112 Term Project Proposal

Hyun Jin Kim

- **Project Description:**
  The name of the term project is “Orbits”. It is a game where players have to launch planets so that they can get on a stable orbit around one or more black holes. They can play pre-built stages or build their own stages and play them.

- **Competitive Analysis:**
  There is a similar game on the mobile market called “Orbit” that I got the idea from. It has the same basic game mechanism and types of modes (pre-built stages and sandbox mode). I am planning to distinguish my project from this published game by adding obstacles and items that can make the game trickier and more interesting.

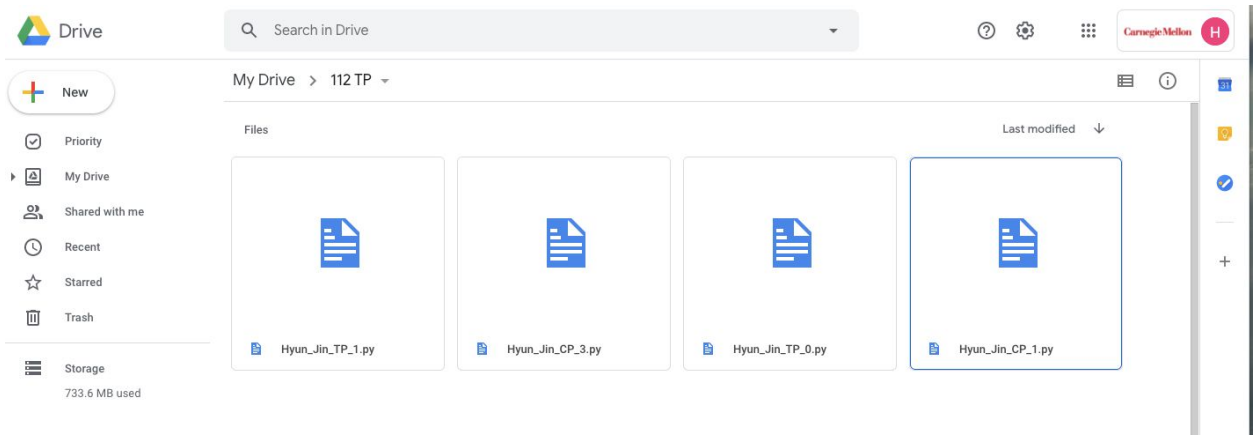
- **Structural Plan:**
  The finalized project would consist of the main python file, an image folder, and a sound folder. The main python file will have the main loop, class definitions, and some helper functions. The class definitions would define objects like BlackHole, Planet, Item, and Obstacles. They will each contain functions that determine how the objects will interact with each other and move. The image folder will contain image files for the logo and the buttons. The sound folder will contain a background music file and sound effects files.

- **Algorithmic Plan:**
  The most algorithmically complex part of my project would be the interaction between the planets and the black holes. This requires the calculation of gravitational force between them based on their positions and the application of that force into every step of a planet’s motion. This involves an algorithm that calculates the vector sum of the planets’ initial velocities and gravitational acceleration due to the gravitational force between them and the black holes. Checking if the planets are on a stable orbit also requires an algorithm that calculates the angular movement of each planet around the nearest blackhole.

- **Timeline Plan:**
  - 4/19:
    - Add items that can prevent the planets from flying off-screen by making them bounce after hitting the screen boundary
    - Add obstacles that can collide with planets and send them off-course
    - Add main menu screen and buttons leading to play screen and sandbox mode screen
    - Improve gauge bar graphics
  - 4/23:
    - Build sandbox mode functions
      - To save files

        - To load files
        - To add black holes
        - To delete black holes
    - 4/27:
        - Work on additional features
    - 4/29:
        - Final check-up for bugs

- **Version Control Plan**
  I am saving my files on google drive for version control.



- **Module List:**
    - PyGame

- **TP2 Update:**

I realized that training AI to find the model answer to a stage would be harder than expected and that it would not be feasible to accomplish in a few days. Hence I will instead try to add more basic features and create a survival mode to improve the overall completeness of this project. Additional basic features include changing the size of the planets, adding gauge bars to change the size of the black holes or the gravitational constant, and adding a pause function. The survival mode would involve launching a single planet over and over again to the next black hole until the planet gets absorbed by a black hole or flies out of the screen.

- **TP3 Update:**

I decided to focus on adding more small features and stages than creating a survival mode. I added functions for saving and loading high scores to stages and more features in the sandbox mode, such as adding items and gauge bars for changing the size of the black holes and the scale of gravitational constant.