**Project Plan**

Sihao Zhou (sihaoz)

**Project Description:**

My project is Flappy Bird. It is a game where a “bird” is jumping through pipes. If the bird hits on the pipe or the edge of the screen, the game ends and returns a score. The way that the bird jumps is that it will automatically falls down if the player does not click the screen, and if the player clicks the screen, the bird will go up by a little bit. A couple other features will include enemies; that is, if the bird hits the enemies, it will be game over and the game would restart.

**Structural Plan:**

File 1: modes file

appStarted(app)

keyPressed(app)

mousePressed(app)

timerFired(app)

Helper Functions

Draw Functions

**File 2:** graphics file

cmu\_112\_graphics

**OOP:** For OOP, I’m using the class “pipe” to store information of different pipes. Specifically, the pipes have two parameters, X and Y, which are the x and y coordinates of the pipe images.

**Important function names:**

1. checkColliion – detects the collision between the bird and other obstacles, including the pig, enemy bird, and the pipes. Using rectanglesOverlapped to check if the two pixel rectangles are overlapping
2. movePig – moves the pig in a projectile motion, using calculations for a projectile motion
3. moveEnemyBird – moves the enemy bird and making it chase the spot that is in front of and below the bird.
4. createPipe – using a dictionary classifying the midpoints of a pipe into “up” and “low” categories, and make the amount of two different pipes equal. Moreover, the pipes are generating more frequently and move faster as the game moves on. Also, the distance between the up pipes and the down pipes are getting smaller and smaller white the game is going.

**Algorithmic Plan:**

There are three complex parts in the project. The first is randomly generating the map with pillars on it. For that part, I’ll use the random module to generate a random number first and then make the range different to generate random pillar combinations on the top and bottom of my screen.

Another complex part of my project is making the making my whole screen move like it is moving forward. The way that I’m implementing this is that I’ll make the “floor” picture continue to move forward so that it looks like the bird is moving forward.

The third part would be creating the enemies (the pig and the enemy bird). Specifically, the pig would move in a projectile motion and the enemy bird would move towards the spot in front of and below the bird (and the spot changes as the bird’s location changes). For the projectile motion, I’m creating a function to keep the y position of the pig according to the pig’s x position. And for the enemy bird, I’m making the bird constantly choosing directions that fits the best way to approach to the target position, which makes it adjusting to the bird position.

**Version Control Plan:**

I send the code to File Transfer in Wechat everyday (Wechat is a social device with a could storing information, I can see files in File Transfer no matter which device I use).

I’m also uploading my tp into google drive to have multiple resources to keep my program.

图片包含 图示

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

**TP2 updates:**

1. created a non-random terrain generating algorithm
2. the distance between up and low pipes is decreasing (interacting with the variable **app.timePassed**)
3. the pipes are generating more frequently and move faster as the game goes on (interacting with the variable **app.pipeTimePassed**)
4. the midpoint of the two pipes are having a balance as the “up” pipes and “low” pipes will be around the same amount (using **dictionary**)
5. smoothed the game graphic by a lot
6. added a start guiding text

**TP3 updates:**

1. divided the pipe dictionary into 4 parts, more specific pipe generating algorithm