# Turnbased Golf Technical Design Document

Alex Mallon

Daniel Perez

## Introduction

Turnbased Golf is a single player minigolf game with the objective of finishing the game with the fewest turns possible. Players have the option to collect scissor paper rock icons which serve only to demonstrate inventory capabilities.

## Game Mechanics

- Main Camera attached to the putter with tank-like left-right turning movement.
- Left click and hold to drag putter back and flick forward as a way of increasing force output upon the ball.
- Once the ball stops moving after the turn, the putter (with camera) will be transported to the new resting location of the ball.
- The hole will have a trigger to end the game once it collides with the ball, which will take the ball and the player to the next level.

## Game Elements:

- Simple level design to make the objective clear and easily achievable.
- In built physics engine with manipulations to physics materials for the balls and scripted strength changes to the putter.

## Development Tools and Resources

- Unity is the game engine of choice.
- Probuilder to create quick levels and assets.
- Class recordings and slides, as well as youtube tutorials for game elements we're less familiar with.

## Art And Assets

The assets and materials used will be very basic, as the objective of this project isn't to showcase artistic abilities.

There is a low likelihood of using free assets from the Unity store as basic shapes made in probuilder will suffice.

Different balls and putters with different properties and capabilities will be differentiated by their colours.

## Version Control

Github will be used for collaboration and data recovery.

## Testing Plan

Continuous testing will be done as the game is being developed, to ensure a playthrough is achievable and the intended elements function properly.

## Debugging

- Issues and breakpoints that persist will be noted in commits in case the project needs to be pushed before issues get fixed.
- Testing functions with input keys in order to quickly determine functionality.
- We will need to read the commits from others when pulling origin

## User Interface & Diagram

After the Unity Splash Screen, the player will be taken to the main menu. They will have the option to begin playing, or to quit out of the game back to desktop. The 'play' option will bring them to the game level that will have them set up in front of the ball with their putter/golf club.

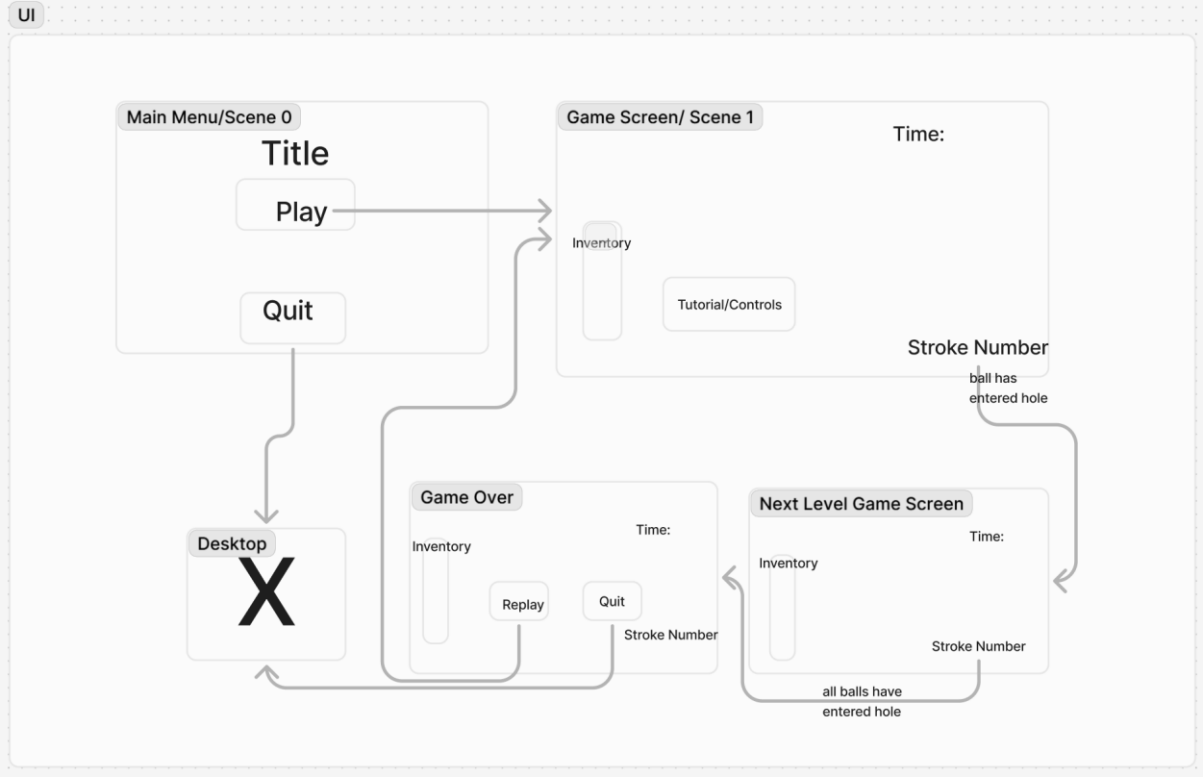
Tutorial will take the form of UI elements at the start of the first gameplay scene.

