

City Pong

Menu (Scene) -

2 Buttons: "Start Game", "Leader Board"

Unity UI Buttons running Functions from UIButtons.cs

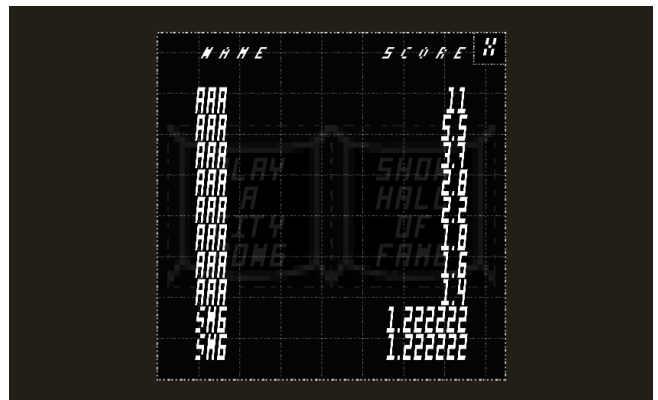
"Start Game" loads the Game Scene.

“Leader Board” shows an overlay panel with a “Close” Button.

The panel has a title and a content area that is populated by **Scores.cs**

Scores.cs looks for a .txt file in PersistentData and loads the record data into newly instantiated prefabs that contain 2 Text objects.

If that text file does not exist, records are loaded from a Default.txt file that is in the Resources folder, loaded as a TextAsset.



Game (Scene) -

Game Scene has 2 paddles, a ball, a scoreboard, a menu Button, and some boundaries, visible and invisible.

"Menu" shows a panel with 3 Buttons:
"Restart Game", "Quit", "Close"

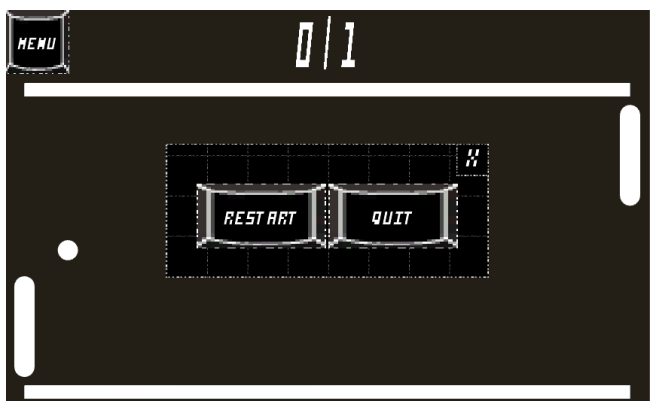
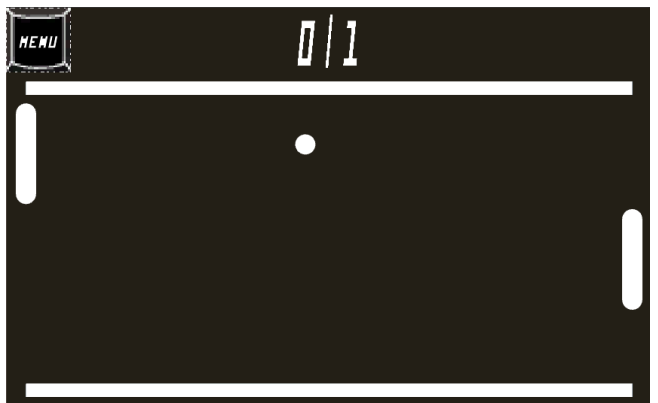
Unity UI Buttons running Functions from UIButtons.cs

“Menu” also stops game time, effectively pausing the game.

"Restart Game" loads this Game Scene.

“Quit” loads the Menu Scene.

“Close” closes the Menu panel.



GameManager.cs manages the game loop and game event related info UI.

It keeps score, and shows info about goals, fouls, victory and defeat.

It spawns the ball after goals, fouls, and at the start of the game.

InputController.cs manages the player's input.

I chose to use the mouse as the controller as it is the closest representation of the original pong analogue input.



I put the player's paddle on the right, as the mouse hand is on the right, 9 times out of 10.

The mouse vertical axis translates the player's paddle within a clamped range.

AI.cs controls the opponent's paddle.

The AI follows the height of the ball, if there's one in play, and if it's moving towards their goal.

The AI has a simple error margin that modifies the accuracy of paddle's move to intercept.

Ball.cs controls the ball's interaction with other colliders and triggers.

The ball has a Rigidbody component, constrained to restrict translation on the y axis.

Collision with a goal trigger, adds score in **GameManager.cs**



Collision with paddles and the top and bottom bumpers, adds velocity to the ball.

If the ball stops moving, it is disabled and reset.

If the ball velocity drops down below a threshold, velocity is added.

If the ball leaves the play area without passing through a goal trigger, a foul is declared, no score assigned, and the ball is reset.

Physics was probably the wrong choice, without clamping the velocity.

When the AI player reaches 11 points, the game is declared over, a message is displayed, and the Leader Board is displayed.

The "Close" and the "Menu" Buttons can be clicked to return to the Menu Scene.

If the player reaches 11 points, everything as mentioned above is true, however an InputField is also enabled.

Scores.cs compares the players score against the opponents score, and that ratio is used to determine their high score.

The high score is compared with the scores in the LeaderBoard and a new record is inserted at that position.

The InputField allows the player to input 3 characters.

On submission, the input is updated in the Leader Board, and the records are saved to a .txt file in PersistentData.

SphereLink.cs and **CylinderShape.cs** were designed to facilitate paddle stretch and squish features that were never implemented.

SphereLink.cs still keeps the two spheres at the ends of the cylinder that is the body of the paddle, but it's purely cosmetic as the collider is a capsule on the cylinder.

