Microgame #2 Breakout Game

Name: Alan L. Perez

id: 004862867  
Unity File: https://github.com/alanlperez/HW2Breakout

1. Create a new 2D project, HW2Breakout
2. Rename scene to “level01”

A white logo with a black background

Description automatically generated with low confidence

1. Create folders under Assets, Scripts, Sprites

A black screen with white text

Description automatically generated with low confidence

1. Create 4 wall game objects, change color, attach collider 2D to walls.
2. Create Player game object, attach box collider and rigidbody2D. Graphical user interface

   Description automatically generated
3. Freeze position of Y and Freeze Rotation of Z for Player game object.Text

   Description automatically generated with medium confidence
4. Create PlayerController.cs script file, attach script file to Player game object
5. PlayerController.cs

Text

Description automatically generated

1. Set the speed for the Player game object to 1000

Graphical user interface, application

Description automatically generated

1. Create ball game object, change size of ball, and attach circle collider 2D and rigidbody2D components

Graphical user interface

Description automatically generated

1. Create Bouncy physics material 2D, friction is 0, bounciness is 1
2. Drag and drop Bouncy asset into material in circle collider 2D. Change mass to 0.5, and make gravity zero. Freeze rotation.

Graphical user interface, application

Description automatically generatedGraphical user interface, text

Description automatically generated

1. Create BallController.cs, and attach to Ball game object. In the script file create Pushball function to move the ball since there is no gravitational force on the ball. Text

   Description automatically generated
2. Set the speed and the randomUp for the ball.Graphical user interface, application

   Description automatically generated
3. Create the function OnCollisionEnter2D in BallController.cs to change the velocity of the ball when it comes into contact with the player.

Text

Description automatically generated

1. Create function OnTriggerEnter2D, to create a trigger event that will place the ball back into the start position if the goal is met.

Text

Description automatically generated

Text

Description automatically generated

1. Create GameObject Brick, attach Box Collider 2D and tag it as Brick.Graphical user interface

   Description automatically generated
2. Drag the Brick game object into the prefab folder

Graphical user interface

Description automatically generated

1. Use the brick prefab to insert multiple bricks game objects into the scene, highlight a row of brick game object and “control + d” to create multiple rows.
2. Go back into BallController.cs to create a conditional statement in OnCollisionEnter2D function to destroy bricks when ball comes into contact. Text

   Description automatically generated
3. Create LiveText game object and BricksText game object to display how many lives and bricks are left.

Graphical user interface, text

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

1. Create GameController game object and attach the GameController.cs to it.

A screenshot of a computer

Description automatically generated with medium confidence

1. Create the GameController.cs. Here we will change the text of the lives in LoseLive() and bricks in HitBrick(). Pause the game while checking if the game is over in GameOver(), and allow you to restart the game by pressing any key in update(). It will also take you to the next level in NextLevel(). It will also restart the game in Restart().

Text

Description automatically generatedText

Description automatically generated

1. Create Game Object GameOverUI text to have a Game Over Screen and storing the game object into the GameContoller game object under the GameController.cs.A screenshot of a computer

   Description automatically generated with medium confidence
2. Disable GameOverUI game object so that it doesn’t execute until its activated in the GameOver() in GameController.cs.



1. Save Player and Ball as Prefabs

Graphical user interface

Description automatically generated with medium confidence

1. Next go to scenes, click on “level01” and press “control + d” to duplicated it. Rename it “level02” change the speed and Random Up of the ball, and Rearrange the bricks to create a level 2.