Flaming Hoop

Submit to Ira via Slack 5pm Friday **1/25**
Demo Demo in class on **1/28**

The goal of this project is to give you the opportunity to familiarize yourself with the tools you'll be using in this class, and to demonstrate competence. If you cannot jump through this metaphorical hoop by Monday, you will not be able to take the course this semester (though you're welcome to take it in the future).

In all future projects in this class, you will be working in a group as a specific role. Possible roles include:

- Programmer
- 3D artist (modeling, rigging, animating)
- 2D artist (painting, UI, 2D art)
- Audio designer
- Producer
- Game designer

You may choose to qualify for any/all roles that you'd like. Pick at least one role and complete all the tasks for that role.

General Hints

When you get stuck, ask the internet, your peers, or Ira.

I used a PC when creating this project, so the hints may not be as helpful for Macs (or different software versions). If you discover something that would be helpful in the future, please message me.

Producers and game designers have projects that require them to talk to other people in the course. Play nice!

Programmer

Tasks

1) Implement the game described in the appendix.
2) Export a build for the web, and upload it to your Hampshire website (or another website of your choice).
3) Send Ira the link. Bonus: also include a link to your source code on GitHub, or wherever you stored it.

Tools

1) Unity is free: http://unity3d.com
2) Some text editor of your choice

Hints

1) As always, presentation matters.
2) Keep your code organized and self-documenting.

3D Artist

Tasks

1) Create a model of a teddy bear.
2) Create at least one animation for it. A simple one is fine.
3) Import it into a paint program, and paint a simple X on the model.
4) Import the model into Unity and play the animation.
5) Export the world from Unity so that it's playable via the web.

Tools

1) Maya is available for free for students: https://www.autodesk.com/education/free-software/featured
2) Unity is free: http://unity3d.com
3) Sculptris is free: http://www.pixologic.com/sculptris/
4) Feel free to use whatever tools you prefer, of course.

Hints

1) I include steps 3-5 so you have a sense of what your teammates must do once you finish a model.
2) In the Unity Tips and Tricks document (also linked from the course website under Useful Links), Connie (’14 grad) wrote several things that might still be useful to artists. Enjoy!
3) If you want to export an OBJ file from Maya, the internet told me:
    a. Go to the Window menu
    b. Go to Settings/Preferences sub-menu
    c. Click Plug-In Manager
    d. Check the boxes next to objExport.mll
4) Sculptris seems to like triangulated objects. If you want to triangulate your model in Maya, the internet told me:
    a. Choose "Polygons" from the dropdown menu in the upper left.
    b. Select the object you want to Triangulate.
    c. Go to the Mesh menu.
    d. Click Triangulate

2D Artist

Tasks

1) Create at least 5 concept sketches for a teddy bear, the world that it lives in, or anything else that might go in the game.
2) Paint a model of a teddy bear. You can work with a 3D artist from the class to get their model, or use the one on the course website.
3) Create a play button, like one that might appear on a start menu for a game.
4) Import the painted model and play button into Unity.
5) Export the world from Unity so that it's playable via the web.
6) Add a link to your concept sketches somewhere so they're accessible.

Tools

- Sculptris is free: http://www.pixologic.com/sculptris/
- Unity is free: http://unity3d.com
- UVMapper classic is free: http://www.uvmapper.com/downloads.html
- GIMPShop is free: http://www.gimpshop.com/
- Feel free to use whatever tools you prefer, of course.

Hints

a) I include steps 3-4 so you have a sense of what your teammates must do once you finish a paint job.

Audio designer

1) Create at least 5 different sound effects related to a teddy bear. Possible ideas include: walk, throw, squish, laugh, etc.
2) Create at least 2 looping ambient tracks, max 30 seconds each: one for outside on a crisp winter day, one for inside just after waking up in the morning.
3) Create at least 2 different looping music tracks, max 30 seconds each: Imagine the casual puzzle game genre, maybe something like Candy Crush or Angry Birds. One music loop will be used during the pause screen – it should be mellow and relaxed. A second music loop will be used during the bonus round, where the player just earned a great reward (maybe once every ~10 games) – should be upbeat, triumphant, excited.
4) Import all your sound effects, ambient loops, and music into a Unity world.
5) Make each of them play (and stop playing) at the touch of a button. Get help from a programmer if needed, or watch some tutorials on the unity site (links included in the Useful Links section of the course website).
6) Export the world from Unity so that it's playable via the web.

Tools

- Unity is free: http://unity3d.com
- Audacity is free: http://audacity.sourceforge.net/
- Feel free to use whatever tools you want.

