Principles of Game Design

John Laird
University of Michigan
Last changed 9/8/05
[drawn in part from Doug Church]

What is a Game?

- How are games different than toys?
  - Games have rules and goals to be achieved.
- Some "computer games" are really toys
  - SIMS, Simcity, Nintendogs, …
- Games can have story, but …
  - Interactive, demands participation

Computer Games

- Completely new form of entertainment
- Completely new worlds to play in
- Allow players to take on a new persona

Game Technology: Computational Worlds

- Representing physical objects – real or imaginary
  - Terrain
  - Buildings (exterior and interior – walls, floors, …)
  - Game objects (furniture, balls, fluids, weapons, vehicles, …)
  - Animate objects (player, opponents, animals, …)
- Providing dynamics to world
  - Physics
  - Behavior: AI
- Supporting interaction
  - Graphics
  - Audio: dynamic sound, music, and speech
  - Input devices: speech?
  - Networking

Engineering: Traditional vs. Game

Traditional engineering:

- High precision
- Realism

Game engineering:

- High speed
- Low memory
- No spiking in resources
- Scalability – O(n) expected
- Believability
- Control
- Low cost development

True for graphics, physics, AI, audio, etc.

Cheat if don't get caught.

Game Design

- Create an experience for the player
  - Player should have fun, not the designer, programmer or computer
- Many different kinds of experiences
  - Some games are based on a story (dramatic arc, etc).
  - Put the player in his dreams, where he/she is the hero (Sid Meier)
  - Some are puzzles without stories are personification
- The player is in some environment
  - Abstract or representational
- The environment has rules of interaction
- The rules and some criteria for success define a game
- All managed by the computer

What Makes for Good Gameplay?

1. Pursuing and achieving goals (challenges)
2. Interactivity
3. Feedback about position relative to goals
4. Interesting choices required to achieve goals
5. Consistency and fairness
6. Avoid repetition

- Play testing to ensure that you get it right

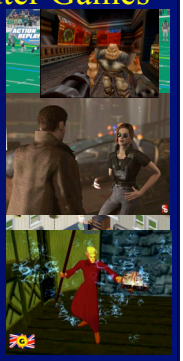
- This is what is behind "Fun" (which ≠ Reality)

Pursuing and Achieving Goals

- Always something to achieve
- Always achieving something
- Not too easy or too hard
- Often at least three levels of goals with rewards
  - Long-term goal [complete game]
    - "I can conquer the world."
  - Medium-term goal [10-30 minutes]
    - "I can take over a city."
  - Short-term goal [seconds to minute]
    - "I can win a battle."
- What are levels of goals from different game genres?

Common Goals in Computer Games

- Eliminate other players
  - Action games
- Score points
  - Sports games
- Get somewhere first
  - Racing games
- Solve puzzles
  - Adventure games
- Gain territory
  - Strategy games
- Improve abilities
  - Role-playing games
- Develop social relationships
  - Massively multiplayer games
- Play god
  - Simulations



Game Interactivity

Player's decisions determine success/outcome
- At least the player thinks their actions do
- Avoid decisions that aren't related to success of some goal

- Player is not just a passive observer
  - Can be fun, but is different kind of entertainment
  - *Always ask: What is the player going to do?*

- User interface should not get in the way of interactivity
  - Simple, consistent

Feedback

- Feedback at all levels so players
  - Know where they are
  - Know where they are relative to other players
    - high score list
  - Know where they are relative to goal
    - Dead and don't even know it
  - Know what they need to achieve
  - Know what is important in the world
  - Can use knowledge to make choices

- Game should not be about
  - how to get information out of the interface
  - how to randomly explore the world (unless that is "fun")
  - how to recall what just happened

Gameplay Variety

- Must develop & execute strategies/tactics to achieve goals
  - Applies across all levels of goals

- Interesting decisions to achieve goals
  - Different strategies/tactics
  - Tradeoffs between strategies/tactics: game balance
  - Avoid tedious goals/decisions: micromanagement
    - Let computer do those
    - Ask – if you take out the tedious activity, what decisions are left?

