



# The Game Development Process

## Game Design




WORCESTER POLYTECHNIC INSTITUTE



# Outline

- Gameplay
- Game Balance



WORCESTER POLYTECHNIC INSTITUTE

## Approaching Game Design

- Used to be thought that could not teach game design ... more of an art
  - But you can teach art! (AR1100 + ...)
- Even to art, there are technical disciplines such as in music, film, poetry
- So, consider computer game design as an art form
  - Game design practices can be taught

Chapter 2.2, *Introduction to Game Development*



## Game Theory

- Some designers approach Game Theory thinking it will help design games
  - Rather, it is a theory about games are *played*
- Game theory is ...
  - Branch of economics
  - Systems governed by rules
  - Mathematically analyzed to determine payoffs of various end points.
- Game theory assumes rational players
  - Abstract model players - not real people
    - Always try to maximize their potential utility
    - Solve problems using pure logic
    - Always fully aware of the state of the game

Chapter 2.2, *Introduction to Game Development*



## Gameplay

- Gameplay (given the definition of game theory)
  - Collective strategies to reach end points of game theory
  - Specific to game activities
  - "What the player does"
- Includes
  - *Utility* - A measure of desire associated with an outcome
  - *Payoffs* - The utility value for a given outcome
  - *Preference* - The bias of players towards utility
- Note, gameplay is not everything
  - Choice of car in *GTA* is not always about payoff, but about what is *fun*
  - Software doesn't have to have gameplay to be entertaining ... consider *SimCity*
- No one expects gameplay in movies or plays
  - Who says: "Hey, where is the gameplay in Hamlet?"
  - Rule 1: It should be fun (entertainment)
  - Rule 2: It should be interactive (make use of computer, else perhaps use film)
  - Rule 3: It can have gameplay (but that is a choice)

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Gameplay Example (1 of 2)

- Adventure game: *Knight and Priest*
- During combat
  - Knight in front with sword
  - Priest casts spells (assume all cost the same)
    - E-bolts (do damage equal to sword)
    - Band-aids (heal equal to sword)
- Fighting a single opponent with sword
- Which spell should Priest cast?
  - Ask: against opponent with 6 arms (e-bolts)
  - Ask: against many opponents with weak attacks (band-aids)
- Can always decide which is better
  - Not so interesting

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Gameplay Example (2 of 2)

- Now, suppose...
  - Band-aids still affect single target but e-bolts have an area affect
  - E-bolts do less damage, but armor doesn't make a difference
- Now, which spell should Priest cast?
  - Answer isn't as easy. Interesting choices. Good gameplay.

"A game is a series of interesting choices."  
- Sid Meier (*Pirates, Civilization...*)

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Implementing Gameplay (1 of 3)

- Choice
  - A question asked of the player
- Outcome
  - The end result of a given choice
- Possibility space
  - Represents the set of possible events
  - A "landscape" of choice and outcome

Chapter 2.2, *Introduction to Game Development*





## Implementing Gameplay (2 of 3)

- Choices must be non-trivial, with *upside* and *downside*
  - If only upside, AI should take care of it
  - If only downside, no-one will ever use it
- Note, this is only regarding *Game Theory*
  - Ex: Could have ray gun that plays music. "Cool", but soon "gimme the BFG"
  - Ex: Nintendo's *Smash Bro's* has "Taunt" ... ask: what for?
  - Ask: other examples from popular games?
- Gameplay value when upside and downside *and* payoff depends upon other factors
  - Ex: Rohan horsemen, but what if other player recruits pikemen?
  - Ex: Bazooka, but what if other player gets out of tank?

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Implementing Gameplay (3 of 3)

- Should be *series* of interesting choices
  - Ex: Use of health potion now may depend upon whether have net for capturing more fairies
  - Having net may depend upon whether needed space for more arrows for bow
  - Needing arrows may depend upon whether killed all flying zombie bats yet
- Hence, well designed game should require *strategy*
- Game must display *complexity*
  - But doesn't mean it must be complex!
    - Don't make too many rules. Less if more.
    - Real world example: termites place one piece of mud. Results in hive, with cooling vents, etc.
  - *Emergence* from interaction of rules
    - Ex: In *Populous*, Priests convert, but not if already in combat. By design? Maybe, but non-intuitive result.
    - Ask: examples from popular games?

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris





## The Dominant Strategy Problem

- Articles with "10 killer tactics" or "ultimate weapon"
  - Ask: what are these doing?
    - Taking advantage of flaws in the game design!
- Should never have a option not worth using
  - *Dominated* strategy
- Should never have an option that is so good, it is never worth doing anything else
  - *Dominant* strategy

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Near Dominance

- Worth looking for near dominance, too
  - *Near-dominated* - useful only very narrow circumstance
  - *Near-dominant* - used most of the time
- Ex: *stun gun* only useful against raptors, so only useful on raptor level (near dominated)
  - Do I want it used more often?
  - How much effort on this feature?
  - Should I put in lots of special effects?
- Ex: *flurry of blows* most useful attack (near dominant) by Monk in D&D
  - Should we spend extra time for effects?

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Avoid Trivial Choices (1 of 2)

- Horsemen → Archers → Pikemen
  - *Transitive*, not so interesting
- Horsemen → Archers → Pikemen → Horsemen (picture)
  - Ask: what game does this look like? (rock-paper-scissors)
  - *Intransitive*, more interesting
  - Ex: from LOTR Battle for Middle Earth
    - Horsemen fast, get to archers quickly with lances
    - Pikemen spears hurt horsemen bad
    - Pikemen slow, so archers wait on them from afar

(Will look at *game balance* in depth, next topic)

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Avoid Trivial Choices (2 of 2)

- A beats B, B beats C, C beats A (could hardwire)
  - But could also have how *much* better
    - 1) Single horseman can beat any number of archers:  
Horseman → Archers ( $\infty$ )
    - 2) Single horseman barely beat an archer:  
Horseman → Archers (1.1)
- Ask: Which is better?
  - Trick question! Both are bad
    - Case 1) equal number of each, all others lose
    - Case 2) doesn't matter which you choose (turns to RPS)
- Don't want to hardwire. Sometimes A way better than B, sometimes a bit better, sometimes worse
  - The answer should depend upon the game situation, weather, terrain, time ... also what opponent is doing

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Environment + Rules = Gameplay

- Battle of Hastings, 1066 A.D. <http://www.battle1066.com/>
- King Harold tired, mostly infantry
- Duke William more archers, cavalry
  - Archers beat slower infantry → game over?
- Not quite ... Harold on hill (arrows less effective) and defensive mode
  - Archers tire, run out of arrows → game over?
- Not quite ... William also smart, cavalry approach, but retreat. Infantry break ranks since they are frustrated, charge
  - Arrows now shred Infantry → Harold loses, game over
- Point: ways to change balance between different troop types. "A good commander isn't the one with the best army; he is the one who knows how to use it best"

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Ensuring Interesting Choices (1 of 2)

- Ex: *Elite* (early '80s, ask: who played?)
  - Accumulate wealth by trading
  - When 1000 credits, trade lazer for better lazer and have 400 left over for trading. No brainer. Always a win.
  - What if could buy lazer with 600? Then no credits left over. Decision is tougher.
- Point: keep difficult choices in hands of player
- Ask: other examples?

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris







## Ensuring Interesting Choices (2 of 2)

- Interesting choices require good judgment on the part of the player
  - Correct choice must vary with circumstances
- Aim as designer, ensure circumstances don't stagnate and have only one right way to win
- No method for finding "best" choices
  - That's where creativity comes in (art)
- Still, some tips ...

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Toolbox of Interesting Choices

- Strategic versus Tactical
- Supporting Investments
- Versatility
- Compensating Factors
- Impermanence
- Shadow Costs
- Synergies





## Strategic versus Tactical (1 of 3)

- Strategic choices affect course of game over medium or long term
  - *Tactical* choices apply right *now*
  - Ex: build archers or swordsmen (strategic)
  - Ex: send archers or swordsmen to defend against invading force (tactical)
- Strategic choices have effect on tactical choices later
  - Ex: if don't build archers, can't use tactically later

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Strategic versus Tactical (2 of 3)

- Ex: *StarCraft*
  - Strategic choice: 1) upgrade range of marines, 2) upgrade damage, or 3) research faster fire
  - Which to choose?
    - If armored foes, Protoss Zealot, more damage
    - If fast foes, Zerglings, maybe faster fire
  - Other factors: number of marines, terrain, on offense or defense

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris





## Strategic versus Tactical (3 of 3)

- Ex: *Warzone 2100* (ask: who played?)
  - Build factories to spawn war machines
  - If build in level, then spawn quickly but factory only used for that level
  - If build at base, spawn slowly (have to ship to front lines) but factory can be used in subsequent levels
- Lesson: Good gameplay should have different choices leading to different *kinds* of payoff
  - Reduces the risk of trivial choices
  - Increase scope for good judgment

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Supporting Investments

- Often game has primary goal (ex: beat enemy) but secondary goals (ex: build farms for resources)
- Some expenditures directly impact primary goal (ex: hire soldier), while others indirect (ex: build farm) called *supporting investments*
- Primary goals are "one-removed"
  - Ex: improve weapons, build extra barracks
- Supporting goals are "two-removed"
  - Ex: build smithy can then improve weapons
  - Ex: research construction lets you build smithy and build barracks (two and three removed)
    - Most interesting since strategic
- Payoff will depend upon what opponents do

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Versatility (1 of 2)

- For balance, a guideline is to ask what is best and worst about choices:
  - 1) This move does most damage, but slowest
  - 2) This move is fastest, but makes defenseless
  - 3) This move best defense, but little damage
- Most should be best in some way
- With versatility, a 4<sup>th</sup> choice:
  - 4) This neither best nor worst, but most versatile
- Versatile good for
  - beginners
  - flexibility (against unpredictable or expert opponent)

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Versatility (2 of 2)

- Ex: beam can mine asteroids and shoot enemies
  - Versatility makes it good choice
- Speed is common way for versatility
  - Don't make fast units best at something else
- If a versatile unit is also cheapest and most powerful → no interesting choice
  - (See "Compensating Factors", next)

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Compensating Factors

- Consider strategy game where all units impeded by some terrain
  - Ships can't go on land, tanks can't cross water, camel riders only in dessert
- Assume flying unit that can go anywhere (Ask: how to balance?)
  - 1) Make slow
  - 2) Make weak, easily destroyed
  - 3) Make low surveillance range (unrealistic)
  - 4) Make expensive
- Note, last choice common but uninteresting since doesn't change tactical use
- Choice should be clear to player. Don't make a gamble before they know.
  - Ex: pick troops (cold weather) then find in jungle

