

Complex Engineering Project Report

PROJECT TITLE: Floating Mario Game

GROUP MEMBERS:

- ❖ Alina Saghir (CS-21121)
- ❖ Sadia Moeed (CS-21123)
- ❖ Syed Shahzain Hassan (CS-21128)

RUBRIC:

Criterion 1: Has the student selected a good game for implementation?

0: The chosen game is too simple. 1: The game is fit to be chosen for a class project. 2: The choice is different and impressive.

Criterion 2: How good is the programming implementation?

0: The project could not be implemented. 1: The project has been implemented partially. 2: The project has been implemented completely but can be improved. 3: The project has been implemented completely and impressively.

Criterion 3: How well written is the report?

0: The submitted report is unfit to be graded. 1: The report is partially acceptable. 2: The report is complete and concise.

Total marks: 7

PROBLEM DESCRIPTION:

Floating Mario is a very interesting game in which the sprite (MARIO) can jump in order to move safely through the obstacles (pipes) without striking. The player has to achieve the highest score by controlling the Mario such that it shouldn't touch with upper and lower pipes in the frame.

DISTINGUISHING FEATURES OF THE GAME:

- Three Levels of Difficulty,
 1. Easy Mode: The speed of Mario and the velocity to display the pipes is slow.
 2. Medium Mode: The speed is increased and the velocity of pipes increased as well.
 3. Hard Mode: Mario's speed and pipe velocity becomes too fast.
- Able to save the latest highest Score.

MODULES:

Pygame Module: To handle graphics, collisions, player inputs and sound effects easily

Tkinter Module: For a better and more engaging graphical interface.

FUNCTIONS IMPLEMENTED:

mainGame():

- This function is the gate to enter the game.
- It initializes the score from '0', and displays the sprite (Mario), the game's background on the screen.
- It further calls three functions in it:
 1. getRandomPipe()
 2. isCollide()
 3. HighestScore()
- This function is also responsible to keep track of Mario and the pipes which is to move. The pipes to the left display a new pipe after removing the currently displayed pipe as it reaches the leftmost position at the screen.
- It stores the count of upper-lower pipes in a list to increment the player's score.

getRandomPipe():

This function generates the random pipes (upward and downward pipes) on the screen.

isCollide():

This function will return true if the player is crashed (collide with the pipes)

HighestScore():

This function writes the highest scores in a file.

game_over():

- This function is called after the iscollide() function returns True.
- It simply displays the game over the image on the screen along with the updated highest score.
- Restarts the game again if the space bar is pressed otherwise quit the game.

DATA STRUCTURES USED:

- The Program uses List and Dictionaries to store the data such that the list to store the count of pipes whereas a dictionary is utilized inside the list to store the x and y coordinates of each upper-lower pipe w.r.t origin.
- We have tried to exclude the unneeded code chunks from our code as to improve the time complexity of the program.
- We gave the instructions or data in the form of a tuple to some functions (or in some steps) hence, making sure of the use of the data structure of a tuple in our code.

FUTURE EXPANSIONS:

- We can add more obstacles like bricks and bugs other than pipes to make the game more entertaining.
- We can add a dual-player mode so that people can get to play it with their friends which will make the game more exciting.
- The file which is currently carrying only the highest score can be further expanded to have the top 10 highest scores with the player's name.
- More moves for Mario other than jump can be added.
- We can use stacks to store the highest scores instead of files to further enhance the time and space complexities.

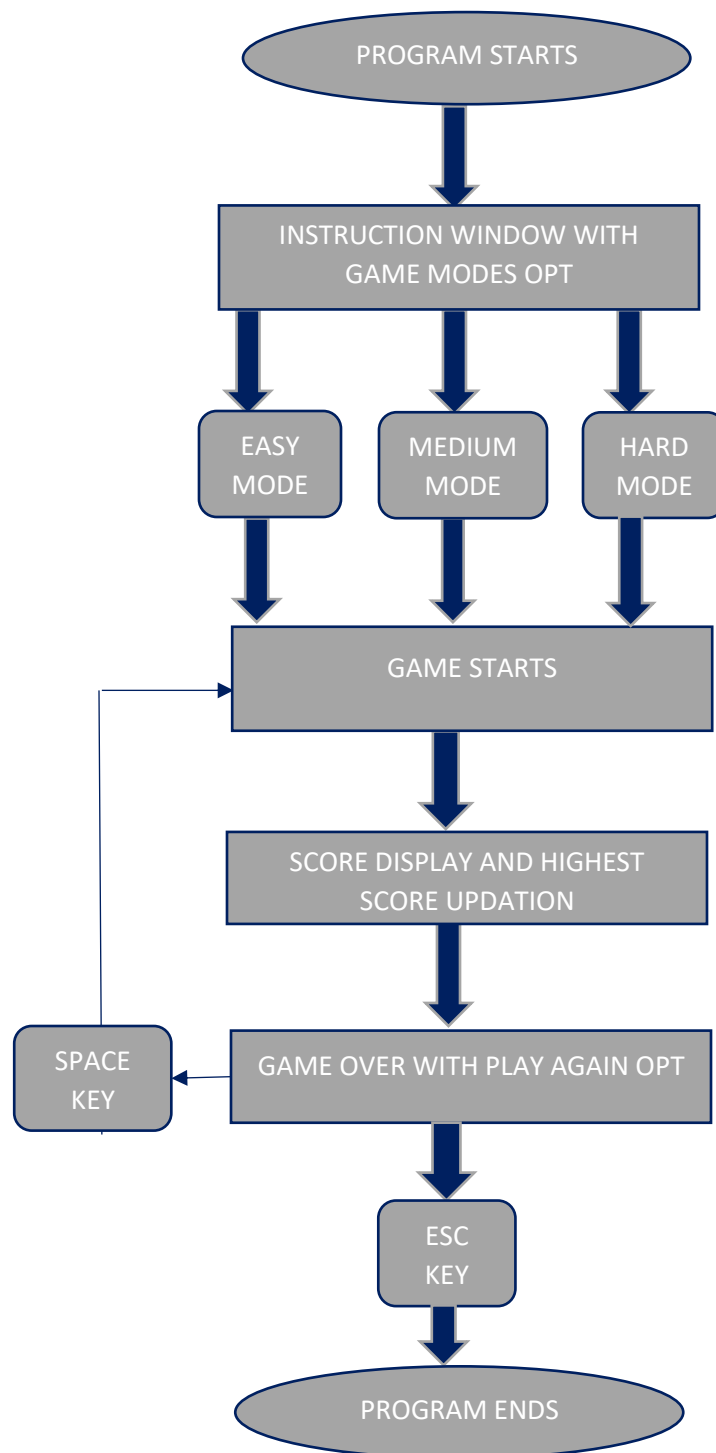
MOST CHALLENGING PART WHILE WORKING ON THE PROJECT:

Overall working experience on this project was really good. No doubt implementation of difficulty levels in the code gave us a little tough time but we went through it by working hard and continuous try.

ANY NEW THINGS LEARN WHILE WORKING ON THIS PROJECT:

While working on this project we learn to use python's pygame module which we didn't use before. Furthermore, our concepts regarding python basics and Tkinter became better.

FLOW OF PROGRAM:



SAMPLE TEST RUNS:

