game design concepts:

an experiment in game design and teaching

by Ian Schreiber

Level 0:

What Is Game Design Concepts?

Originally posted March 31, 2009 and April 21, 2009

My name is Ian Schreiber. I've been working in the video game industry since the turn of the millenium, first as a programmer and then as a game designer. I've taught college classes in game design since Fall 2006. For any other information, you can Google me.

What is this book?

This book is an experiment in game design and pedagogy. During Summer 2009, a series of lectures, course notes, readings, and challenges were posted to this blog on the subject of game design.

This book is a course in game design (specifically, non-digital systems design).

» Tuition: *none*. This class is open to all.

» Prerequisites: *none*. It is my intention to make this course accessible to all levels of experience, while providing useful additional resources for those who are advanced.

» Schedule: Take it at your own pace; when originally posted, there were two articles a week posted over the course of 9 weeks.

» Audience: anyone with an interest in game design. This includes students who are interested in game design; faculty who teach courses in game design and would like to compare course material; game developers with an interest in design or a desire to see an example of what students are being taught these days; or relatives of game designers who are curious about what these people do all day.

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.

Why are you doing this?

I have many motivations for starting this project, some selfish and some altrusitic. Best to be up front about it:

» Game design is my passion, and I love to share it with anyone and everyone.

» I have taught some classes in a traditional classroom and others online, and I want to experiment with alternate methods of teaching. By exposing my course content and viewing

» the comments and discussions, I can improve the course when I teach it for money. It is a career move. If this course is successful, it gives me greater exposure in my field and promotes my name as a brand.

Is this really, totally, 100% free?

The book is free. There are some minor costs:

» There is a required textbook. It retails for under $25 US.

» Part of the course will involve the creation of a fully realized non-digital project, so you may need to purchase materials. These usually range from $25 to $50, depending on the game. It's still cheaper than college tuition.

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 Comcs: 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.

Syllabus

The following syllabus is a reprinting of the syllabus as it appeared on the original site during the course. It is reprinted here to give you an idea of the pacing of the original course, in addition to acting as a summary of the upcoming course.

Original 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
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

Where can I get more information?

You can send an email to gamedesignconcepts@yahoo.com asking for more information.

Level 1:

Overview / What is a Game?

Originally posted June 29, 2009

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 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 33 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!

» **z**. 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.

Level 1 - 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 moving on to the next level, 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* (available at `http://www.costik.com/nowords.html`), 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 at `http://www.costik.com/nowords2002.pdf` [PDF] if you are interested, but I prefer the original for its historical importance.)

» *Understanding Games 1, Understanding Games 2, Understanding Games 3, and 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. They are all available at `http://www.kongregate.com/`.

Level 2:

Game Design / Iteration and Rapid Prototyping

Originally posted July 2, 2009

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***.

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 Level 1, 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.

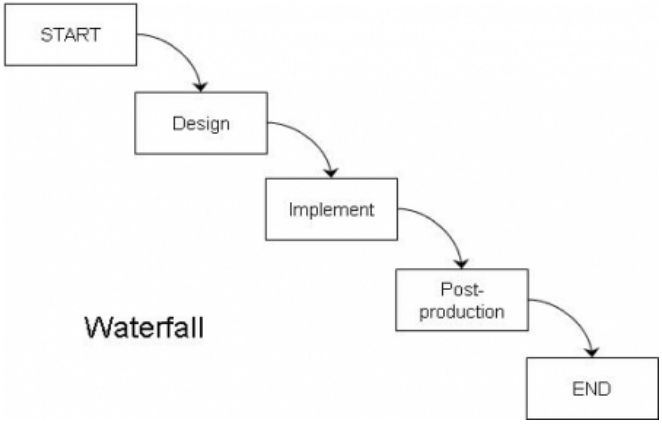

Figure 1 - The Waterfall Method

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.