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Impermanence (1 of 2)

- Some permanent (ex: you get to treasure first), others not (ex: I got storage near mine, but you can grab it off me)
- Really, another kind of compensating factor
  - i.e. - impermanence can compensate for something being really good
  - used since such a common, and valuable technique
- Can be used for interesting choices
  - Ex: choice of medium armor for rest of game or invulnerable for 30 seconds?
- Advantage (or disadvantages) can be impermanent in number of ways:

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris





## Impermanence (2 of 2)

- Examples (mostly from *Magic the Gathering - Battlegrounds*)
  - Can be destroyed (enchantments, ex: *gratuitous violence* makes units tough, but can be destroyed)
  - Can be stolen or converted (ex: *threaten* steals or converts enemy for short time)
  - Can be applied to something you don't always have (ex: *goblin king* gives bonus to goblins, but must have goblins)
  - Certain number of uses (ex: three grenades, but grenade spamming)
  - Last for some time (wears off, ex: Mario *invulnerable star*)

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Shadow Costs (1 of 2)

- In a game, continually presented with costs and trade-offs. But not all direct.
  - Ex: soldiers for gold, but need armory first for weapons and barracks for soldiers
  - Called *shadow costs* for supporting investments
  - And shadow costs can vary, adding subtlety

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Shadow Costs (2 of 2)

- Ex: Age of Mythology has wood and food. Food is inexhaustible, wood is finite
  - Charioteer
    - Costs 60 wood, 40 food and 40 seconds to spawn
    - Shadow costs vary over game
      - Early on, food and wood expensive, spawn doesn't matter (since make few)
      - Mid-game, much food and wood, spawn makes it harder to pump out new units
      - End-game, no wood, spawn is priceless
- Use variability to add subtlety to game. Vary environment and vary shadow costs (ex: more trees to vary cost of wood)
  - Challenge for level designer
  - Expert players will appreciate

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Synergies (1 of 2)

*Synergies* are interaction between different elements of player's strategies (note, terms may be different than ch 2.2)

- Positive Feedback
  - Economies of Scale - the more of one type, the better (ex: wizards draw strength from each other)
  - Economies of Scope - the more of a set, the better, or advantage of combined arms (ex: trident and net, infantry and tanks)
- Negative Feedback
  - Diseconomies of scale - first is most useful, others have less benefit (ex: diminishing returns from more peasants entering a mine since get in each other's way)
  - Diseconomies of scope - (ex: mixed troops go only as fast as slowest)





## Synergies (2 of 2)

- Ideally, all go together at once, but can emphasize
  - Ex: Chess is a game of positive feedback
  - Small advantage early on, exploited to crushing advantage
- Game of negative feedback needs other ways to keep interesting
  - Ex: trench combat makes a "catch-up" factor, or as get far from base, supply long grows, game lasts a long time
  - Ex: *Super NES NBA Jam* - catch up setting as an equalizer
- Be aware of each



## Final Word on Gameplay

- Need to make sure choices interact
  - Ex: no fun winning just because out-optimize guy on resource production
  - Ex: no fun if winning just because know right thing to do else lose → no game, just forgone conclusion
- Want choices to interact with choices of opponent, want it to *depend*







## Review: Use Tools from Toolbox of Interesting Choices

- Strategic versus Tactical
- Supporting Investments
- Versatility
- Compensating Factors
- Impermanence
- Shadow Costs
- Synergies
- Groupwork:
  - Use 1-2 in a game about graduating from college. Discuss.



## Interactivity versus Gameplay

- Interactivity is the *heart and soul* of entertainment software
- Ex: Kick the soccer ball around, practice headers, bicycle kicks, etc. (interactivity, like soccer)
  - Play a game of soccer on the pitch (interactivity + gameplay)
  - Can you have gameplay without interactivity? Maybe. But even so, gameplay without interactivity could be fun (ie- television), but would start wondering if time is better spent doing something else
- Gameplay is important because it allows you to take the experience someplace
- Interactivity is more important than gameplay
  - Interactivity without gameplay can be fun
    - Ex: Black and White, Sims



Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris

## Kinds of Interactivity (1 of 2)

- Can interact in many ways - game designers sometimes restrict themselves to facts
  - Ex: if you hit w/BFG, do 50 points damage
- Think broadly. Player could potentially:
  - 1) Directly control characters (Ex: move Laura Croft)
  - 2) Affect world (Ex: make *Stronghold* guys "insane")
  - 3) Influence characters actions at one remove (Ex: give weapons, like Zeus to a hero)
  - 4) Influence at two removes (Ex: provide inspiration, like a Muse)
  - 5) Decide who to follow, rather than what to follow (Ex: observer mode)
  - 6) Select what parts are interesting and give more time to that (Ex: like a child with a bedtime story, Saahil likes the hero build up and powers most)

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Kinds of Interactivity (2 of 2)

- (Ask: others?)
- In the above list, how many are done?
  - 1 most everything, 2 for changing difficulty
- But many not done:
  - Why can't you say to computer opponent:
    - "Hey, let's build up a big army before we fight" or
    - "Don't attack me since I'm having fun building"
  - Or, why can't you switch sides in a battle?
- Avoid making mutant versions of films, novels or even board games
  - Use *imagination* for interactivity

