Breakout Progress Report – Microgame #2

Nathan Grove

006254920

1. Rename the initial scene to Scene01 in the Scenes folder
2. Create walls and floor sprites as squares
3. Add a Box Collider 2D to all the walls and the floor
4. Create another square sprite for the player
5. Add a Box Collider 2D and a Rigidbody 2D to the player
6. On the Rigidbody 2D we are going to freeze the Y and Z axis, and change the Gravity Scale to 0, Linear Drag to 5
7. Create a PlayerController.cs script and drag that on to the Player object

PlayerController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerController : MonoBehaviour

{

Rigidbody2D playerRigidbody;

float input;

public float speed;

// Start is called before the first frame update

void Start()

{

playerRigidbody = GetComponent<Rigidbody2D>();

}

// Update is called once per frame

void Update()

{

input = Input.GetAxis("Horizontal");

playerRigidbody.AddForce(Vector2.right \* input \* speed \* Time.deltaTime);

}

}

1. Create a Circle sprite for the ball
2. Add a Rigidbody2D and a Circle Collider 2D to the ball
3. Create a Physics material called Bouncy and set the Bounciness to 1, and Friction to 0
4. Drag that Bouncy Physics material to the Ball object
5. On the Rigidbody2D we want to freeze the Z axis and make the mass 0.5 and gravity scale 0
6. Create a BallController.cs script and drag that on to the Ball object

BallController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class BallController : MonoBehaviour

{

Rigidbody2D ballRigidbody;

public float speed;

public float randomUp;

Vector3 startPosition;

// Start is called before the first frame update

void Start()

{

ballRigidbody = GetComponent<Rigidbody2D>();

startPosition = transform.position;

}

private void OnEnable()

{

Invoke("PushBall", 1f);

}

void PushBall()

{

int dir = Random.Range(0, 2);

float x;

if (dir == 0)

x = speed;

else

x = -speed;

ballRigidbody.AddForce(new Vector2(x, -randomUp));

}

// Update is called once per frame

void Update()

{

}

private void OnCollisionEnter2D(Collision2D collision)

{

if(collision.gameObject.CompareTag("Player"))

{

Vector2 vel;

vel.y = ballRigidbody.velocity.y;

vel.x = ballRigidbody.velocity.x / 2 + collision.collider.attachedRigidbody.velocity.x / 2;

ballRigidbody.velocity = vel;

}

}

private void OnTriggerEnter2D(Collider2D collision)

{

if(collision.gameObject.CompareTag("Goal"))

{

ballRigidbody.velocity = Vector2.zero;

transform.position = startPosition;

Invoke("PushBall", 2f);

}

}

}

1. Add Player tag to Player object
2. Add Goal tag to the Floor object
3. Create another square sprite for a Brick
4. Add a Box Collider 2D to the Brick and tag the Brick as Brick
5. Add to BallController.cs under OnCollisionEnter2D: if(collision.gameObject.CompareTag("Brick"))

{

Destroy(collision.gameObject);

}

1. Add a UI Text object to keep track of lives and bricks
2. Create an empty game object and a GameController.cs script and attach it to the empty game object

GameController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SceneManagement;

public class GameController : MonoBehaviour

{

public int lives = 3;

int numOfBricks;

[SerializeField]

private Text LivesText;

[SerializeField]

private Text BricksText;

public GameObject gameOverUI;

bool gameOver = false;

// Start is called before the first frame update

void Awake()

{

LivesText.text = "Lives: " + lives.ToString();

numOfBricks = GameObject.FindGameObjectsWithTag("Brick").Length;

BricksText.text = "Bricks: " + numOfBricks.ToString();

}

// Update is called once per frame

void Update()

{

if(gameOver && Input.anyKeyDown)

{

Restart();

}

}

public void LoseLive()

{

lives--;

LivesText.text = "Lives: " + lives.ToString();

if(lives <= 0)

{

GameOver();

}

}

void GameOver()

{

gameOver = true;

gameOverUI.SetActive(true);

Time.timeScale = 0f;

}

public void HitBrick()

{

numOfBricks--;

BricksText.text = "Bricks: " + numOfBricks.ToString();

if(numOfBricks <= 0)

{

Invoke("NextLevel", 2f);

}

}

void NextLevel()

{

Time.timeScale = 1f;

SceneManager.LoadScene("Scene02");

}

void Restart()

{

Time.timeScale = 1f;

SceneManager.LoadScene("Scene01");

}

}

Add an Image object to begin the game over screen process

1. Add game over text and drag that on top of the image object
2. Add these pieces to BallController.cs

At the top of the file:

GameController cont;

Then when the ball collides with the Goal:

cont.LoseLive();

And when the ball collides with a Brick:’

cont.HitBrick();

1. Ctrl+D Scene01 and rename it to Scene02
2. Add more bricks and move all of them around
3. Boom, you done.