

# Introduction to Design Issues in Video Game Development

Bernard Yee ■ [bhy@bernieyee.com](mailto:bhy@bernieyee.com)

Fall 2010

# Credits

- Doug Church
- Mark Leblanc
- Richard Garfield
- Many, many others...

# Outline of Talk

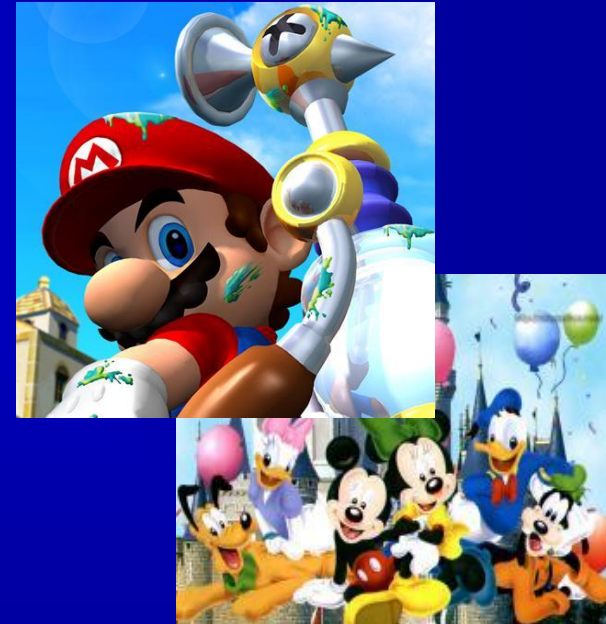
- Introduction to the Game Development Process
  - Unique problem set in game development
- Computer Game Examples
- Design as Feature Set
  - case studies
- Future directions

# DISCLAIMERS

- The topic is huge and this is, **at best**, an overview
- No one knows how to definitively teach game design
- Many different approaches and viewpoints
- Games are meant to be played, not watched
- Web has many resources, including demos, reviews, screenshots, analyses
- There are a huge range of games
- If you are interested, just go play...

# If you really want to read...

- Escapist
- Gamasutra
- Joystiq/Kotaku
- Develop
- Terranova
- Blue's News
- Ludology.org



# What are “Computer Games”?

- Game (from dictionary.com)
  - activity providing entertainment or amusement
  - period of competition or challenge
- Computer
  - machine for performing calculations automatically
  - Algorithmic Decision processing

# The Development Process

- Concept
- Design document – game systems architecture
- Technical Design Review – schedule, technical plan of attack
- Various development stages (proof of concept/prototype, iteration, alpha, beta, release candidates) and teams (designers, programmers and artists)
- Testing and bug fixing
- Launch

# Probably familiar to you, eh?

Some key differences make games uniquely  
painf ..., err, *challenging*.

I will focus on game design as the unique  
aspect of this particular field of software  
development

# So what makes games different?

- Software engineering is a major part of game development
- Feature set ~ fun – but how to spec “fun”?
- Goal is to *entertain*, a psychological process not easily defined
- Constantly changing platforms and technology

# How important is technology?

## Top 10 PC games May 2009

1. *The Sims 2 Double Deluxe* - Electronic Arts
2. *World Of Warcraft: Wrath of the Lich King* - Blizzard
3. *Spore* - Electronic Arts
4. *World Of Warcraft Battle Chest* - Blizzard
5. *World Of Warcraft* - Blizzard
6. *Empire: Total War* - Creative Assembly/Sega
7. *Reel Deal Slots Adventure* - Phantom EFX
8. *Civilization IV: The Complete Edition*
9. *The Sims 2 Apartment Life* - Electronic Arts
10. *Left 4 Dead* – Valve

