# Flappy Bird

Skylan Recana
CSE | BS Computer Science Student
University of Minnesota Twin Cities
15th of January 2021

## 1 Introduction

Flappy Bird is one of the most addictive mobile games in the history of gaming. It was so addictive that the creator himself had to remove it from both Play Store and Apple Store. This practice project introduces to Animation in Python using the Pygame library while simulating Flappy Bird with the exact sprites and objects.

## 2 Explanation

### 2.1 Creation of Variables

The pygame library uses init() and quit() to determine the start and the end of the game. The size of the screen, the font to be used for the entire animation, and the clock for limiting the frame rate per second are set for the animation.

To obtain the images, pygame uses image.load() to load the images onto the screen as well as transform.scale2x() to make the images bigger. The convert() converts the images into the best resolution. For the bird, there are three images illustrating the different bird movements for the animation. For the pipes, the pipe has the assigned heights for random spawning. The time.settimer() determines how fast the event of flapping the bird when going up and down and the event of spawning the pipes.

To obtain the sound, the mixer.Sound() is used to get the sounds necessary for the animation.

To override the high score, a text file is used to update the high score obtained from the game.

### 2.2 Visualization of Floors

The floor is drawn by having two floors connected to each other. This will create a moving floor illusion when the animation is started.

### 2.3 Creation of Pipes

The game uses two pipes: the top pipe and the bottom pipe. The size of each pipe is determined by getting random sizes from the pipe height array.

### 2.4 Motion of Pipes

The movement of the pipes are determined by modifying the centerx of each pipe and collects the pipes visible on the screen to eradicate the unused pipes to save memory.

### 2.5 Visualization of Pipes

The pipes are drawn based of the position of the pipes. If its the top pipe, the image will be flipped upside-down.

### 2.6 Collision

Collisions can be determined when the bounding rectangle of the bird collides with the bounding rectangle of any pipe. This is done using the colliderect() of a Surface (in this case, the bird) with the pipe. The sound will be played depending which conditions were triggered (death, swoosh, or fell).

### 2.7 Rotation of the Bird

The images are tilted when flying up

### 2.8 Animation of the Bird

The animation determines the image of the bird to be displayed depending on the bird index given.

### 2.9 Score Display

The score will be displayed depending on the state of the game. If game over, a game over image will be displayed as well as the highest score obtained. This will modify the text file that stores the highest score.

### 2.10 Score Update

Updates the highest score by comparing the score obtained from the game with the score from the text file.

### 2.11 Pipe Score

Determines whether the player gets a score when going to the pipes. If successful, the score sound will play and update the scores obtained so far.

### 2.12 Animation Loop

The loop runs in 120 frames per second per iteration. The loop handles the key events through event.get() to synchronize the animation with the key input from the player. It also checks if the game is still on or if the game is over. The collision is also verified every iteration as well as the gravity of the bird.

## 3 Future Work

Add a welcome image when the game starts before the animation is activated. Add more features such as options to color the bird/change the background/pipes.