Report of Project

Flappy Bird 2.0

Introduction

Flappy bird is a popular game that was available for smartphones. In this game a player simply has to maintain the velocity of the bird and have to keep passing between pipes and it will keep updating the scores. It is easy to play and quite fun, that is what makes the game more interactive.

Objective

What we learned in our software project course and Object-Oriented Programming with python, from that we were asked to do a project. As we loved the flappy bird game earlier. And what we know about **python**, **Jason,** data structures and problem solving was enough and met the prerequisite for creating the game.

Our objective was simply:

- To use what we learned about python, **JSon** and software project development
- To know how programming, problem solving are used to build projects
- To learn basic game development fundamentals that can be used for future projects
- To learn pygame

Tools and Platforms

We built the game with using best and latest tools that goes with the latest technologies. These tools are: -

- Python 3
- **JSon**
- Pygame
- Visual Studio Code

Methodologies

JSON

JSON is an open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types. It is a very common data format, with a diverse range of applications, such as serving as a replacement for XML in AJAX systems.

In our **project** to create a **user profile** and **save** the game **score** and other **data** we used **Jason** file format. Its very easy to understand and implement. We used this because its supported by almost every operating system and modern browsers as well, which makes it really very flexible to use. It will also help us do further improvements and add related features as it is widely supported in all platforms.

Pygame

In this game we used pygame which is a cross-platform set of python modules designed for writing video games. It includes computer graphics and sounds libraries designed to be used with the python programming language. It is built on top of the highly portable SDL (Simple DirectMedia Layer) development library. Pygame can run across many platforms and operating systems. By using the pygame module, we can control the logic and graphics of your games without worrying about the backend complexities required for working with video and audio.

What we know and must before using pygame is:

- Python 3
- Pylint

Game Development Methodologies

To build this game we had to know about game development methodologies. We learned very basic but important principles that has to be followed while making any games. Which are: -

- Game Loop
- Events
- Sounds
- Sprites

Game Loop

A game loop is the overall flow control for the entire game program. It's a loop because the game keeps doing a series of actions over and over again until the user quits. Each iteration of the game loop is known as a frame.

Events

Events in game where the flow of the game is determined by events such as gamers actions, sensor outputs or messages from the game itself.

Sprites

Sprites are the assets of the game like graphics of background, design of the main character and other elements that would be needed to run the game. They play a major role in any game to make it more interactive.

Sounds

The audio experience of any game makes it interactive and plays a major role in games.

To run the completed project our system, need to meet very few hardware and software specifications

Hardware and Software Requirements

- Any modern machine with windows 7,8 or 10
- 50 MB of space
- 512 Ram (min)
- Processor speed 1.5Ghz or higher

Limitations

The game is a console project and it doesn't run outside of the IDE we used. Also, there is no pause option in the game. The screen size also small which is as small as smartphone screen. The game graphics can be improved as well.

Possible Future Improvements

As the game is a console project, it can be converted into a executable file by making an installer for the game with the help of NSIS or other programs that helps to create installer for windows or the operating system programs. The game can be also converted into a full-size desktop game just by changing the display size it creates for the game. Instead of pygame, other game engines like: panda3d, OpenGL can be used to make the similar project.

As this project is the clone of the actual game, we can simply take inspiration from it and by changing and upgrading few things a new game can be created which could be something different as well.

Project Member:

- Raduan Ahamad **ID:**191-15-12943 **Sec:** O14
- Sakib Rokoni **ID:**191-15-12961 **Sec:** O14
- MD: Tariqul Islam **ID:**191-15-12963 **Sec:** O14

---Thank You---