Golf Hero

By Just Some Randos

Members

John Fraser

Swarnajyoti Datta

Bryan Delgado-Bosso

Michael Chiu

Zixing Gong

Eugenia Zhang

Description

Our software is being designed in Unity for Windows machines. It is programmed in C# and organized using the MVC model. Our game is inspired by mini-golf with an abstract spin. This document contains in depth details of our software design and decisions.

Table of Contents

CRC Cards

ArrowController
- Reflect direction camera is facing   - Keep arrow ahead of the ball

BallCamController
- Camera follows the ball   - Camera move based on mouse position   - Camera zoom with mouse wheel   - Hide objects in way of the camera

CollisionTrigger
Change collision layer of ball to fall in the hole

HitBallBehaviour
- Apply a force to the ball with left mouse click   - Adjust the power applied to the ball with the number keys (TEMPORARY)   - Hide arrow while in motion   - Only allow shooting when stopped

HoleTrigger
- Detect when the ball is in the hole   - Reset ball position (TEMPORARY)

MinimapIconBehaviour
Keep the 2D minimap in place

Classes added for Sprint 2

CloudMovement
● Move clouds based on wind direction

PMMController
● Ingame menu ● Pause the game ● Resume the game ● Reset the current stage

WindText
● Display wind direction and speed on the HUD

BallSpawner (*Multiplayer*)
● Spawn player ball ● Keep track of all balls in play ● Keep track of spawn point

PlayerController (*Multiplayer*)
● Perform server setup ● Allow local player input ● Request ball spawner ● Request player shot ● Request reset of ball ● Hide arrow from nonlocal players

Software Architecture Diagram

Expected system(s)
* Our project is expected to run within a desktop environment (e.g. Windows), where the user is able to provide input using their keyboard and mouse and output is able to be sent and displayed on their screen(s) attached to their computer.

Software Architecture Diagram
[![Software Architecture Diagram](http://i.imgur.com/XOgFYjw.png)](http://i.imgur.com/XOgFYjw.png)
* Using Unity, our project's architecture is made up of mostly components.
* The view (which the player sees) is the current scene (menu, level, etc.) that they are on.
* The model (which 'model' items found in golf - thank you Professor Dema for the clarification) are the game objects/prefabs. This includes (and is not limited to) the cameras objects, the player object, as well as the objects making up the level.
* The controller are the scripts which are part of the game objects within the model.

Dealing with errors & exceptions
* Unity's game engine will handle most errors/exceptions which occur during the run-time of the game.
* For user errors (e.g. User providing a display name which contains invalid characters), an in-game dialogue will appear informing the player of the error and how they should fix it.