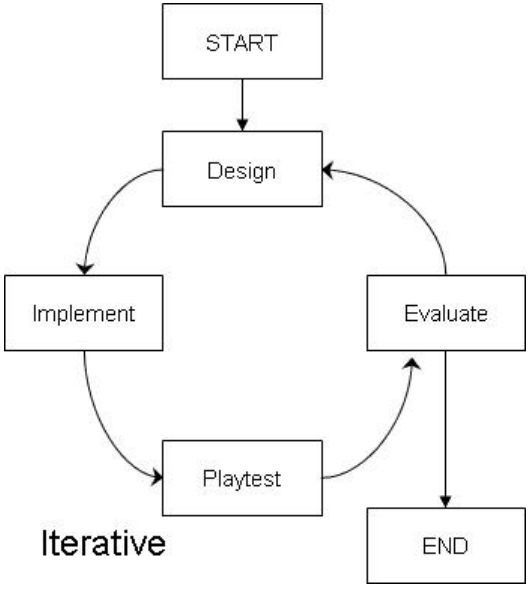

Figure 2 - The Iterative Approach

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.

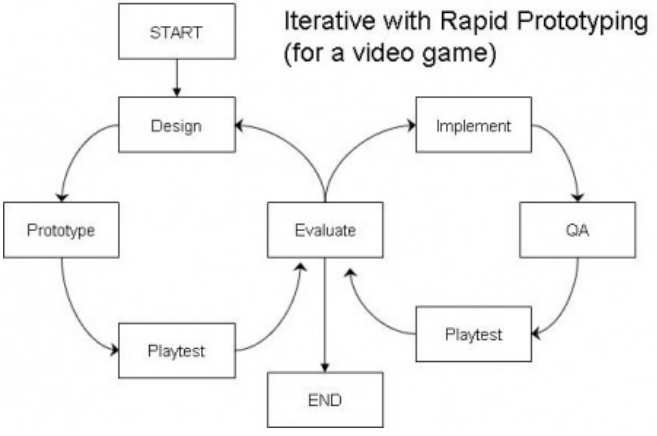

Figure 3 - The Iterative Approach with Rapid Prototyping

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…

1Some 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 in Level 1, 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.

Level 2 - Homeplay

Before reading Level 3, read the following. I will be referencing these in Level 3'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* (available at `http://www.gamasutra.com/features/19990716/design_tools_01.htm`), 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

Originally posted July 7, 2009

Today marks the last day that we continue in building a critical vocabulary from which to discuss games; in Level 4 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?

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 in Level 1. 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, constitutative, 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.

Level 3 - 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.

On the forums for this course (`http://gamedesignconcepts.aceboard.com/`), you should find one area for each difficulty level. Post your game rules in the appropriate level. 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. Make sure that everyone can get feedback and post on those who haven't gotten any yet.

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, viewable at `http://gamedesignconcepts.pbworks.com/Legal-Issues-for-Game-Developers`. 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

Originally posted July 9, 2009

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.

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 basics. 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 (Mi-

chael'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 (`http://www.craftparts.com/`) 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*** (`http://www.costik.com/vidiot.html`) 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***.

Level 4 - Homeplay

Do Challenge 4-1 in the text. It cannot be a game that already has a commercially-available board game adaptation. (Check BoardGameGeek at `http://www.boardgamegeek.com/` 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. 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.

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 at `http://www.raphkoster.com/2005/11/01/how-to-prototype-a-game-in-under-7-days/`.

» An article on paper prototyping, written for an audience of video game developers, available at `http://www.gamasutra.com/features/20060508/henderson_01.shtml`.

Level 5:

Mechanics and Dynamics

Originally posted July 13, 2009

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.

Battleship Changes

I compiled 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 when we've made it to Level 7.

Readings

For this Level, 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* (at `http://www.cs.northwestern.edu/~hunicke/MDA.pdf`) 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 (Level 8).

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 (at `http://clicknothing.typepad.com/Design/hockingc_GDC06_Intentionality.zip`) 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 (at `http://gamedesignconcepts.pbworks.com/L5-Homeplay`). On that page, write the following:

» What was your rules change?

» How did you expect the dynamics of the game to change?

» 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!