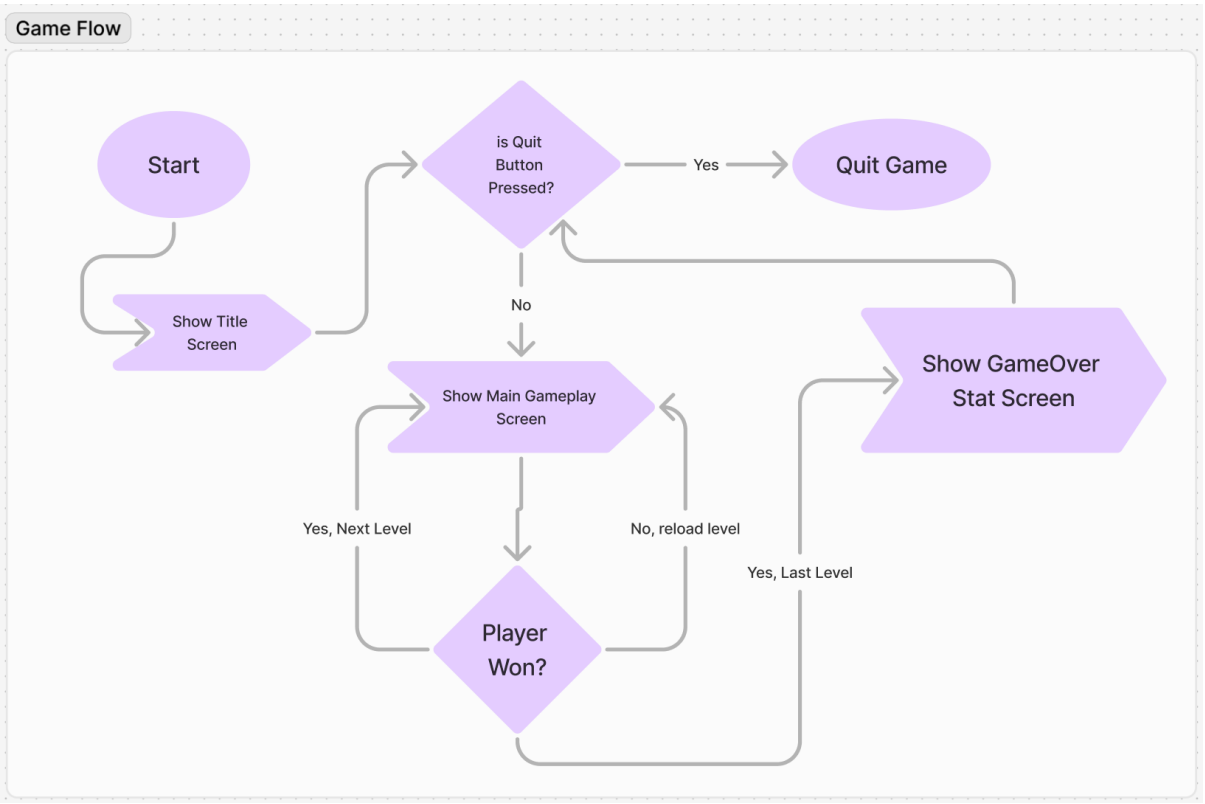
When the win condition is met, the next scene will trigger. If the level is the last one, then the gameover screen will be triggered showing the player's time taken and strokes made to complete the game.

There will be an inventory on the left so the player has the option to collect things by hitting the ball over them. This is less of a game mechanic and more a way to include scriptable objects and experiment with Inventory implementation for future projects.

From the gameover screen the player will have the option to replay the level or quit the application.

## UI Diagram






## System Architecture and Code Structure

Game Manager instance will take care of score/stroke number and time keeping, as well as functions for level changes and quitting the game that will be connected to UI buttons.

The UI will also be a child of the Game Manager with certain UI elements turned off and on depending on the scene.

Most prefabs will have their own script. Any scripts relating to stats will connect to the Game engine instance for continuous record keeping.

The inventory will use scriptable objects to store item data, as well as a List and Dictionary data structure.

The ball colour change will use inheritance.

## Tasks Breakdown

UI: Alex will deal with the UI scripting and implementation.

Scripting and implementation: Daniel will deal mostly with scripting the ball and club assets, and Alex will take care of connecting them to level elements like pickups and the hole/flagpole.

Asset creation: Daniel will create the ball, club and level assets with probuilder

Level creation: Alex will make some quick levels using the assets created by Daniel.