Flappy Ghost

By: Rishi, Peter, and Sam

Roles

Rishi Sahoo
Project Manager and UI/UX Design

Ismail Iqbal
Product Manager and Release Manager

Peter Agaiby
Quality Assurance

Sam Lippert
UI/UX Design

Game Synopsis

Game Title

01

- Flappy Ghost

Gameplay Mechanics

02

- The player controls the bird with the space key.
- Based off our experiences with Karel missions
- This will be a solo game collaborated on Unity

Game Description

03

- The player must navigate through obstacles.

Style

04

- Colorful 2D graphics with a vintage/retro style.

Communication

05

- We will work as a group, collaborating on a daily basis
- Use Discord to communicate, and solve our technical problems.

Team Dynamics

Team dynamics play a crucial role in large software engineering projects because they directly influence communication, collaboration, and overall productivity. Effective team dynamics foster a positive work environment, ensuring that diverse perspectives are integrated, leading to more innovative and robust solutions. On the other hand, poor dynamics can lead to miscommunication, redundancies, and understanding and optimizing team dynamics can mean the difference between a project's success or its stagnation.

Team Dynamics continued . . .

In our group, we recommend our team members give full participation when it comes to communication efforts. As the project leader we are very welcome to open-mindedness. We are open to new ideas and perspectives, which can help foster an innovative and collaborative environment. Some strengths we have are members who are already heavily involved with club activities related to computer science and are adept in programming languages such as C# and Swift.

Starting Design

Main Design

We intend to use a space in the wall for the bird to maneuver through.

Wall

We will use a Wall to represent obstacles

Ball

We will use a bird as the main object.

Obstacles

We will use squares for the obstacles

Background

The game will be replicating the famous game Flappy Bird.

Possible Challenges

Scheduling

- As busy people, we will likely not all be able to work at the same time. We will have specific roles

Technical

- There will be a learning curve trying to apply our coding knowledge to learn Unity.

Algorithms

- We will have to work with multiple complicated game aspects.

Unity

- We will have struggles with members who aren't as fluent using Unity.

Testing

- We will not have an easy manner of testing our code while programming.

Github integration

GitHub Projects is a tool that helps you plan and track your work on GitHub. You can create projects that integrate with your issues and pull requests, and customize them to suit your needs and processes. You can also use command-line automation to manage your projects more efficiently. Command-line automation is the process of using scripts or commands to perform repetitive or complex tasks without manual intervention. For example, you can use command-line automation to create, update, or delete projects, add or remove items from projects, set or change custom fields, and more.

RELEASE MANAGER

GitHub Repository

- I will set up the GitHub repository and make sure that the proper Git/GitHub conventions are followed.
  - A README file will be created in order to explain what the repository contains, how to install and use it, and provide other important information. This will make it easier for my fellow contributors to understand the project.
  - I will utilize branching over forking, as they allow us to develop features and debug errors freely in a contained area. Unlike a fork, a branch remains part of the original repository (like a tree branch), which is ideal for this project.
  - Proper file-naming conventions will be employed, with lowercase file names separated by hyphens, rather than underscores to make it consistent and easily readable.
  - Issue or pull requests will be created discuss and review the potential changes with collaborators before committing to the changes. This will make it easier to review and merge our work., and ensures the codebase remains clean and merge conflicts are resolved swiftly.

QA METHOD TESTING

- Test Cases: We will create detailed test cases with specific steps, expected results, and inputs for the software or system under test. This structured approach helps ensure that every aspect of the application is thoroughly tested, and it's easier to determine if the actual outcome matches the expected one.

- Boundary Testing: Test the boundaries of the system by deliberately using values at the edge of acceptable input ranges. This helps identify issues related to minimum and maximum limits, which can be critical for ensuring the system's robustness.

- Exploratory Testing: I could also interact with the software or system without predefined test cases. I will explore the application to identify defects, assess its functionality, and determine if the outcomes match their expectations. It's a more unscripted and creative method that can help discover unexpected issues or behavior.

GitHub Integration Continued

GitHub CLI is designed to work seamlessly with GitHub features, allowing users to perform a wide range of GitHub operations directly from the command line. This includes creating issues, pull requests, repositories, and more, which aligns with the needs of developers who prefer or require command-line tools. The commands and syntax of GitHub CLI are consistent with common command-line conventions, making it easier for users to learn and remember how to use the tool. The output is also formatted for clarity, which aids in understanding the results of the commands executed. GitHub CLI supports scripting, which means that commands can be written into scripts and reused across different projects and environments. This promotes efficiency and consistency in workflow automation. GitHub CLI can be installed on multiple operating systems, ensuring wide accessibility. By enabling command-line automation, GitHub CLI saves time and reduces the need for manual intervention. It allows for quick and direct manipulation of GitHub resources, which can significantly speed up development workflows. Github CLI has a set of commands for working with projects. You can use gh project to list, view, create, move, or delete project columns. You can also use flags and options to specify the project name, number, owner, body, state, and other parameters.

In summary, some benefits of GitHub CLI are:
Speed: CLI allows you to implement commands quickly, and is faster than using a graphical interface to navigate menus
Resources: Fewer computing resources are required to implement commands than a GUI
Efficiency:: CLI enables you to automate repetitive tasks at any given time
Power-User: CLI provides access to certain commands which are unable in GUI