played the game before.
- Describe the MDA dynamics and aesthetics, and show how they emerge from the mechanics (or guess, if you aren't sure).
- State the game's design goals. What was the designer trying to do? Then, say whether you feel the game met those goals, and why or why not.
- If there was anything else of note in the game (such as a particular issue with game balance or a unique use of game components), say that.
- Lastly, if you were the designer, what would you change about the game (if anything)? Make specific recommendations. For example, don't just say "I would make the game more interactive between players" or "I would fix the problem that I identified earlier" – say *how* you would fix things. What rules would change, and what would they change to? Would you change any game objects or values?

Remember, your audience is other game designers. Write your analysis so that other designers can learn from the mistakes and successes of the game you chose. Your goal is to educate and inform, and to discover new lessons about what makes games work or not work. Your goal is not to give a review score.

Homeplay

Post your critical analysis as a comment on the wiki. Go to the page of the game you chose, and leave a comment. I highly recommend writing out your analysis in something else like a word processor first, and then copying and pasting when you're done. Do this on or before the end of the course (Sunday, September 6). You will not have any other homeplay this Thursday, so take your time on this and be thorough.

You only need to do this for one game. If you *want* to provide feedback for more, feel free, but complete your original analysis first to make sure you'll have enough time to do this for others.

Level 20: Course Summary and Next Steps

By ai864

Here we are, at the end of the course. I appreciate the time that you have put in, if you have read this far and stayed with me from beginning to end. I hope that you got out of this course something of more value than what you put in.

We've covered a lot of topics in this course. We started by building the beginnings of a critical vocabulary to allow us to talk about games and game design. We looked at the process of designing games, particularly the advantages of rapid prototyping and iteration. We broke down the concept of "game" into its formal elements, and learned to analyze each element individually. We talked about common concepts that come up in game design: the MDA framework, feedback loops, emergence and intentionality, flow theory, kinds of decisions, kinds of fun and player types. We looked at narrative and the various roles it can play in a nonlinear game experience. We looked at games not only as entertainment, but also as an art form and an educational medium.

And, of course, we made games. Lots of games. Some little ones, and one longer project (which I'm sure you all feel now was *way* too short, even though you worked on this for a full month). We dove right in, applying the theory in order to make a game. Along the way, we discussed techniques for playtesting (in all its incarnations), balancing, and designing the UI for the game.

At this point, you might be wondering… what next? If this course is over, what are the next steps on this journey towards becoming a better game designer? And when does it end?

Lifelong Journeys

Since none of us are perfect or will ever be, there is *always* a way for us to improve. If game design is your passion and you want to design better games, you'll continue to improve over time, and this is a process that continues for as long as you make games.

No, the journey of a game designer does not end. But it does get more interesting, because you can put higher-level concepts together much easier. The kind of thing that used to take you a month eventually takes you a week, and the rest of that time can be spent doing even more for your games.

You might wonder, then, what is the next step on this journey? I have created a page on the course wiki with my thoughts, so visit over there (and leave your own thoughts) if you are done with this course and you want to get to the next level.

Some Other Questions You Might Have

There are some questions I've been asked a bit during this course, about what happens when it all ends. Here are my answers.

I want to pass some of this material along to other students/friends/colleagues. Can I get the rights to use it? Can I link to it from my own blog/course/whatever?

I've received a number of emails asking about permissions, and I have just upgraded the blog with a fresh new Creative Commons license that should make it clear. Bottom line: feel free to use any or all of the content that I've posted here. I created this class to share information, after all. However, please do credit me as the original source if you use my content. My name is Ian Schreiber, and the title of this course/blog is Game Design Concepts.

I came to this class late / I fell behind, so now I'm on my own. What happens to this course now that it's technically "over"?

I plan to leave this blog right where it is for posterity. Anyone who finds it later can feel free to drop by, going through the course on their own time and at their own pace. They won't be able to participate with other students as the course is happening, of course, but the material is still all here.

