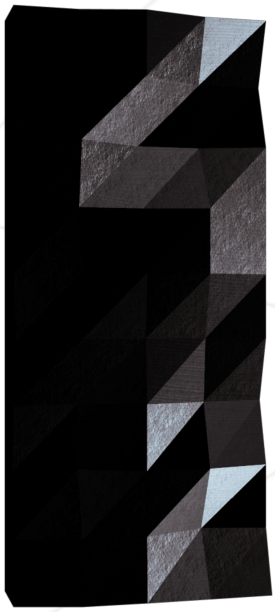

Game Design 2: Game Jam

Physics class is for collisions and explosions!

Kai kai@42.us.org

Summary: Now that you have your feet wet with the new library, allow your imagination to run wild.

Contents

Chapter I

Remain Objectively Oriented

This assignment is wide open. The goal is, simply, to create a game which helps you practice the technique of Object Oriented Programming. Several genres of game would work for this, including physics-based games with collisions, grid-based games with pieces that move around, or card games or battle games where various items have different abilities.

You could do this project in the Python version of Processing. Start by reviewing the objects tutorial for their overview of OOP.

If you use the p5.play library, study the reference material to get an understanding of the OOP patterns are already built in. In the p5.play documentation, the links at the left are the names of some classes which p5.play added on to the foundation of p5.js.

- Animation - a collection of .png images to be displayed sequentially.
- Camera - the object representing the viewer's perspective, which can move relative to the canvas.
- Group - a grouping of sprites which helps you keep track of, for example, all the "enemies" at once.
- Sprite - a character on the board that moves around.
- SpriteSheet - a collection of different images for the same character.

In this project we want to see you write a program which uses a system of classes and objects to organize your game world.

Chapter II

Some ideas

- Snake
- Single player agar.io (although there is a mutliplayer tutorial on youtube if you want to try it ;))
- Tetris
- Bejeweled
- Mancala
- Typeracer
- Minesweeper
- Bumper cars
- Angry Birds
- Plants vs Zombies, graphcis edition
- Chess
- Hearts, blackjack, etc - other card games
- Clue, chutes and ladders, etc - classic board games
- The Incredible Machine
- Pinball, pacman, etc - classic arcade games

Feel free to simplify the game mechanics to make sure that your project is accomplishable within a week. Simple and polished is better for your pride than ambitious with lots of loose ends. You can always add features later, next session.

There will be a showcase and contest on Friday! Try to build a different game than everyone else and make it shine with your own creativity.

Chapter III

Evaluation

Aim to fulfill the basics of a well polished game for the grading scale:

- The game should welcome you and give you instructions on how to play.
- Does the game area display and update correctly?
- Does the program respond to user input until the game is won or lose?
- Is there a win condition and a lose condition, and the game tells you when one of those has happened?
- After a win, lose, or tie, the program should ask you if you want to play again. It should play again if you say yes, and exit cleanly if you say no.
- Does the program handle bad input (like a number instead of a letter, a move that has already been taken, or a move that is off the board) gracefully? Bad input shouldn't cause the program to crash or act glitchy.
- Ask the programmer to talk about the part of the program that was hardest for them to build. They should be able to tell you something interesting about it, i.e., how many different types of objects they use in the game.
- Ask the team how they divided up the work, and ask each team member to describe how the code works on one part of the program that they contributed to.
- Give bonus points for how nice looking the output is. Consider symmetry, animation, and color!
- Does the program keep score, or a timer for how it takes the player to win, and display it?

