Malika Dikshit

Super Flappy Bird

Project Description: My project will be a unique take on the classic Flappy Bird Game.

My Minimum Viable Project will be the Flappy Bird Game where the bird representing the user has to dodge the randomly generated pipes on the screen. The score of the user keeps increasing until the bird hits an obstacle. Once an obstacle is encountered, the game resets itself and places the bird back at the center of the screen with a score of 0. The user has three lives before the Game Over Screen is displayed.

To go beyond the MVP, my game will allow users to choose the universe they want to play in and each universe will have a different implementation of the flappy bird game.

Additional changes to the project have been described below in TP3 Update section.

Competitive Analysis: The MVP of the project will be similar to the online Flappy Bird game with the graphics being as close to the original game as possible. The graphics I will use will allow the wings of the bird to flap as it moves across the screen and allow the bird to tilt up and down depending on the user’s instructions. My project will have unique features which allow users to select different universes to play in, namely: the normal flappy bird universe, underwater and in space. Each of the universes will facilitate different implementations of the game and I will make use of different mechanics for the bird to move in the different universes. For example, in the space universe, there will be no gravity acting on the bird and the bird will have to dodge the obstacles which will themselves be moving vertically. In the underwater universe, the viscosity experienced by the bird will also play a part in determining how the bird will move on the screen.

Structural Plan: The final project folder will have a main file which will contain the source code for the game, a sound file and the images used in the game. Inside the main file, the project will be divided into functions that will deal with creating the obstacles, moving the different sprites on the screen, rotating and animating the birds and dealing with the collisions between the bird and the obstacles. There will also be a function to continuously update and display the score of the bird on the screen. The Term Project 3.py file imports the voiceController.py file which will be included in the submitted folder.

Algorithmic Plan: The trickiest part of the project was getting the frequency of the user in real time and then converting it into a note which I did with the help of the music21 library .The trickiest part of the project will be implementing the different physics and mechanics required by each universe. I will first focus on creating the MVP and the flappy bird game in the day universe with smooth graphics and a clean user experience. I will then work on the exact physics required to operate the bird in the other universes and then integrate the different themes into a whole where the user can choose their preferred environment (Voice or normal).

Timeline Plan: By the TP1 deadline, I intend to complete the major part of the MVP, i.e. the original Flappy Bird game. By TP2 deadline, I will add some extra features to the MVP like sound and between TP1 and TP3 I will work on the different universes that will be integrated into my game.

Version Control Plan: By the TP1 deadline, I will submit the MVP code to Gradescope and also publish it on my Github account. By TP2 deadline, I will once again submit the updated project to Gradescope, update it on my Github account and email it to my mentor. I will submit the final project on Gradescope on 1st December by 10 pm and publish it on Github as well as email it to my mentor.

Module List: Pygame, aubio, numpy, pyaudio, sys, random, math ,time, argparse, threading, queue, music21

TP2 Update:

I have modified my game to include 2 new environments. The user can select the environment they want to play in and each environment has a different implementation of the Flappy Bird Game.

TP3 Update:

I have made 2 modes of my game: Voice Controlled and Normal.

The voice Controlled version will implement voice controlled flappy bird where the program first asks the user for their vocal range: the lowest and highest possible note they can sing and then allow them to play the game where the bird's movement will be controlled by the pitch/frequency of the user’s voice. A higher pitch voice will move the bird upwards and a lower pitch will move it downwards.

The normal mode will display a home screen with instructions where users can choose 3 different environments: Classic, Underwater and Space.

I have also implemented a GUI for getting vocal range of user and improved the user experience and interface and provided instructions in the normal mode.