- New decisions enabled as game progresses
  - Few, simple decisions at first, and then more and more
  - Decisions have impact on how game turns out

Gameplay Consistency and Fairness

- There is consistency in the actions and associated outcomes for trying to achieve goals
  - Must be a *reason* for failure (or success)
  - Not arbitrary: Players know what to expect and can plan
    - A pinball game uses "pinball" physics *all* the time
  - Don't solve problems by unique, unlikely actions
  - Don't break suspension of disbelief
    - Kill self but dead body falls over wall, hits lever to open door
  - No "dead man" walking
- Fairness
  - Player thinks they have a fair chance – game balance
- Can still be plot twists, but must be explainable

Avoid Repetition

- Allow user to skip parts already seen
  - Skip cut scenes
  - Good save system
- User must to same thing over and over again
  - Endless waiting to heal
  - Lining up at spawn points waiting for an item to respawn

Game Design is Hard to Show

- A 20 second look doesn't tell you much about a game design
- Download demos… need to play to "get it"

Gameplay/Game Design Sins

- Poor production
  - Break the suspension of reality
  - Bad writing, bad voice acting, long load times, saving, …
- Linear plot/gameplay
  - Player's actions don't affect how the plot progresses
- Micromanagement
  - Player is forced to perform meaningless tasks
  - AI should take care of all the obvious choices
- Repetition
  - Player must do same action over and over again
  - Player must sit through same cut scenes every time they play
  - Have to replay 90% of level to fight boss
- Doesn't track user's learning curve
  - Should start easy and get harder as game progresses
- Poor game balance
  - Same strategy always works
  - Trial and error is not fun gameplay

Gameplay/Game Design Sins

- Not enough variety
  - Same graphics, objects, monsters, level design, sounds...
- Awkward user interface
  - Must do lots of mousing to do simple task
- Limited feedback
  - Player is confused about goals
  - Player is confused about current progress to goals: no map
- Inconsistency in story
  - There are not compelling and consistent goals for the player
- Dead and you don't even know it

Design Principles (Shigeru Miyamoto)

- Start with a simple concept
  - "running, climbing, jumping"
- Design around the computer's limitations
  - Character wears dungarees so easier to see arms move
  - Wears a hat because don't have to have hair
  - Has mustache because couldn't draw nose and mouth
- Minimize the player's confusion
  - What to do should be clear without consulting a manual
- The importance of play testing
- Incorporate a smooth learning curve
- Accommodate all skill levels

Sid Meier Game Design

- Player should have fun, not designer, programmer, or computer
- Begin your game with a great first few minutes
- Great game-play is a stream of interesting decisions that the player must resolve
- The inverted pyramid of decision making (have few decisions to deal with first, and then let them multiply until the player is totally engrossed
- Put the player in his dreams, where he or she is the hero

Nokia Series 40 Game Study: Top 10 Usability Recommendations

- **1. Provide a Clear Menu Structure**
  Use only one main menu, accessible with the left soft key. Keep the menu short. In general, use the left soft key for OK, select, and menu; use the right soft key for cancel and back.
- **2. Simplicity Is Key**
  If two solutions are equally valid, use the simpler. Make sure each entity in the game is unique, and not easily confused with any other. Provide different game modes only if they are truly different and valuable.
- **3. Provide Help When Needed**
  Keep help text short. If feasible, scroll text one screen at a time, not one line at a time. Display short text on the screen to explain new items, characters, and situations in the game. Provide a setting to disable in-game help. Provide a graphic representation of which keys are used for which functions. Do not expect players to read help text or force them to do so.

Recommendations 4-7

- **4. Be Relentlessly Consistent**
  Use the mother tongue of the user. Be consistent with the phone's UI, with game industry conventions, and within the game itself. Use the left soft key for OK, select, and menu; use the right soft key for cancel and back.
- **5. Don't Waste the User's Time**
  Allow her to skip the introduction. Do not require re-entry of data. Provide shortcuts and reasonable default values.
- **6. Use Natural Controls**
  Use the 2, 4, 6, and 8 keys for horizontal and vertical movement as well as the arrow keys; use the 1, 3, 7, and 9 keys for diagonal movement, if enabled. Use the 5 key as the action button. Design the game so that it does not lure the user into pressing two keys at once, since many mobile devices (and all Series 40 devices) do not support simultaneous keypresses.
- **7. Enable Save and Pause**
  Provide a simple save-game feature. Have the game auto-save when the user presses the red phone button - use the destroyApp() method to do this. Provide a pause mode (left soft key, which goes to the game menu); this can be done using the hideNotify() method. If the user quits the game from the pause mode, have the game auto-save.

Recommendations 8-10

- **8. Conform to Real-World Expectations**
  For example, when jumping or throwing objects, the flight path should be predictable. There must be no invisible barriers that the player cannot pass or holes that he cannot reach. Do not end the game arbitrarily. Implement a realistic physics model if relevant (for example, racing games).
- **9. Go Easy on the Sound**
  Provide sound for feedback, but ensure that the game is playable with the sound off, and provide an easy way to turn sound off within the game. No annoying sounds: not too loud, not too high-pitched. Avoid background music, if possible.
- **10. Implement a High Scores List**
  Tell the user what score he reached before asking for a name; provide the previously entered name as the default. Do not force the user to enter a name; make it optional.