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris



## Concentrate on "Why" not just "What"

- Doesn't have to be about what happens
  - Ex: ER. Noah Wylie is avatar. With a sick patient - does he ... follow rules, give experimental drug, play basketball?
  - But ... not only way to be interactive. Instead, follow Noah, switch to patient, go to other Dr., back to Noah (learn about characters, the "Why")
- Drama unfolds because of understanding of characters
  - True in non-interactive drama, so true in games, too
  - Ex: D&D dungeon, series of rooms w/monsters. Much richer if "why" behind scenes. Why were dwarves there? Why did they die? How orcs break in?
- Goal of entertainment is to make audience *care*
  - Use interactivity as a way to powerful technique to help this




## Core Design

- Brief, since overlaps material in
  - IMGD 1000. Critical Studies of Interactive Media and Games
- Topics
  - What is a Game (Overmars + Ch2)
  - Gameplay (Ch 3)
  - Game Balance (Ch 5)
  - Look and Feel (Ch 6)



Based on *Game Architecture and Design*, by Rollings and Morris



## Game Balance - Introduction

- Beauty in balanced games
  - Like Rolls Royce or Ball Machine in Airport
- Game without balance often unsatisfying and wasted effort (parts not in balance not used, so wasted effort)
- Broadly, game balance includes:
  - **Player-Player** - advantage only in skill (can be luck, but should be equal to both)
  - **Player-Gameplay** - learning curve matched by reward
  - **Gameplay-Gameplay** - Composite longbow does twice damage, should cost twice \$

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Mini-Outline

- Broadly, game balance includes:
  - **Player-Player** (next)
  - **Player-Gameplay**
  - **Gameplay-Gameplay**





## Player/Player Balance (1 of 2)

- Players should have "fair" chance of winning
  - advantage only in skill (can be luck, but should be equal to both)
- Ex: *Virtua Fighter* (ask: who has played?)
  - Say, Sarah Bryant beats Lion every time?
  - Does that mean unbalanced?
    - No, look more closely
- Suppose friend said could beat everyone as Sarah Bryant all the time. Would say "prove it"
  - Would only be a problem if beginner as Sarah always beat expert as Lion
  - And if could choose characters? Sarah versus Sarah?

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Player/Player Balance (2 of 2)

- Allow to arrange victory by *skill and judgment*
- Avoid results mostly as stroke of luck
  - Right from the start or magnified as game progresses (ex: start close to gold mine provides escalating advantage)
- Simplest way is to have symmetry
  - Same weapons, maneuvers, hit points (sports do this)
  - (But note, not always the most interesting. Want different moves on fighters, say. More later.)

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris





## Symmetry - Example

- Two heroes square off for duel, poised in kung fu stance
- Hours pass. Days pass.
- Breeze comes by, spec of dust in one's eye
- Blinks, frowns then bows
- Know result without fight ... tiny asymmetry enough to decide outcome
- If breeze or dust decided game, is that ok?
  - No ... you'd want your money back!
- Don't want to decide by factors out of control
  - Keep symmetric

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Symmetry

- Symmetry is fine in abstract games (ex: chess, even basketball)
- In realistic games, would be problem (ex: U.S. versus Iraq, game symmetry would be bothersome since not realistic)
- While easy, kind of an insult
  - Ex: LOTR BfME Warg's same as horses ... but Wargs can bite in book/movie!
- Better is *functional* symmetry that is not obvious

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris





## Symmetry in Level Design

- Can avoid obvious symmetry
  - Ex: each player has impassible region on flank (water or mountain range)
    - Knights and soldiers can't cross
    - Later on, advanced units can cross
    - Choice of unit depends upon barrier
      - Mountaineers to storm, ships to cross sea
      - Or bluff, and then go up middle
- Players can choose asymmetric start location
  - Should not be deciding factor (Ex: you choose downwind port, so you lose - like dust in eye)
  - Avoiding making start location critical decision
  - Ex: potential mines in many spots, so not critical

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Symmetry in Game Design (1 of 2)

- Make all choices for players functionally the same
  - Ex: *Warcraft 2* - humans have griffons and orcs have dragons; both flying toughies.
- But even slight differences make interesting
  - Ex: *Warcraft 2* - orc player's runes explode, making use in mountain passes good
- "Just broken" asymmetry easier to manage than total asymmetry (can compensate)

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris





## Symmetry in Game Design (2 of 2)

- Making choices for players different, yet balanced is tougher
- Ex: *Starcraft*: Protoss, Zergs, Terrans - all very different (Same with *Command and Conquer - Generals*)
  - Imagine the hours of playtesting!
  - Recommend only for deep pockets
  - *Starcraft* is often a "benchmark" against which to judge other RTS game balance
- Also, if re-creating historical simulation, tradeoff between fairness and authenticity
  - Ex: *Conquistadors vs. Aztecs* - Aztecs are doomed, but may be no fun. Certainly not symmetric

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Mini-Outline

- Broadly, game balance includes:
  - Player-Player
  - Player-Gameplay (next)
  - Gameplay-Gameplay







## Player/Gameplay Balance - Introduction (1 of 4)

- Means remembering that the business is about interactivity - think about player's relationship to the game
  - Ex: If had to "tune" the T.V. every time channel surf, would not do it much
  - Likewise, should not struggle for small reward
- Ex: *Baldur's Gate* (ask: who's played?)
  - Attributes are 3-18 (ask: why?), can re-roll if don't like. So, re-roll until all 18's. Ugh. Test of endurance!

