Project Name: 112 Joyride

Description: Based on the popular 2011 game Jetpack joyride, the user can continuously jump in the air to dodge obstacles and collect coins for points and see how long they can last and how far they can go.

Competitive Analysis:
https://www.youtube.com/watch?v=sdPzgUmyM1o&t=38s
https://www.youtube.com/watch?v=MASH1yAY0VI
I looked at previous 15112 project on jetpack joyride. In these previous projects, the coders implemented features such as leaderboards, powerups, multi-player, and intelligent terrain generation such that the maps get harder as the player progresses. My project will also contain those features with the addition of AI that give players hints on where to go next to get the most coins. It will also contain a features like enemies and algorithmic complexity to generating the coins, most likely using midpoint generation.

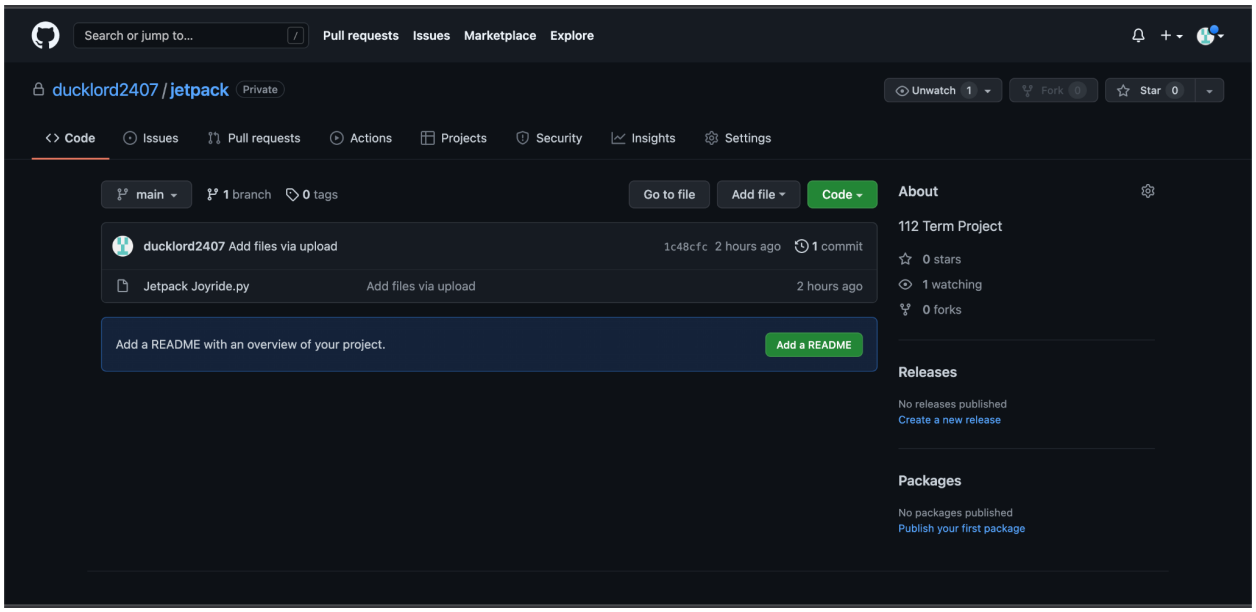
Structural Plan:
- Classes for player, powerups, enemies, coins, and obstacles
    - Keeps the position and size, and checks for collisions
- Function for gravity
- Functions for drawing players, enemies, scores, background, obstacles, coins
- Functions for solo mode or coop mode
- Functions for
- File for keeping track of the leaderboard

Algorithmic Plan:
The biggest algorithmic complexity is generating the terrain to be harder as the player gets further into the game and coding AI to give hints as to take the path that gives the most coins. For terrain generation, I will be grid out the display and placing elements in there, whether it is obstacles or coins. I will construct patterns of which the coins can appear in, and randomly place the patterns into the grid. I will also change the seed each time the players do not keep playing the same kind of pattern each time. The reason for using a grid is because the I can use backtracking starting from the players cell and try to get to the last column in the grid. If I am unable to due to obstacles in the grid, that means the map is unplayable and therefore reset the grid and place obstacles again so that it is playable, and the AI can give hints as to which direction to go. I will also put a limit on the number of obstacles placed, and increase the limit as the player progresses as to make the maps harder.

Timeline Plan:
Week 0
TP0 - Create user interface, get a running and moving player, establish gravity
Week 1
TP1 - Get decently through terrain generation and obstacles, powerups etc, get started on AI
Week 2

TP2 - Finsih MVP (Terrain generation, powerups, coins, basic AI) before deadline and start coding post MVP goals
Week 3
TP3 - Finish post MVP goals (leaderboard, multiplayer, buffed up terrain generation, more powerups)

Version Control Plan:
I am using Github to back up and keep track of changes of my code. I will make sure to update it everytime I work on the code. This is an image of my github with my code uploaded to it.



Module List:
Not using any modules other than PIL for sprite animations which are in 15112 notes!

Storyboard explanation
Slide 1 - Start of the game. User can choose whether to play in solo mode or multiplayer mode. User can also choose to view the leaderboard. When asked to play in solo mode, the player is asked whether they are new or existing player.
Slide 2 - Obstacles will be in the players path. There is AI that hints for the user to go a specific direction to maximize the amount of coins they can get. The score is displayed at the top
Slide 3 - In multiplayer mode, both players will work together to get as far as they can. Missile will lock onto the inital position of players.
Slide 4 - There will be powerups which if the player collects, they obtain an ability. For example, the green boost moves the player forward, and blue shield grants invincibility for some time.
Slide 5 - If the player hits an obstacle, the game ends. The score is displayed, and the player is asked if they want to play again. If they would like to, the leaderboard will be shown.
Slide 6 - The leaderboard is displayed with all the names of all players and their respective scores.

Updated TP2
No major design changes, except generation with coins changed from midpoint to backtracking instead.

Updated TP3
Added powerups and leaderboards. Powerups are done by turning on a variable in playerClass that makes the player invincible, and leaderboards are done using an external file. That is all.