PUI Final Project Write-Up

Part 1: Application Overview

The purpose of this application is to recreate the popular mobile game Flappy Bird using WebAssembly.

Some components of this include the bird, the pipes, the score, and the background. The goal of this game is to make the bird pass through as many pipes as possible. In order to do so, the user must flap the bird just enough to pass through the middle of the two pipes. If the bird collides with the pipes or falls onto the ground, then the game stops. A numeric data conveyed through this application is the score, which updates every time the bird successfully passes through the pipes.

This is a very interesting app as it allows the user to constantly interact with it. It is simple enough that almost everyone can play. However, it is not necessarily “easy” since there is no upper bound for how high the score can go, which keeps the users entertained and interested to continue playing.

The target audience of this application is people from age 9 to 30. It is intended for people who don’t want to invest too much time into a game but rather just want to kill some free time.

Part 2: Interaction

- Click the game container to start/restart the game
- Press the spacebar to flap up the bird.

Part 3: External Tools

Name of tool: WebAssembly

Why did I choose it?

I have heard of WebAssembly but never used it before, so I thought it would be interesting to experiment with it. I also didn’t know that WebAssembly can be used to implement a game, so I decided to try it out through this project. I am also really intrigued by the idea of front-end development without the use of Javascript.

How did I use it?

WebAssembly served as the skeleton of my application. I used it to implement the entire game from back-end to front-end. Two big parts of it consist of the components, which adds the front-end styles and structures, and the models, which contains all the back-end game logics. This allowed me to organize and structure the code in a clean and efficient way.

What does it add to the application?

It allowed me to add cool features like a score tracker as a component of the game. It allows for easy updates to any changes to the front-end. It also allows the game to be played locally, which makes it more accessible as no internet connection is required.

Part 4: Iteration

I added a score feature in my final iteration. I added a Score Model in my application which I update every time the bird successfully passes through the pipes. I added this feature to allow the users to constantly keep track of their score in the game, instead of just mindlessly playing.

Part 5: Challenges

One of the big challenges I encountered was using WebAssembly. I have never used it before and I also have no experience with C#. As a result, I spent a lot of time trying to figure out what structure to organize the code, how different parts of the code can be connected, how to update the front end with no javascript, as well as the syntax. If I was given more time, I would definitely work on adding more features to the application (i.e. different game modes, sound effects). However, since I spent a lot of time figuring out the basics of WebAssembly, I didn’t get enough time to implement as many features as I would like to.