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris




## Player/Gameplay Balance - Introduction (2 of 4)

- Player/Gameplay balance entails balancing challenges against player's improvement curve
  - (Draw picture)

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris





## Player/Gameplay Balance - Introduction (3 of 4)

- Problem
  - Game starts easy (most do), and stays easy too long
    - Player quits from boredom
  - Game starts easy, then gets suddenly hard (add timing or requires new skill)
    - Player quits from frustration
- Ideally, game difficulty adapts to skill of play (track stats, etc.).
  - Ex: (Give a lot of health for newbie, guy that gets wounded.)
  - Great! But a *lot* of work to build and testing to get right

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Player/Gameplay Balance - Introduction (4 of 4)

- More common, have difficulty settings (player manually selects)
  - Still challenge of making the "Normal" level right.
- Compromises
  - Could ask player up front some questions (ex: have you played FPS before?), then recommend setting
  - Could have player do tutorial level, then recommend setting
- Getting more difficult
  - Many RPG's have monsters get tougher with level
    - Ex: Diablo 2 does this
    - But boring if that is all since will "feel" the same
  - Want *widening options*, too
    - Ex: character gets more abilities

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris





## Sub-Outline

- Again, true balance is an art, but three guidelines that can help
  - 1) Reward the player
  - 2) Let the machine do the work
  - 3) Make a game that you play with, not against

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Reward the Player

- Player will have to learn. Will make mistakes (discouraging). Want to offset with reward when do something right
- Ex: *Virtua Fighter*, takes longer to learn complicated moves
  - Sarah's backflip. Reward comes from seeing flip (eye candy) and punch in kidneys (payoff)
- Best when expand game options
  - Ex: "Now with backflip, I can see new use for reverse punch"
- In general, better to reward player for something *right* than punish for something *wrong*
  - Punishment makes players not want to play

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Let the Machine do the Work

- Interface should show player the world and let him/her manipulate
- Computer is tool to take care of wide-range of tedious tasks
  - If tasks are not fun, don't make player do them
- There is a blur of boundary between chore and game feature
  - RPG could provide graph so player can manually draw map as explore ... but is that fun?
  - Ex: In D&D, can tell D.M. "we go back to the dungeon entrance". Easy, fun. What if a game makes player walk back over map that has been seen? Boring, no fun.
  - Ex: *Myst* provided lightning bolt move to avoid tedium
  - (Ask: other examples?)
- Also, if game option is no-brainer, consider AI taking care of it

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris




## Make a Game that you Play With, Not Against

- Consider great story, graphics, immersion but only progress by trial and error ... is this fun?
- Ex: crossbowman guards exit
  1. Run up and attack. He's too fast. Back to save point (more on save points next).
  2. Drink potion. Sneak up. He shoots you. Back to save.
  3. Drop bottle as distraction. He comes looking. Shoots you. Back to save.
  4. Drink potion. Drop bottle. He walks by you. You escape!
  - Lazy design!
- Should succeed by skill and judgment, not trial and error


Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris






## Specific Example - The Save Game Problem (1 of 2)

- Designer talking about RPG
  - *Designer*: "I've got a great trap!" ... platform goes down to room. Player thinks treasure but really flame throwers. Player is toast!
  - *Tester*: "What if player jumps off?"
  - *D*: (thinks it's a loophole) ... "Ok, teleport in then toast"
  - *T*: "What is the solution?"
  - *D*: "There isn't one." (surprised) "It's a killer trap. It will be fun."
  - *T*: "So, there's no clue for player? Charred remains on platform or something?"
  - *D*: "No. That's what the 'Save' feature is for."




Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Specific Example - The Save Game Problem (2 of 2)

- Should be used only so players can go back to their Real Lives™ in between games
  - Or maybe to allow player to fully see folly of actions, for exploratory and dabbling
- Don't design game around *need* to save
  - Has become norm for many games, but too bad
  - Ex: murderous level can only get by trying all combat options
- Beginner player should be able to reason and come up with answer
  - Challenges get tougher (more sophisticated reasoning) as player and game progress, so appeals to more advanced player
  - But not trial and error



Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Mini-Outline

- Broadly, game balance includes:
  - Player-Player
  - Player-Gameplay
  - Gameplay-Gameplay (next)



## Gameplay/Gameplay Balance - Introduction

- Consider Warcraft 2, with dozens of units. Nearly perfectly balanced.
  - No unit costs so much don't want
  - No unit too weak can do without
- Either got lucky *or* lots of play testing (probably the latter)
- Strong RPS relationship - have to play all units, none are dispensable



## Gameplay/Gameplay Balance - Introduction

- Challenges when balancing aspects of gameplay?
  - Want variety of interesting choices, rather than single, dominant choice
  - Best choices depend upon choices of other players (or on AI)
  - As a designer, not easy to see how frequently different choices will be worth making, but need to know to balance game
- Sounds like catch-22? Can use simple concepts to make first guess
  - Then lots of play testing to fine tune! ☺

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Game Balance

- Establish the value of each game choice
- For game balance, each choice must
  - not be reducible to simple value (else easy to determine if dominates or dominated)or
  - factors must even out
- Example where evens out: Pirate game
  - Dreadnoughts > Galleons > Brigantines
  - All have identical functions
  - If Dreadnoughts 2x more power, then (for balance) Galleons should take  $\frac{1}{2}$  time to spawn so will have 2 Galleons for each Dreadnought

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Game Balance

- Example where doesn't even out: *Starcraft*
  - Mutalisks fly over any terrain, but cannot fight other fliers
  - Wraiths are not as tough, but can attack other fliers
  - Observers can see enemy, but not fight
  - There is no expression for values since different things!
- Another example, in the Pirate game
  - Instead of spawn rate, compensate by making Dreadnoughts slowest, Brigantines fastest
  - Getting more interesting gameplay, but what about balance?

