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

CameraController
• 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

PlayerController
• Apply a force to the ball with left mouse click • Adjust the power applied to the ball based on timing • Hide arrow while in motion • Only allow shooting when stopped

BottomTrigger
• Detect when the ball is in the hole • Load the next scene

MinimapIconBehaviour
• Keep the 2D minimap in place

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
• Spawn player ball • Keep track of all balls in play • Keep track of spawn point

PlayerManager
• Perform server setup for Player • Set active Player state • Request Ball for Player • Request Ball shot • Request Ball reset • Sync variables

BallManager
• Associates a PlayerManager to a Ball • Gets the PlayerManager associated to a Ball

BallsManager
• Spawns a Ball for each player (with start position) • Resets a Ball for each player (to start position) • Updates the start position (spawn point) for each player

GameManager
• Handles disconnecting clients • Handles connecting clients • Handles switching levels for clients • Handles scoring for clients

WindManager
• Gets the current Wind information

Software Architecture

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.

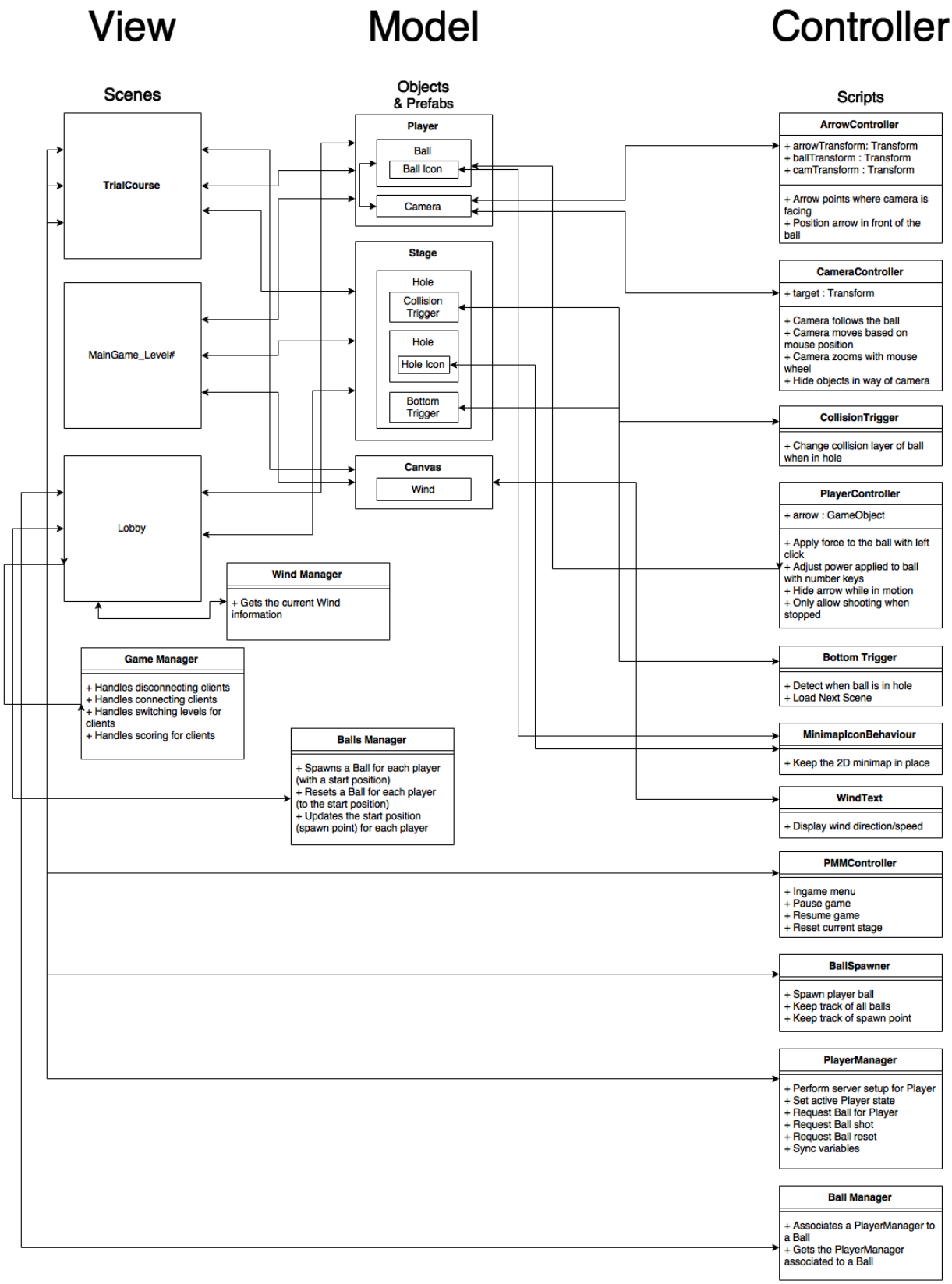
Dealing with Errors & Exceptions

- 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.

Diagram Information

- 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 are the game objects/prefabs. This includes all objects found in either of the scenes.
- The controller are the scripts which are part of the game objects within the model.
- The only models shown are those that are associated with a script in order to show the design process behind each class and the reason for their connections

Diagram


View
Model
Controller
Scenes
Objects & Prefabs
Scripts
TrialCourse
MainGame_Level#
Lobby
Player
Ball
Ball Icon
Camera
Stage
Hole
Collision Trigger
Hole
Hole Icon
Bottom Trigger
Canvas
Wind
Wind Manager
+ Gets the current Wind information
Game Manager
+ Handles disconnecting clients
+ Handles connecting clients
+ Handles switching levels for clients
+ Handles scoring for clients
Balls Manager
+ Spawns a Ball for each player (with a start position)
+ Resets a Ball for each player (to the start position)
+ Updates the start position (spawn point) for each player
ArrowController
+ arrowTransform: Transform
+ ballTransform : Transform
+ camTransform : Transform
+ Arrow points where camera is facing
+ Position arrow in front of the ball
CameraController
+ target : Transform
+ Camera follows the ball
+ Camera moves based on mouse position
+ Camera zooms with mouse wheel
+ Hide objects in way of camera
CollisionTrigger
+ Change collision layer of ball when in hole
PlayerController
+ arrow : GameObject
+ Apply force to the ball with left click
+ Adjust power applied to ball with number keys
+ Hide arrow while in motion
+ Only allow shooting when stopped
Bottom Trigger
+ Detect when ball is in hole
+ Load Next Scene
MinimapIconBehaviour
+ Keep the 2D minimap in place
WindText
+ Display wind direction/speed
PMMController
+ Ingame menu
+ Pause game
+ Resume game
+ Reset current stage
BallSpawner
+ Spawn player ball
+ Keep track of all balls
+ Keep track of spawn point
PlayerManager
+ Perform server setup for Player
+ Set active Player state
+ Request Ball for Player
+ Request Ball shot
+ Request Ball reset
+ Sync variables
Ball Manager
+ Associates a PlayerManager to a Ball
+ Gets the PlayerManager associated to a Ball