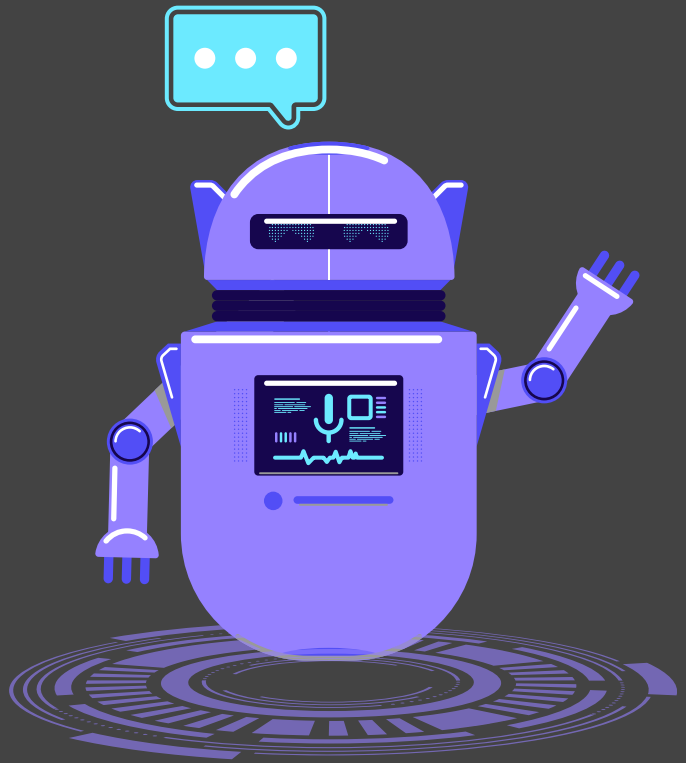
# Github Integration

Project Manager Notes



# TABLE OF CONTENTS

# OBJECTIVES

## My Aim

My aim as the Project Manager is to call meetings, discuss project goals and establish a schedule for everyone in the group to successfully accomplish the delivery of our game.

## THE GOALS

Goals inside a company are usually specific and measurable, with clearly defined deadlines and outcomes. The groups goals help focus the actions of the organization and ensure resources are used effectively

# Phase 2 – System Design Document (SDD)

A System Design Document (SDD) describes design goals and considerations, provides a high-level overview of the system architecture, and describes the data design associated with the system, the human-machine interface, and operational scenarios. For this Unity Project , the system requirements would include the hardware and software needed to run the Unity game effectively. This includes a compatible operating system (Windows 7 SP1+, macOS 10.13+, or Ubuntu 16.04/18.04). The system architecture of a Unity 2D game involves defining a high-level structure of the game, including the main game loop, scene management, and the interaction between different game objects and components. This also includes the use of Unity's Entity Component System for efficient game object management. In Unity, system components are typically represented as GameObjects and there attached components. For a 2d game, fundamental components include Transform, Sprite Renderer, Camera's, and Collider 2D. Each components design will depend on its role within the game. System interfaces in a Unity game can refer to how different systems within the game interact with each other. This could be how player input is handled, how data is passed between scenes, or how game objects interact with each other. Unity provides various built-in systems to facilitate these interactions. The data model for a Unity game could include the game state, player data, game settings, and more. This data can be stored in many ways, such as scriptable objects, PlayerPrefs, or external databases. The SDD document for a Unity game would detail all the above steps, providing a comprehensive overview of the game's designs and architecture. This document should be kept up to date as the project evolves.

# Stakeholder Communications

As the Project Manager of Flappy Ghost, there is a current lack of communication regarding status of our project. One helpful solution to this problem is to have a change in schedule where we will commit communication notes and updates from the stakeholders ensuring that they are informed of our progress and any changes. We will do this through a change in our management from an organized schedule. I plan to implement 10 minutes at the beginning and end of every class including a 5 minute break halfway through our class days. Every member of the group will make sure they have a task assigned to them during the first meeting of every class and I will make sure at the end of every class that progress has been made. Every member of the group will be responsible for completing the tasks assigned to them on Github as well as committing any completed work in the process. The stakeholder will be informed of all the progress as well as the individual ratings of each members progress during the meetings scheduled by me.

# Sprint – 12

During development, we encountered several challenges that required attention. One issue we faced with Peter (our QA Analyst) was our character intermittently switching from left to right, disrupting gameplay flow. To address this, our development team is diligently working on refining the control mechanics to ensure a seamless and consistent experience for players. Additionally, our background initially appeared blurry, detracting from the game's visual appeal. Through optimization techniques and adjustments from Sam (our UI/UX designer) to image resolution, we successfully enhanced the clarity and crispness of the background, enhancing the overall aesthetics of the game. Another obstacle we encountered from testing our product was the inability for the game to restart automatically upon collision with obstacles, causing frustration for players.  Ismail, our Release Manager was able to actively implement a restart feature, ensuring that collisions trigger a prompt for players to restart the game, thereby maintaining engagement and fluid gameplay.