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris




## Game Balance

- Two levels to balancing: *component* and *attribute*
  - Component balance deals with relative values
    - Ex: how much does it "cost" relative to others?
  - Attribute involves interaction of abilities
    - Ex: how important is speed relative to damage?
- Envision as a set, where relative values based on one component only:
  - Speed: Brigantines > Galleons > Dreadnoughts
  - Tuffness: Dreadnoughts > Galleons > Brigantines
  - Range: .
- Use weights to combine to get average set combining all factors based on perceived importance
- Then, adjust component values so all units are useful
  - How to adjust? Lots of play testing!
- Often need tools so level designers can balance
  - Ex: new\_tank2.gm6

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris









## Component versus Attribute Balance

- Mnemonic to remember:
  - Component choices are about artifacts
    - Ex: "Hmm. Should I use the ion cannon or laser?"
    - Depends upon the tactical task
  - Attribute choices are more abstract regarding use:
    - Ex: "I should sneak past troll or take extra health"
    - Depends upon the strategy
- Attribute balance is harder (set of all problems)
- But if can get approximate picture of better strategies, can tweak component costs to get game balance  
(Next, component balance)




Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Intransitive Game Mechanics (1 of 5)

	Rock	Paper	Scissors
Rock	0	-1	+1
Paper	+1	0	-1
Scissors	-1	+1	0

- Payoff, match your choice with opponent
- Suppose I always picked rock. Then opponent would notice and pick paper. Then I would start to always pick scissors, then...
  - spiral to center of triangle where all options equal
  - only break even, like thermodynamics
- Note, too, that player must chose all. No option that can do without (or opponent will exploit). It is *balanced*.



Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris

## Intransitive Game Mechanics (2 of 5)

- Suppose scissors costs most, rock costs least
  - May use rock more often, scissors less
  - But wait, that would mean paper less useful, too... what is optimum choice now?
- Suppose scissors costs 3 ki, paper costs 2 ki, rock costs 1 ki and hit does 5 ki damage

	Rock	Paper	Scissors
Rock	0	-4	+7
Paper	+4	0	-4
Scissors	-7	+4	0

Ex: I choose scissors, you choose rock. Ki diff is -2. Plus damage is -5, so -7 total.



## Intransitive Game Mechanics (3 of 5)

- Say payoff is R, P, S and frequency r, p, s
  - Want to know how often used (r, p, s)
- Net payoff R is  $(0 \times r) + (-4 \times p) + (7 \times s)$ 
  - 1)  $R = -4p + 7s$
  - 2)  $P = 4r - 4s$
  - 3)  $S = -7r + 4p$
- Sum must be zero (zero sum game, whatever one player gains other loses. Both cannot have net gain.)
  - $R + P + S = 0$
- All net costs must be equal else would favor (remember, triangle example)
  - $R = P = S$

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Intransitive Game Mechanics (4 of 5)

- Solve: (3 equations in 3 unknowns)
  - (eq1)  $-4p + 7s = 4r - 4s$  (eq2)  
 $11s = 4r + 4p \rightarrow s = (4r+4p) / 11$
  - (eq2)  $0 = 4r - 4(4r+4p)/11$   
 $0 = 44r - 16r - 16p$   
 $0 = 28r - 16p \rightarrow p = (7/4)r$
  - (eq3)  $0 = -4(7/4)r + 7s$   
 $0 = 7r + 7s$   
 $r = s$
- Ratio  $\rightarrow r:p:s = 1 : 1.75 : 1$ 
  - Rock and Scissors used 27%, Paper about 46%
  - Probably not what expected. Often result ... if one option more expensive, others are most affected

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Intransitive Game Mechanics (5 of 5)

- Enhance to more choices.
  - Ex: could do combination moves.
    - Rock + Scissors + Scissors = *Garden Shears*
    - Could be countered with *Paper Weight*
    - Strategy becomes complicated
- Can use technique to:
  - Adjust costs to fit envisioned game play
    - Ex: if it turns out "too many" tanks relative to infantry
  - Justify spending more artistic assets
- Fine, all is balanced. Players must avoid predictability because clever opponent will exploit.
  - But that is barely above where have only 1 choice!
- To balance so interesting, must have attribute factors that interact (remember, the Battle of Hastings)

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris





## Other Intransitive Relationships

- Can extend RPS? Sure (otherwise not useful)
- More than 3 options → Table 5.3 and Case Study 5.5
- Less regular are 4 options → Table 5.4
- Figure 5.7 discusses another 4-way relationship
  - Infantry dominated
  - But, looking further, infantry has attribute that only one that doesn't have to move
    - Can hold territory! (In game that needs that)
  - Ex: In AoE, could "teleport" supplies by building base. Didn't need to hold territory. Infantry useless. Even making them cost less doesn't (expansion pack). Still great game, but didn't need.