The course wiki will remain exactly as it is, readable to the public. By popular demand, the course forums will remain as they are, allowing those of you who signed up to maintain a community.

Are you teaching this class again?

At present, I have no plans to do an *exact* repeat of this summer. However, all of the information is here, so anyone wishing to go through Game Design Concepts can do so at their own pace at any time.

Are you going to teach any other classes like this?

YES. Next summer, I will do something similar to this one, but with new material.

The topic for next summer is **game balance**. I want to go into greater detail on ways to take an existing design and get it feeling right: identifying feedback loops and other relationships in a game, cost curves, metrics, randomness, payoff matrices, and similar topics. It's an area of fascination for me, and I would love the opportunity to share what I have learned among a community of like-minded game designers and hobbyists. Since *Game Design Concepts* is experimental in nature, I'd like to take the opportunity to go into an experimental topic, the kind of thing that I would probably not be allowed to teach at most schools because it is too specialized. No textbook for it exists (yet). I'd like to write one, some day, but I've learned to not write a textbook unless you've taught a class in it *first*. Next year I'll teach that class, and I am already looking forward to it.

Details for signing up will be posted here on this blog in early 2010.

And Now, Some Shameless Plugs…

Game Design Concepts reached a lot of people. As you see in the sidebar, there were over 1400 people who sent an email to formally sign up for the course ahead of time, covering nearly every state in the US and an additional 47 countries. Here are some other stats for the interested:

- 400+ game industry professionals (including 160+ professional game designers)
- 200+ teachers and educators (including 60+ who teach game design)
- 400+ students, ranging from middle school to graduate school
- 350+ people who signed up in a group (that is, not alone), forming 120+ groups
- All of the above is just for people who signed up in advance. There are, of course, many more who visited the blog but did not formally sign up ahead of time.
- Starting out, the blog had 6,000+ unique visitors for both if the first two lessons, with 13,271 hits in the first week total.
- More recently, we get about 1,000 hits when there's a new blog post, and about 400 on days in between posts.

So, if you're reading this, you are in good company.

As a result, I ask the following:

- Do you work for a school or company that would be interested in sponsoring next year's course? I can offer your logo and link on the main blog page, and mentions at the end of each blog post with your message. Your message could be seen by thousands of repeat visitors. Contact me by email to receive sponsorship info.
- If you are a professional educator (teacher, professor, etc.) at an institution that offers online classes: would you be interested in an adaptation of *Game Design Concepts* for your institution? I can do that for you. Are you looking for game design professors to teach an existing online class? I can do that, too. Contact me by email, and let's talk.
- Do you work at a game company that may eventually look for a freelance game designer on a short-term contractual basis? Ask me for my full contact details by email. Résumé and references available on request.

Also, if you liked the textbook, my co-author and I would appreciate if you'd leave a review on Amazon. For as many people as have bought the book, there are currently very few reviews, and it would be nice to see that change.

Course Evaluation

At the end of most college courses, students are given an evaluation to fill out. It asks all kinds of questions about the strong and weak points of the class and the professor. The answers are collected, compiled, and given back to the professor. I continue this tradition here.

I want to know what I did right, and more importantly, what I did wrong.

I have set up a survey here: http://survey.constantcontact.com/survey/a07e2kh03mmfz3laohv/start

If you are reading this at all, I would like to hear from you. This is true whether you originally signed up for the course, or came in late; whether you kept up with the coursework, or not; whether you enjoyed yourself or whether you thought this was a complete waste of time.

Go there and fill out the survey. I’m asking you to do this as a personal favor to me, in exchange for the time I have given you in putting together the content for this course. Your responses help me to make things even better next year. Thank you.

Final Words

I would like to thank you for your interest, your participation, and your time. I wish you well, in games and in life. Keep playing, keep designing, and keep learning.

- Ian Schreiber