Game Design Studio III (CMPS 172)
Course Overview, Creating an Effective Tutorial
Upcoming Events

- Nicole Lazzaro, President of XEODesign
  - Wednesday, April 11, 2012
  - E2 180, 11am

- Daniel Benmergui
  - Wednesday, April 18, 2012
  - E2 180, 11am

- Note: no class in PSB 110 on Thursday: meetings with game teams in the game lab instead (details later)
Introducing

- Peter Mawhorter
  - Is joining Chris Lewis as a TA for CS 172
High level goal for the quarter

- Have your game realize the full potential inherent in its game design.

- By choosing a particular kind of game, with certain features, on specific platforms, your game has a maximum potential for greatness.

- Our goal this quarter is to help you unlock this potential, and to come as close as possible to reaching this top potential.
Learning Goals

- This course has slightly different learning goals as compared to CS 171. Main changes over last quarter:

  - Gameplay experience tuning
    - Refining game controls
    - Creating a pleasing and unified end-to-end gameplay experience
    - Creating an effective tutorial system
    - Crafting levels to ensure strong pacing and flow
    - Developing a splash screen, menu system, credits page

  - Release engineering
    - Creating a final, complete, self-contained game image
    - Creating a complete web page

  - Art style:
    - Communicating with artists to develop and refine a game's art style
Course Format

- High-level organization of the course:
  - Two-week sprints, two releases during the quarter

- Week 1-2
  - Sprint I

- Week 2-4
  - Sprint II

- Week 5-6
  - Sprint III, end of Beta release

- Week 7-8
  - Sprint IV

- Week 9-10
  - Sprint V, Final release
Upcoming deadlines

- Tuesday (April 3)
  - Due: Sprint I plan
  - Due: Beta Release plan
  - Use sprint and release planning document formats from last quarter

- Friday (April 6)
  - Team status reporting

- Friday (April 13)
  - Sprint I ends
  - Sprint II begins
Deep dive meetings with TAs

- This quarter there is an emphasis on team meetings with the TA and with the instructor.

- Each team will have a TA specifically assigned to that team, and will have a regular “deep dive” meeting scheduled with the TA each week.

- Deep dive meetings
  - Can vary, but generally
  - TA plays the game in its current form, reviews playtest data from most recent playtest
  - TA offers specific feedback and critique, game elements to focus on
  - TAs can also help with technical problems teams may be encountering
  - TA will also be checking to see what progress has been made on items identified at previous week’s meeting
  - Teams should feel free to ask TAs to focus on certain areas, levels, etc. where they would like feedback
Finding times to meet with TAs

- Chroma
  - Peter, Mondays, 2pm

- Devil’s Bargain
  - Peter, Friday’s, 1pm

- Firewall
  - Peter, Thursdays, 11am

- Hello World
  - Chris, 3:30pm Wednesdays

- MicroVentures
  - Peter, Monday, 12noon

- Puzzle Defenders
  - Chris, 10am Wednesdays

- Sonar
  - Chris, 12noon Wednesdays
Meetings with instructor

- Each sprint the instructor will also meet with each team for 30-40 minutes.
- Meetings will take place in the game lab, and generally take place Thursday (week 1)/Tuesday (week 2) of each sprint.
- Similar format to the deep dive meetings:
  - Instructor plays the game, reviews playtest data
  - Team receives feedback on the game, develops list of items to focus on improving
  - Discussion with the team on how to resolve design and game feel challenges in the game
- Members of teams I am meeting with need to be present for the meetings.
- If I am not meeting with a team, no need to be present that day.
Meetings with instructor

- Thursday
  - Sonar
  - MicroVentures
  - Devil’s Bargain

- Tuesday
  - Puzzle Defenders
  - Firewall
  - Chroma
  - Hello World

- Meetings start this Thursday, April 5
Reminder: Lab Cleanup

- A reminder about teams responsible for lab tidying duty

- Week 1 (this week): Sonar
- Week 2 (week of April 9): Hello World
- Week 3 (week of April 16): Chroma
- Week 4 (week of April 23): Chroma
- Week 5 (week of April 30): Sonar
Game Tutorials

- Once upon a time, players of computer games were expected to read a manual in order to learn how to play a game.
- Memory limitations made it difficult/impossible to include an in-game tutorial
- However, many times players didn’t read the manual…
- … and today the expectation is that you should be able to play a game without having to read a manual or instructions page
- In order to be perceived as top-quality games, your games also need to have an in-game tutorial system.
Ways of educating the player

- Contrary to conventional wisdom, players actually do know a lot coming into your game.
- One way to teach people about your game is to make it so things they already know easily map onto the game and its interface.
- What are some examples?
Ways of educating a player

- Use real-world conventions
  - Provide interface elements that map to player’s innate understanding of the world
  - Example: player is being attacked by a monster. A button on the interface shows a weapon. Player clicks on the weapon button to attack.
  - People generally know to either block or attack when they’re being attacked (alternately, could have a button for “run away”).
Real-world conventions: example

- Plants vs Zombies
  - Is a tower-defense style of game, so want to convey fact that towers are stationary
  - People intuitively know that plants don’t move on their own

- From popular culture, people know that zombies move slowly
  - This is a good fit for the game’s single-screen, non-panning interface

- Currency system: uses coins and diamonds
  - People intuitively know that diamonds are more valuable