Hints

a) The Audio Designer will generally have less work than any of the other roles, so you may be working on multiple teams simultaneously.
b) I include steps 4-6 so you have a sense of what your teammates must do once you finish a sound effect.

Producer

1) Work with two other people of your choice in the class to create a task list for their projects. Help each person identify, prioritize, and track their tasks, and set appropriate deadlines. Pick two people with different chosen roles (and not another producer).
    a. Remember that it's their project, but hopefully you're helping them!
2) In a single shared area, collect all the files used and generated by each person, so someone else could continue working on it where they left off.
3) Reflect on all aspects of this project, including its intended goals, and identify ways for Ira to improve it. Document those ideas in some way, and share them with Ira, along with whatever task list and support docs you created for your partners.

Tools

- Google Docs (http://docs.google.com) and Dropbox (http://dropbox.com) are free

Hints

a) Producers often need to collect information from other people or help those people get things done, even when those other people are busy and have no actual obligation to help or talk to the producer. Therefore, your diplomatic skills are very important!
b) In case it isn't obvious, you will not only be judged on your ability to complete the task, but also on how nicely organized and accessible it is.

Game Designer

1) Create a game design document for a game that integrates with Twitch.tv in some way, that a small team of Hampshire students could create in half a semester (i.e. the document you'd hand off to your team).

Tools

- Google Docs (http://docs.google.com) and Dropbox (http://dropbox.com) are free

Hints

a) Game designers often need to talk with programmers and artists to understand what they can and can't do. Feel free to talk with them! Your diplomatic skills are very important.
b) In case it isn't obvious, you will not only be judged on your ability to complete the task, but also on how nicely organized and accessible it is.
c) The document should include all reasonable things that a game design doc should include:
    a. Overview
    b. Screenshot examples (wireframes) of all different modes/screens
    c. Gameplay details
    d. Tuning variables

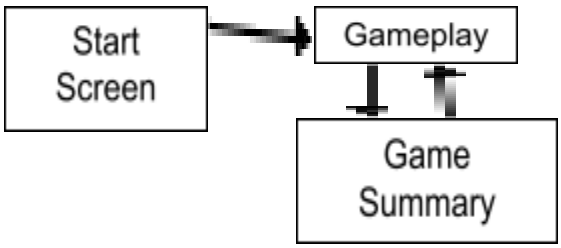
Appendix: The Game

Appendix: The Game is a single player action-puzzle game, playable in a web browser. The player uses the keyboard and mouse to interact with the game. (Note: We might eventually make a mobile version that uses only touch controls.)

We will be making this game as a tech demo, so art isn't a concern. If we have time later, we will add art to make it prettier.

The game has four states:

1) Start screen
2) Tuning variables
3) Gameplay
4) Game Summary



Start Screen
When players first start the game, they should see a very large [Start] button, and a smaller [Tuning Variables] button. When they click the [Start] button, the game starts. When they click the [Tuning Variables] button, they go to the tuning variables screen.

Tuning Variables Screen
This screen shows all the tuning variables in the game, arranged nicely in columns. The name of the variable is listed, along with a text box. If the player clicks on a text box, they can edit the value for that variable. If the player enters an invalid value (like a string for something that requires an integer), it's OK if the game breaks. The player deserves it!

There are two buttons on this screen:
[Save] will save the current values, and retain them for the rest of this session. If the player closes the browser window and opens the game again, it's OK if all the values are reset to their normal starting values.

[Menu] will return the player to the Start Screen.

See below for the specific tuning variables needed.

Gameplay
The game is timed, and takes 10 seconds (tunable). The current time remaining should be clearly visible to players. When the time gets to 5 seconds or less (tunable), the timer should turn red.

The game is played on an 8 x 5 grid of large white cubes (tunable). At the beginning of the game, the game secretly picks one cube to be the "appendix."

The player may click cubes at will. If it's the appendix, it turns green, the player wins, and the game ends. If it's not the appendix, it turns red, and the game continues.

The player loses (and the game ends) if time runs out before the player finds the appendix.

End of Game
The game can end in the following ways:

1) The player finds the appendix before time runs out. The player wins.
2) Time runs out before the player finds the appendix. The player loses.

Game Summary
Once the game ends, display the Game Summary screen. It should include:

- A big “Victory!” if the player won, or a bit “Defeat!” if the player lost.
- The players’ score.
- A big [Menu] button that will return the players to the Menu screen.

Tuning Variables
Below are the tuning variables needed in the game:

GameLength = 10 seconds
RedFontTimeThreshold = 5 seconds
GridWidth = 8
GridHeight = 5
Start button text on Menu screen: “Start”
Tuning variables button text on Menu screen: “Tuning Variables”

Original assignment designed by Professor Randy Pausch, updated by Professor Ira Fay.