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Combinatorial Explosions

- How many components should there be to make interesting?
    - Too few? Then becomes trivial (Ex: in Hastings, only way to change power base is to put infantry on hill)
    - Too many? Then too hard to have skilled play
  - Rule of thumb: N factors that could modify core mechanics, and each boolean (hill or not, rain or not ...) →  $2^N$  possible combinations ... explodes rapidly (remember,  $N=24$  gives about 16 million combinations)
    - Err on the side of caution
- "In Populous (EA god-game), should have lots of characters or half-dozen? Noticed would be easier to understand game experience with few, versatile units rather than many specific ones."

Richard Leinfellner (executive in charge of Bullfrog)

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## Design Scalability

- Intransitive designs are inflexible
  - If have balanced relationship and remove one, will have dominated strategy
  - Ex: RPS and remove R ... always choose S!
- If project lead says behind schedule, so don't include 5<sup>th</sup> orc type
  - Elegant design falls like a house of cards!
- But is relatively easy to add components
  - Doesn't have to be symmetrical, can be redundant or useful in only a few cases
    - Ex: scout, or special spell
- Lesson, if you are going to scale, scale *up* not *down*

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## A Game Balance Checklist (1 of 3)

- *Player-Player*
  - Ensures game is fair
  - Especially important for multiplayer games
  - Symmetry works for this, but asymmetry may be needed or more appealing (try "just broken")
  - Make sure any asymmetry doesn't magnify imbalance as game progresses
- *Golden rule*: a player should never be put in an unwinnable situation through no fault of their own

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris





## A Game Balance Checklist (2 of 3)

- *Player-Gameplay*
  - Ensures player never becomes frustrated.
  - Continually brings player back for more.
  - Interface should not present obstacles.
  - Small rewards are needed to guide player
    - Ex: Fancy animation or new powers
  - Best rewards *widen options*
- *Golden rule*: The game should be fun to learn as well as to play, and it should be *more* fun the more you master it

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris



## A Game Balance Checklist (3 of 3)

- *Gameplay-Gameplay -*
  - Ensures no element redundant or useless
  - Can do briefly by making factor table for each attribute (Ex: fire, range ...)
    - Make sure each best at something
  - RPS ensures each component dynamically best rather than statically so
  - Oblige player to alter tactics
  - Don't have to have every component equally useful
  - But cost, availability and ease of use should reflect value
  - Get right through playtesting
- *Golden rule*: all options in game must be worth using sometime, net cost of each option must be on par with payoff

Based on Chapter 5, *Game Architecture and Design*, by Rollings and Morris





## Bit Bucket



## Notes

- The rest of the topics are to be covered on students own time
- Or, possibly in class, as time allows





## Look and Feel

- Create a sense of alternate reality - *Immersion*
- Ambience
- Interface
- Storytelling

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Ambience

- Everything that contributes to innate look and feel of game
  - Not just spiffy graphics - GLFOPS and trilinear filtering
  - Rather, *how* graphics are used
- Two fighters on bare stage. Fine. How about dirty street, realistic crowd hooting and hollering. Dark skies...
  - Ex: "Fiery hell" when battling boss in Battlegrounds
- Ambience is about providing background for story
- Broadly - Sound, Vision, Touch

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris





## Sound

- Wistful guitar in Diablo
- Ethnic rhythms in AoE
- Stirring call to arms in Warcraft
- Whimsical in Q'Bicles
- Best does ambience plus gameplay
  - Ex: Thief
    - "Come out taffer", looking for you
    - "Just a rat", you are safe
  - Ex: LOTR
    - "Stirring" music when level nearly over (but can still die!)

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Vision

- The "look" of the game
- Concept art
- Broad strokes, not pixel finished detail
- Rough sketches of characters or settings



- Paolo Piselli

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Vision Example



Concept Art



Movie



## Touch

- Not really "touch", but physicality of games look and feel - *handling of game*
  - Ex: early animation characters did not move right - Disney pioneered with physical attributes that felt right, moved with weight
- Contrast
  - Ex: comic-book acrobatics in *Smash Bros*
  - Ex: bouncing vehicles in *Mario Kart*
  - Ex: realistic crashes in *Mid-town Madness*
  - Ex: super-players in *Lego Soccer*
  - Ex: realism in *Madden* (actually, guys 1.5 times faster)

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Interface

- Ideal is transparent
  - Ex: LOTR BfME novel way when click expands with choices
  - Ex: Status can be in formation or appearance (not health hit-points, say)
- Doesn't have to be invisible
  - Ex: racing game expects dash
  - Ex: flight sim interface can look like cockpit
  - Ex: less is more (small square more annoying than framing with interface)
- Can enhance look and feel

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Storytelling

- No need of story? After all, supposed to be *interactive*.
  - "If you want to tell a story, write a book."
  - Bah. Consider "choose your own adventure"
- Ex: Doom - two factions
  - 1) Strong setting and backstory enhance game
  - 2) "Story? We don't need no stinking story!"
    - Action takes care of itself
- Interactive can help user create story
  - Ex: *Half-life*
- Stronger - want to suspend disbelief of user but need to make them *want* to suspend
  - Ex: Starwars merely some sword fights and vehicle chases. Need to know who Luke is, why he's in the spaceship. Why the battle ...

