Project Description

**112 Jetpack Joyride!**

This is a game that’s based off the classic side-scrolling runner action game Jetpack Joyride. It has the same basic gameplay where the player has to avoid obstacles and enemy attacks as they move forward. The player would be able to collect coins and upgrade packs along the way. In this version of the game, additional features such as shooting bullets to kill the enemies will be added.

Competitive Analysis

Similar projects:

1. Mario Kart Rainbow Road
   1. <https://www.youtube.com/watch?v=pQq9cOSKgns>
   2. The project uses the classic Mario Kart gameplay and allows the user to play against the computer. It also has item boxes that give players various types of items to use during the game.
   3. This project is unique because it uses Game AI for the computer-generated character.
2. The Infinite Tower
   1. <https://www.youtube.com/watch?v=Str2HP2S1NA>
   2. The project is an adventure game in which a man is trapped in an infinite tower filled with monsters. There are multiple levels to the game and the player get through each level by reaching the top right corner of the map. In doing so, the player has to kill the monsters without getting themselves killed. There are items that the player can pick up that, for example, will give them an extra life.
   3. This project is unique because there is a boss every five levels and the difficulty goes up as the levels increase.
3. Run 112
   1. <https://www.youtube.com/watch?v=5TmMq6-mR4A>
   2. The project is a scrolling game where the player has to avoid falling off the “gaps” by moving left and right and jumping.
   3. The game is unique because the walls rotate when the player hits the wall. The gaps are also auto-generated.

My project will have some similar features to these projects, such as shooting bullets, item boxes, and side-scrolling features. It will be different because my game will have gravity and power-ups, as well as coin-collecting features that these games don’t have.

Structural Plan

My game will be organized into a couple python files.

* PygameGame: A modified version of the animation framework that is posted online.
* GameObject: A class that initialize the shared basic attributes of each object in the game. It also includes some basic methods such as update().
* The object files:
  + Player class
  + Obstacles class
  + Special attacks class
  + Bullets class
  + Coins class
  + Item boxes classes

Algorithmic Plan

The trickiest part of my game would be implementing gravity and the power-ups features. I will create variables to keep track of the player’s velocity and acceleration and use physics formulas to calculate its change in position (x = v0 \* t + ½ \* a \* t^2, v = v0 + a\*t). Each time the “up” arrow key is pressed, a positive upward acceleration is applied. I will also take into account the acceleration from gravity.

Timeline Plan

November 20-21: Wall hit features & Bullets

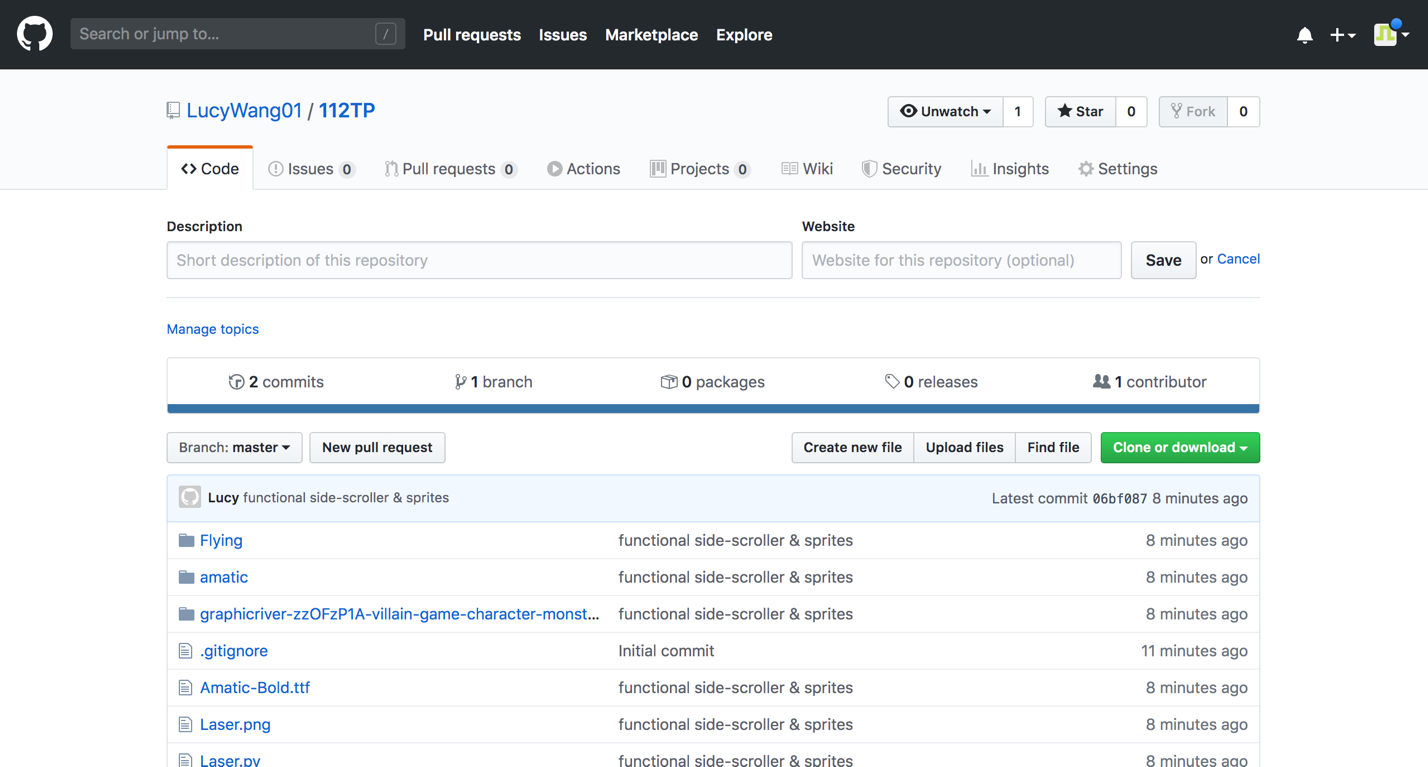
November 21: Coins & score tracking

November 22-23: Mode control (start screen, end screen)

November 24: miscellaneous stuff & debugging (add complexity)

Version control

I’m using GitHub to back up my files. I created a repository and uploaded all my TP files.



Module List

* Pygame

Storyboard

1. Start screen: I will add “play” and “help” buttons instead of doing “press any keys to start”.



1. Game mode: The player will die if they touch the bottom of the game frame or if they run into the lasers. The player will also have to avoid monsters or kill them with bullets (to be implemented). There will also be item boxes that give players upgrade packs (to be implemented).
2. End screen: I will add a score board here as well.