- Top 10 PC games June 2009
- 1) *The Sims 3 / EA*
- 2) *The Sims 3 Collector's Edition / EA*
- 3) *The Sims 2 Double Deluxe / EA*
- 4) *World of Warcraft: Wrath of the Lich King / Activision*
- 5) *World of Warcraft: Battle Chest / Activision Blizzard*
- 6) *Spore Galactic Adventures / EA*
- 7) *World of Warcraft / Activision Blizzard*
- 8) *Spore / EA*
- 9) *Empire: Total War / Sega*
- 10) *Starcraft: Battle Chest / Activision Blizzard*

# How important is technology?

- Top 10 console games April 2009
  1. Wii Fit w/Balance Board (Nintendo, Wii)
  2. Pokemon Platinum Version (Nintendo, NDS)
  3. Mario Kart w/Wheel (Nintendo, Wii)
  4. Wii Play w/Remote (Nintendo, Wii)
  5. The Godfather II (EA Redwood Shores, 360)
  6. Resident Evil 5 (Capcom, 360)
  7. New Super Mario Bros. (Nintendo, NDS)
  8. Mario Kart DS (Nintendo, NDS)
  9. Guitar Hero Aerosmith (Neversoft, 360)
  10. The Godfather II (EA Redwood Shores, PS3)

# How important is technology?

- Top 10 console games July 2009
  1. Wii Sports Resort (Nintendo Wii)
  2. NCAA Football 10 (Xbox 360)
  3. NCAA Football 10 (Playstation 3)
  4. Wii Fit (Nintendo Wii)
  5. Mario Kart Wii (Nintendo Wii)
  6. Mario Kart DS (Nintendo DS)
  7. Pokemon Platinum (Nintendo DS)
  8. Fight Night Round 4 (Xbox 360)
  9. New Super Mario Bros. (Nintendo DS)
  10. EA Sports Active (Nintendo Wii)

# Primary Goal of Game Development

- Focus on **player experience** is (or should be) primary goal
- Technology and design further that end, rather than being an end
- Few (if any!) of the top 10 games are using “cutting edge” technology on PC
- Less true on console

# Many valid approaches to design problems

There is no “right design”

There are designs which are appropriate for certain situations, and appeal to certain types of players.

The crucial task of a designer is to consider, understand, and facilitate the player’s experience.

# What is *Meaningful* Interactivity?

- Interactivity is so overused and ill-defined
- Both a technical and psychological challenge
- Game development inseparably combines technology (engineering) and creative (design)
- What is interactivity in a games context?

# A game designer...

...uses rules the way a writer uses words.

That's why programming expertise is a huge bonus for game designers.

Why are so many game designers former writers or nontechnical people, then?

# Interactivity and Game Design

## 1. Agency

The player is able to affect the environment/world state in a meaningful manner

## 2. Choice

The player is given meaningful choices that reflect the player's *intention*

## 3. Intention

The player can form an *intent* and then act.

## 4. Consequence

The player's choice results in a meaningful consequence

## 5. Feedback

Consequence is fed back to player, giving player more/new *choice*

# What do players get from games

- They learn methods to succeed at the game
  - How to score the most points
  - How to play the longest
- They have an experience in the environment
  - Feel like part of some story or event
  - Get to try out given roles or characters
- Partake in things they otherwise could not

So How Do We Do This?

# The Tasks of Game Design

- Create an experience for the **player**
- The player is usually in some **environment**
- This environment has **rules** of interaction
- The rules are managed by the **computer**
- The computer, as mentioned, is **algorithmic**
- This is a feedback loop/system function

# So...

At the core, designers build up a collection of algorithmic systems that work together to create interactive environments for the player to experience.

# Progress Quest

- Game?
- Or not a game?

# How do we design games?

- Well, let us investigate the medium of games
  - Modern PC and Console environments
  - Continually moving tech base, evolving platform
  - Rule based systems which must work in concert
- Build the system, then the player starts using it
  - No chance to intervene or direct
  - Players can try out whatever they want
- Flexibility is crucial if anything is to work

# Game Examples

- Grand Theft Auto III
- Some scripted game play (win conditions, loose goals)
  - Overlapping simulations allow player to experiment with different approaches
  - Flexibility and illusion of flexibility

# The brittleness of software

- Computers have really bad dynamic range
  - SimNET was fine for tanks, not much good for ants
  - Any computer system has huge numbers of constraints
- Combining many constrained systems produces a very complex and even more constrained whole
- Players will try anything possible in the system
- Hard to test, impossible to get “right”

# Game Design = Tradeoffs

Choosing, enforcing, and hiding constraints is, to some extent, the art of game design.