- "These decisions seem insignificant, but if you're a little off, it can lead to a confusing experience.” – George Fan, creator of PvZ
Ways of educating a player

- Use accepted computer conventions
  - With computers, people generally know to use the mouse to point and click.
  - Also: tool tips, drag and drop
  - The “ville” style social network games depend on this – most of their interface is pointing and clicking
Ways of educating the player (cont’d)

- Use standard gaming conventions
  - For keyboard: WASD for directional controls, ESC go to main menu, left mouse button: select, right mouse button: bring up detail
  - For controller: one thumbstick for movement, the other for camera, closest button to right thumb is fire, etc.
  - Most consoles have platform-specific UI standards that dictate uniformity of use of their controllers
  - With arcade games, there are limited numbers of buttons, so any buttons available are action buttons
Types of tutorials

- **In-game tutorial**
  - A series of levels in the game that educate players
  - The tutorial takes place in the game world
  - This has emerged as a best practice for most games

- **Standalone tutorial**
  - Presents information with the player outside the game world
  - Concern: players may view this as just another kind of manual, and start playing the game without going through it. They then become frustrated at the game’s difficulty.
  - A standalone tutorial makes sense for games that play to an experienced audience, such as a game sequel.
  - Can never assume all players have been through the tutorial.
Tutorials: progression

- Introduce a game’s features in an orderly sequence
  - Start with most general/most often used features
  - Proceed to more specialized and less frequently used ones
  - Might want to introduce a particularly cool or interesting feature earlier: go for wow factor.

- Introducing features
  - Game features can be introduced very slowly.
  - Typically introduce only one new feature per level.
  - Intersperse tutorial levels with non-tutorial levels, which only require players to use features they already know.
    - Perhaps using features in a combined manner
  - Might consider introducing some new mechanics very late in the game
    - Example: Zen Garden mode of Plants vs Zombies, which comes late in the game
Use of text in tutorials

- Better to “do” than to read
  - Players learn best by doing, rather than reading
  - If you can get your players to perform a mechanic at least once, then they will likely be able to do it again
    - Players remember action...reaction pairs very well
- If you have to put in text...
  - Keep it to a bare minimum
    - "There should be a maximum of eight words on the screen at any given moment” – Plants vs Zombies creator George Fan
  - Avoid modal windows
    - I.e., don’t pop up a window where you have to read something, then click “OK” to move on. Breaks the flow.
  - Display messages so they don’t disrupt the flow
    - Show messages in a passive way (in-game signs, floating text, etc.)
  - Use adaptive text
    - Some messages may only appear for some players
    - In Plants vs Zombies, players who didn’t get to put plants on left hand side received tips about this; other players didn’t see them.
Communicating without text

- Highlight user interface elements
  - Don’t describe user interface elements (“look for the row of hearts in the upper right corner of the screen”)
  - Instead: highlight a particular UI element
  - Typical: put a glowing outline around the element to draw attention
  - Can combine this with a popup, and element that appears on the screen to draw attention, then disappears in a bit

- Use animations
  - Instead of text, consider using a short animation to convey an idea
  - This is usually more powerful than text, and can be consistent with the game’s art style
Things to consider

- Can a player turn off a tutorial?
  - **Pro:** can be annoying to go through a long tutorial you have already completed
  - **Con:** players may turn it off when they really shouldn’t, then get frustrated by the game

- **Test it and fix it**
  - It is **very** hard to get tutorial levels correct.
  - Make this a focus of user testing (but, requires fresh testers)
  - A good place for automated metric collection
    - Time to complete a level
    - Failures and retries
    - Track when they give up or stop playing
  - Look for when players have forgotten earlier lessons later in the game
Keep it easy

- Make the game very easy in tutorial levels
  - Have tutorial levels start out very, very easy, and then gradually get more difficult.
  - It is very common for game designers to underestimate how difficult it is for other players to play their game the first time.
  - People lack confidence when they’re learning new things. Your game is a new system: you want people to learn, be confident, and then become invested in the game to learn more.

- University of California–Berkeley professor Jennifer Chatman, in an unpublished study, sought to see if there was some point beyond which flattery became ineffective. She believed that the effectiveness of flattery might have an inverted U-shaped relationship, with flattery being increasingly effective up to some point but beyond that becoming ineffective as the flatterer became seen as insincere and a “suck up.” As she told me, there might be a point at which flattery became ineffective, but she couldn’t find it in her data.
Weave tutorial into the game

- Some of the best game tutorials don’t even look like tutorials
- They are game where the first few levels have been designed in such a way to gradually introduce new game mechanics, in an intuitive way
- Most good SNES titles are like this
- Example:
  - A platformer game starts with a barrel rolling towards the player
  - If the barrel hits the player, they die
  - If the player hits the button closest to their thumb (the typical action button for the console) then they can jump over the barrel
  - This acts to teach the player about jumping
  - Downside: player may die a few times before they get it, frustrating
More resources

- **10 tutorial tips from Plants vs Zombies**
  - Tom Curtis, Gamasutra, March 9, 2012

- **12 Tips for Making a Game Tutorial**
  - [http://greyaliengames.com/blog/12-tips-for-making-a-game-tutorial/](http://greyaliengames.com/blog/12-tips-for-making-a-game-tutorial/)
  - Jake Birkett, Grey Alien Games

- **Eight Ways to Make a Bad Tutorial**
  - Ernest Adams, originally in Gamasutra, June 14, 2011