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris





## Toolbox of Storytelling Techniques

- Best → not chunks of action with static facts
- Details revealed to audience - let them figure it out
- Get emotional involvement from audience
- Storytellers knew tricks for creating good stories long before Shakespeare - Game Designers should employ
  - Obstacles, Plot Points, Foreshadowing ...
  - (More next)

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Obstacles

- Old man runs to hero in inn. Says "Vampire on hill. You have to kill it."
  - Poor
- Old man enters inn. Avoids hero. Purchases crucifix from another. Mumbles "you better have one if you are in these parts."
  - Not great, but better. Has obstacle
- Viewer must find out himself/herself
- "Tricked" into level of acceptance not obtained if just told, too artificial

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris





## Foreshadowing

- A story depicts the intrusion of the world on status quo
  - Ex: AoE - settlement grows to large city
  - Ex: Total Recall - construction worker spy
- *Foreshadowing* occurs early, before intrusion, hints at what is to come
  - Ex: AoE - small bandits come, fought off
  - Ex: Total Recall - dream of spy

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Personalization

- Novice author - Ex: save the world, because big
  - But not compelling, so only *you* can save it
  - Still weak
- Need to add person reason so audience cares
  - Ex: you have two hours to save the world versus you have two hours to find your niece lost at sunset
  - Ex: Luke told must save galaxy. Why? Drawn in by personal (Princess Leia)
- Careful not to make personal hook in backstory - might skip

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Resistance

- Back of mind saying "it isn't true"
  - Need to pull them along
- Ex: Bruce Willis, drinking at dingy strip club. Two suits say "you must save president from terrorist." Does he jump up and get to work? No. Snarls "I'm retired." Takes another drink.
  - We want him to change his mind. Rooting for him before main character does.

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Plot Points (1 of 2)

- Importance of confounding expectations
  - Ex: Gandalf on quest to Mount Doom. Boring if that is exactly what happens
    - Gets killed early on (and comes back), not expected
- Adventure games benefit most, but can do for other games, too
- Aristotle - reversal, discovery, calamity
  - Ex: trying to save kid, causing her death (reversal)
  - Ex: finding Swiss account number on victim (discovery)
  - Ex: bomb going off, killing hero (calamity)

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Plot Points (2 of 2)

- Games, too.
  - Ex: strategy game - find cliff so army built up won't work (reversal)
  - Ex: tunnel for small commando force (discovery)
- *Whammo* every 10 minutes, turn story in different direction
  - Big ones ("Luke, I am your father") divide into levels or acts
- Movies - setup, conflict, resolution
  - But games whole season (40 hours), not one movie
  - Best if can integrate in game without cut-scenes
- Early plot points deepen mystery, later clear it up (not always completely)
- Overarching structure, hierarchical in plot points

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Suspense

- Ex: Unbeatable foe (Gorgon, only beat by lure to trap). With "save game" let hero try. After 9 times, try something else.
  - Death of lead character destroys illusion
- Instead, provide clues, suspense. Bodies, rotting, see NPC get eaten. Hear sounds. Can see gorgon survive rock crash.

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Dialog

- One picture worth a thousand words
  - Don't have needless dialog when visual will do
- Good dialog serves more than one purpose
  - Ex: telling about bomb. "How long?" Plenty of time ... smoke cigarette, call mom ... don't read War and Peace. Reminds of mortality.
- Don't tell what know, but also reveal
  - "Do you expect me to talk?"
  - "No, Mr. Bond, I expect you to die"
- Surprise

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## Resolution

- Should be:
  - Hard won - no reward is satisfying if too easy (most computer games not this)
  - Not obvious - don't want ending one been seeing for 10 hours (yet should still make sense looking back)
  - Satisfying - usually morally (hero wins) but could be aesthetically (tragedy)
  - Consistent - with character, style development
  - Achieve closure - resolve story
- Many examples of stories/games where above fails
  - Diablo 2 - defeat Diablo ... then 60 seconds to end.
  - Might and Magic 2 - long struggle, mystery. Very end, control panel ... 15 minutes to decode "Fourscore and seven years ..." Solved it, asteroid missed, thank you and go home
  - Ex: A Christmas Story - decoder ring drink Ovaltine

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris







## Change

- Stories set in interesting times
  - No "Sir Gawain shops for bread." Rather, "marries hag, one week until green knight chops head off"
  - Sometimes, return to normal
- Inner change is often point of story
  - No "Frodo lives in Shire with friends", rather "learns of evil, innocence to self-knowledge"

Based on Chapter 6, *Game Architecture and Design*, by Rollings and Morris



## What's Next?

- Art (2d, 3d, audio)
- Architecture
- Wrap up



## Ensuring Interesting Choices

- Kinds of choices in gameplay can involve options:
  - 1) Should sometimes be taken, sometimes not
  - 2) Timing is critical and depends upon context (upgrade armor or build more troops)
  - 3) Makes little difference whether taken or not
  - 4) Always worth taking (target nearest)
  - 5) Never worth taking (remove armor, pay guy for tapestry in Vici)
- First and Second most interesting
- Third valid, but really only "chrome"
- Fourth should be handled by AI
- Fifth should seriously consider removing

Based on Chapter 3, *Game Architecture and Design*, by Rollings and Morris

