Design Proposal

○

Project Proposal

Name: Sicheng Xing
Andrew ID: sichengx
Section: E

■ Project Description

The name of my project is “Doodle Jump Modified”. As suggested by its name, this project is a rebuild of the classical game “Doodle Jump” with self designed features.

■ Competitive Analysis

In comparison to the official version, this project will add extra features including new “jumping-boost” items, etc. Also, I plan to use algorithms to better simulate the jumping motion by considering the gravity.

■ Structural Plan

This project uses OOP extensively. There are 2 most important classes:

1. Class Platform

   This class organizes all attributes and methods needed for the platforms on which the player can jump. It is inherited by several subclasses including moving platforms and “temporary” platforms (only endure a certain number of jumps), etc.
2. Class Player

   This class is for the jumper.
3. And there are other classes including RocketBag, Armor, File, Bullet to make the programming easier.

■ Algorithmic Plan

1. This project uses carefully designed algorithms to randomly generate platforms which (a) adjust the difficulty as a function of a variable called density (platforms are more sparsely scattered at higher levels); (b) always ensure that the game is playable; (c) most platforms don’t overlap.
2. This project employs a non-linear motion using an extra variable g (for gravity) in each jumping cycle, which makes the motion a lot smoother than constant speed motion.
3. This project uses two coordinate systems. The first is the canvas coordinate system in which the origin is at the top left corner. And there is another more intuitive coordinate system in which y means the actual altitude (height) of the object. Then I use functions to convert between these two systems.
4. The screen is always vertically pivoted in a way that the it does not strictly center at the player (otherwise the screen will be too bumpy as the player jumps up and down). Instead, I keep track of the maximum height the player has reached so that the screen scrolls up as the player ascends, but in a visually stable way.

- Timeline Plan

Start to TP0: write the code for using sprite sheet

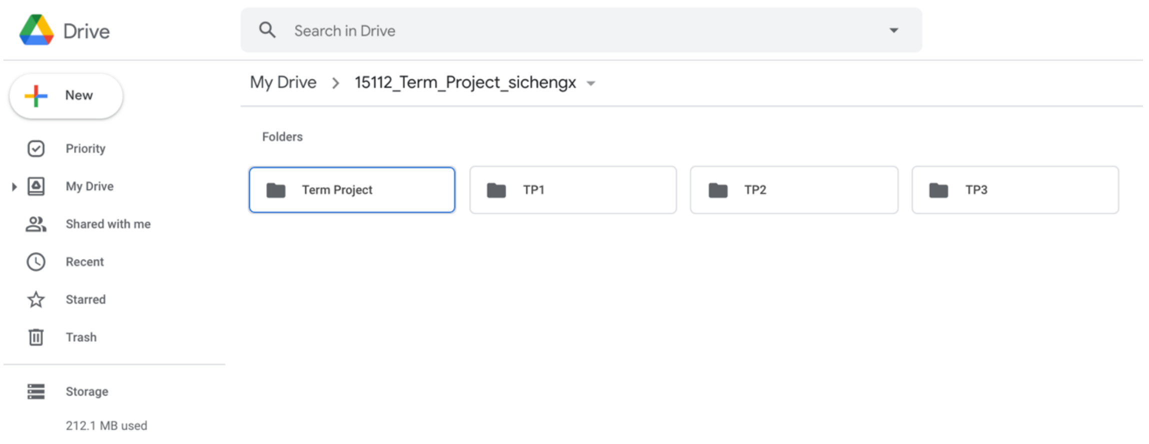
TP0 to TP1: write the basic game (abstract but playable).

TP1 to TP2: experiment with different parameters to find the most reasonable ones; replace geometric shapes with actual images; add gravity (non-linear motion); add two power-ups and one enemy.

TP2 to TP3: design the splash pages to make the app complete; use file system to keep track of the highest score; add new variables to record data which is used on the "gameover" summary page; add Bullet class so the player can attack the enemies.

- Version Control Plan

The code is stored in Google Drive.



- Module List

(No extra modules used)

1. PIL
2. cmu_112_graphics

- Citation

I used images for the game:

(1) playerWithNothing.png
https://poki.com/en/g/doodle-jump

(2) playerWithArmor.png
https://commons.wikimedia.org/wiki/File:Doodle_Jump.png

(3) playerWithRocketBag.png
https://icons8.com/icons/set/doodle-jump
https://bsmknighterrant.org/2011/09/27/doodle-jump-lands-on-the-android/

(4) armor.png
https://pngtree.com/so/cartoon-shield

(5) rocketBag.png
https://www.modelsresource.com/mobile/doodlejumpgalaxyprototype/model/30037/

(6) enemy.png
https://www.pinterest.com/walls360/doodle-jump-wall-graphics/

In addition to that, I used an online background remover:
https://www.remove.bg/

- TP1 update

This part is the same as in the time line. Build basic games (playable); use geometric shapes to represent players and platforms.

- TP2 update

Use actual images in place of geometric shapes; add gravity (non linear motion); add objects onto platforms (rocket bar, armor, and enemy).

- TP3 update

Add Bullet class which allows the player to attack the enemies; add welcome page and after-game summary page; use file I/0 to record the highest score.