Microgame #1 Ping Pang Game

Name: Alan L. Perez

id: 004862867

Unity File: https://github.com/alanlperez/cse-4410

1. Create a new project, PingPang
2. Create folders under Assets, Scripts, Sprites, Prefabs
3. Create 4 wall game objects, attach collider 2D to Walls
4. Create LeftPaddle game objects, attach box collider 2D and rigidbody 2DGraphical user interface

   Description automatically generated with medium confidence
5. Create PaddleController.cs script file, attach script file to LeftPaddle
6. Save LeftPaddle as a prefab
7. Create RightPaddle by dragging LeftPaddle to Scene, set position
8. PaddleController.cs

Text

Description automatically generated

LeftPaddle: check leftPlayer

RightPaddle: uncheck leftPlayer

1. Create ball game object, attach circle collider 2D and rigidbody 2D componentsGraphical user interface, text, application

   Description automatically generatedGraphical user interface, application

   Description automatically generated
2. Create Bouncy physics material 2, friction is 0, bounciness is 1
3. Create BallController.cs, attach to ball game object

Text

Description automatically generated

1. Drag and drop Bouncy asset into material in circle collider 2DGraphical user interface, application

   Description automatically generated
2. Next implement if the ball collides a paddle, the ball’s velocity will change in the y direction, but stay the same in the x direction
3. If the ball collides with a wall, the player on the opposite end will receive a point and the ball will reset to the center of the screenGraphical user interface, text

   Description automatically generated
4. In the prefab for leftpaddle, assign the tag playerGraphical user interface, application

   Description automatically generated

This will then tag the GameObjects LeftPaddle and RightPaddle as playersGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

1. For the two walls on the y axis behind the paddle will be checked as triggered, and create the function OnTriggerEnter2D in order for the game to know when a player has scored a point.

Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generatedText

Description automatically generated

1. Next we create text in order to keep score. We position it at the top center. Assign font size, and pick the color of the text.A screenshot of a computer

   Description automatically generated with medium confidence
2. Then we update the score in the On trigger Enter 2D function in if/else statementsText

   Description automatically generated
3. Create an empty GameObject named GameContoller, and create a GameController.cs. We then attach the script file to the GameObject. This script file keeps track of the score, increments the score when a player has received a point, and checks if the maximum score has been reached in order to determine whether or not the game is over.

A screenshot of a computer

Description automatically generated with medium confidenceText

Description automatically generated

Graphical user interface

Description automatically generated

1. We reference the GameController.cs in the BallController.cs to check which player scored.

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

1. Then we drag and connect the GameObject text into the GameController script in Score Text.

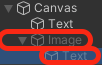
Graphical user interface, application

Description automatically generated

1. Set Max Score to 3 in GameContoller GameObject, it will check if the max score has been reached in the GameController.cs. It will also display and change the score.Graphical user interface, application

   Description automatically generatedText

   Description automatically generated
2. We then create the Game Over text under the Image GameObject which is under the Canvas GameObject.



A screenshot of a computer

Description automatically generated with medium confidenceGraphical user interface

Description automatically generated

1. Then we create the GameOver function in GameController.cs which checks if the game is over, activates the game over screen, and freezes the game.
2. Then in the Update function in GameController.cs if the user presses any key the game will restart. We create a restart function which resets the game to its initial parameters and unfreezes the game.



Text

Description automatically generated

1. We then uncheck image in the Image GameObject, and it will be activated when the game is over.

Graphical user interface, application

Description automatically generated