Building systems to support gameplay interactions is obviously vital.

But choosing the boundaries for them, and coherently setting, supporting, and testing these limits, is the part where it all comes together

# Where do you start?

- Many people want to start with genre
- Need to focus on player experience
- A fishing game, say, could be
  - Real time arcade twitch gameplay
  - Stats and simulation, tournaments, inventory
  - Simple rules, random die rolls



# Design choices to make

- Pacing? Event frequency? Obstacles?
- Skill based or stat based?
- Real time or Turn based?
- How much can the player really impact?
- Dramatic Arc or Continuous Tension
- Spreadsheet gaming? Or Random?

# Who is in control?

- Is it a rigid environment, where the player has to solve the exact things you have set up
- Flexible simulation, where players interact with a wide set of rules, trying things out
- Who sets the “victory condition”? Is there one? Are there closure points?
- What provides the value? Getting to a goal? Just playing? Some of both?

# Game Example

- Warcraft 3, real time strategy game
- Appears to be a flexible simulation, but RTS can also be a rigid environment requiring choreographed solutions
- Many seemingly elaborate game designs boil down to simplistic systems (ProgressQuest)

# Scripting vs. Simulation

- Simulation provides deep rule based reactive systems, but often is very limited, and there are many things it cannot create
- Scripting can do very precise and complex behaviors, but they are rarely really about the player, instead being about the designer
- Choosing when each is appropriate is vital

# Get The Player On Stage

You are designing for the player, not yourself

They usually aren't playing to watch you, the designer, show off... they are playing so that they themselves can show off.

Get the player on stage, get yourself off of it.

# More about “Fun”

- Fun is way overused, and not very useful
- Getting beyond “fun”... why is it involving?
  - Fast paced action?
  - Felt like you really had to think it through?
  - The style and setting appealed to you?
  - Got to feel like you were doing cool things?
  - Sense of accomplishment?
- Why will the player enjoy and care?

# Case Study

Car Games

# Car games “goals”

- Play through a season, growing in skills
- Single races, go for best lap time
- Cars have to be earned and improved through successes on the track
- You simply pick a car and try it out
- Move “car tokens” around a board
- Drive car using real-time physics models



# Full on Simulation Model

- Cars has components, real physics model
- Optimal line (perhaps parameterized by speed) computed for the AI
  - Avoid collisions, manage speed and line
  - “personality”: hangs back, reckless...
- Over course of season, need to qualify

# Design implications

- Player had better be a pretty good real driver, or they are going to have no fun
- Any ways the simulator differs from real driving will train the “wrong” skills
- Need to balance car models and capabilities, which requires tuning physics



# An arcade/game mechanic model

- Each turn, cars move along track squares
- After each race, can buy car upgrades
- Power-ups found on track squares
  - Speed boosts, attack or defense objects
- More is always good for the player
- Basically, a stats game with a race theme

# Design implications

- Car rules automatically grow “skill”
- While racing sense is useful, mastery of the power-ups and rules more important
- Need to balance power-ups and rules
- Must explain non-real-world rules to players, so they can make decisions
- Low barrier to entry, no dexterity needs

# Both games are about the experience of racing

- They have very different feels, but both are recognizably “racing” games
- One provides the physical side of driving
- The other provides the trappings and structure of a race, without details
- And both have similar problems... a race that isn't close isn't much fun

# How do we make races fun?

- As designers, we want to recreate racing, not just driving around on a track
- Competition is a crucial part of that
- Need to increase likelihood of a close race
- So we could count on players getting good
- or, better yet, we could **cheat**

# How do we cheat well?

- We have to slow the front, speed the back
- Easiest way is just with speed
  - Cars in front slow down, in back, speed up
- This can be very obvious to players
- And, worse, risks removing player agency

# Dynamic Difficulty Adjustment

- Game monitors player behavior
- As player struggles, game changes to try and help the player through it
- If player does well, game becomes harder

# Notes

- If you make it too real, you make it too hard for players to actually succeed and enjoy it
- If you cheat to improve tension, you risk taking control from the players
- Both risk pushing players away, as they lose any feeling of agency

# Risks of DDA approaches

- It seems obvious adaptive models are better for tuning an experience
- However, if a player realizes they are involved, they can exploit them
- Slowing down until the end of the race, for instance

# What the manual says isn't the point

- If the way to “win” is to fight, you can say “hide” all you want, but they will fight
- If there are bugs in the rules, they will find and exploit them, even if they enjoy it less
- Players learn to “win the game”. Thus gaming to learn is a tricky design problem

# Game Example

- Ultima Online
- Massively multiplayer online game
- Traditional “fantasy role-playing” genre
- Heavily evolved since 1997 to expand gameplay AND combat exploiting “gamed” algorithms

# Emergent Behavior

- The castle
- Crates
- Teleportation/recall
- Emergent behaviors at different scale
- Genre (and business model) is new

# Players use the rules

Players learn to excel at the provided rule-system,  
not the ideas in your head

- They don't learn the manual
- They don't play what you thought was cool
- They don't only do "reasonable" things
- They poke and prod the systems, and exploit any weaknesses they can find
- 200,000 players are more thorough than your development & testing team

# Case Study: AI Guards

- In many action/shooting games, there are NPC guards who the player must defeat
- In many games, these AI's are in fact not working to defeat the player at all
- If AI's worked hard to defeat player, the player would always lose
- AI's instead work to challenge the player

# For Instance

- If the AI catches a player unaware, AI says something before attacking
- AI's are extremely forgetful/forgiving
- AI's first shot often a deliberate miss
- AI's stand down from alerts very easily
- i.e. AI's are not trying to be realistic
- AI's as babysitters, not ruthless opponents

# Note: Reality in Gaming

- Many say “obviously right” design is skill based
- Sure, beginners may have problems, but
  - Go ahead and do a beginner league
  - Have training and learning modes
  - Maybe a role-playing aspect, where you grow your character into better leagues
- But really, people don’t necessarily want to make a career of your game

# Realism as red herring

Just because it is “realistic” doesn’t mean it will be interesting.

Players are paying us to entertain.

Want a sense of accomplishment and learning, not necessarily the real work that entails

# Was there a point to all this

- Game creation is complicated
- Many decisions are made during design
- Each creates and limits player possibilities
- Smoke and mirrors to fill out experience
- If you force things, interest is rapidly lost
- How do you create and tune an experience, while keeping the player at the center of it?

# What do players learn

- How to probe the system to learn behaviors
- To find the optimal approach to rule-set
- They learn about the fiction/setting
- Discover what happens when failures occur
- How “seeming” good-ideas break
- Ideally, that they can impact the world

# Gaming provides ways to...

- allow players to try different approaches
- show visceral consequences of choices
- create an rich and dynamic environment
- involve in and motivate players to complex situations with much to understand

# But it also provides...

- players a lot of flexibility to break things
- constrained and artificial rulesets
- reliance on smoke and mirrors

# Challenges in game development

- Balancing/tuning increasingly complex systems
  - Simulation-based game play (Grand Theft Auto)
  - Heavily numeric/spreadsheet games (D&D-esque, stat driven systems like EverQuest)
  - Massively multiplayer player base (hundreds of thousands of “testers/exploiters”)

# Challenges, more

- Creating game content
  - Hardware – the more powerful the graphics hardware, the more demanding the art requirements
  - Online – the bigger the user base, the more game content needed
    - Laying track before the train

end