Shump Progress Report – Microgame #3

Nathan Grove

006254920

1. Create new project and create Animations, Prefabs, Scripts, and Sprites folders
2. Add the sprites used into the Sprites folder
3. Click Tiles sprite, change sprite mode to multiple and go into the sprite editor to select the top left corner of 32x32 pixels and click apply
4. Create a Tilemap, go into the Tile Palette and Create a new palette with our Tiles\_Grass sprite
5. Change the PPI in the Inspector of the Tiles sprite to 32, and go back into the Tile Palette and click on the box with an arrow in it, and click and drag onto the Tile Map to make the background
6. Drag the player sprite on to the Hierarchy
7. Change the Order in Layer of the background Tile Map to -1, and make sure that the Player’s Order in Layer is 0
8. Change Player size to x2
9. Add a Box Collider 2D and Rigidbody 2D to the Player and change the Mass and Linear Drag to 5 and freeze the Z-axis. In addition to this make the Box Collider smaller
10. Create a PlayerController script and attach it to the Player Game Object
11. Create 2 Empty Game Objects and place them at the sides of the player for our bullet spawn positions
12. Add the bullet sprite to the Hierarchy
13. Add the Circle collider 2D and Rigidbody2D to the bullet, setting the Linear Drag to 0 and gravity to zero as well as freezing the z axis
14. Create a BulletController script and attach it to the Bullet Game object
15. Drag the bullet into the prefabs folder
16. Drag the bullet prefab to our Player object, and drag the bullet spawn positions into the correct fields
17. Drag the 2 enemy sprites to the Hierarchy panel
18. Add and adjust box collider 2D and the Rigidbody2D with a linear drag and mass of 5, freezing the z axis
19. Create an EnemyController script and drag to both enemies
20. Add both enemies to Prefabs
21. Add a bullet prefab to the Hierarchy and rename it to EnemyBullet
22. Unpack the EnemyBullet completely and change the sprite to the correct one
23. Create an EnemyBulletController script and remove the BulletController from EnemyBullet game object and attach the script we just made
24. Add EnemyBullet as prefab
25. Create an empty game object and a script called GameController and attach the script to the empty game object
26. Drag the enemy prefabs to the enemies array field in the Inspector
27. Add the player tag to the player game object
28. Add the enemy tag to the enemy prefabs
29. Add a canvas for the health bar
30. Create an empty game object under the canvas and add a rect transform and a horizontal layout group to it
31. Add an image under the empty game object and drag the heart image to the Source Image
32. Open the explosion sprite in the Sprite Editor
33. 185x185 boxes, one for each stage of the explosion
34. Click and select the 7 sprites we blocked out in the above step in the assets panel and drag it to Hierarchy to save it as an animation in the animations folder
35. Create an ExplosionController and drag it on top of the explosion game object you just made
36. Add the flash sprites into the Assets folder
37. Click the first sprite of the flash animation, and then the last and drag that into the Hierarchy
38. Drag the ExplosionController script onto it and set the clip as the flash animation you just made
39. Drag the flash animation game object to the prefabs folder and drag that to the flash attribute in the inspector on the player game object

PlayerController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerController : MonoBehaviour

{

[Header("Starting States")]

public float speed;

Vector2 input;

Rigidbody2D playerRigidbody;

[Header("Shooting")]

public GameObject bullet;

public GameObject[] bulletSpawnPositions;

private float cools;

public float timeBetweenShots;

public GameObject flash;

[Header("Health")]

public int maxHealth = 10;

public int health;

public GameObject healthImage;

public GameObject healthParent;

public float timeBetweenHurts = 0.3f;

float iframe;

// Start is called before the first frame update

void Start()

{

playerRigidbody = GetComponent<Rigidbody2D>();

cools = timeBetweenShots;

health = maxHealth;

iframe = timeBetweenHurts;

for (int i = 0; i < health - 1; i++)

AddHeart();

}

void AddHeart()

{

GameObject heart = Instantiate(healthImage);

heart.transform.SetParent(healthParent.transform);

}

void RemoveHeart(int n)

{

if (healthParent.transform.childCount > 0)

{

if (healthParent.transform.childCount < n)

n = healthParent.transform.childCount;

for(int i = 0; i < n; i++)

{

Destroy(healthParent.transform.GetChild(0).gameObject);

}

}

}

// Update is called once per frame

void Update()

{

input = new Vector2(Input.GetAxis("Horizontal"), Input.GetAxis("Vertical"));

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

if (Input.GetKey(KeyCode.Space) && cools <= 0)

Shoot();

if (cools > 0)

cools -= Time.deltaTime;

if (iframe > 0)

iframe -= Time.deltaTime;

}

void Shoot()

{

for(int i = 0; i < bulletSpawnPositions.Length; i++)

{

Instantiate(bullet, bulletSpawnPositions[i].transform.position, Quaternion.identity);

}

Instantiate(flash, transform.position, Quaternion.identity);

cools = timeBetweenShots;

}

public void TakeDamage(int damage)

{

if(iframe <= 0)

{

health = health - damage;

RemoveHeart(damage);

if (health <= 0)

GameOver();

iframe = timeBetweenHurts;

}

}

void GameOver()

{

FindObjectOfType<GameController>().gameOver = true;

FindObjectOfType<GameController>().gameOverUI.SetActive(true);

gameObject.SetActive(false);

Time.timeScale = 0f;

}

void OnCollisionEnter2D(Collision2D collision)

{

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

{

TakeDamage(1);

Destroy(collision.gameObject);

}

}

}

BulletController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class BulletController : MonoBehaviour

{

Rigidbody2D bulletRigidbody;

public float speed;

int damage = 1;

// Start is called before the first frame update

void Awake()

{

bulletRigidbody = GetComponent<Rigidbody2D>();

}

private void OnEnable()

{

bulletRigidbody.AddForce(Vector2.up \* speed);

Invoke("Disable", 5f);

}

private void Disable()

{

Destroy(gameObject);

}

// Update is called once per frame

void Update()

{

}

private void OnTriggerEnter2D(Collider2D collision)

{

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

{

collision.gameObject.GetComponent<EnemyController>().TakeDamage(damage);

Invoke("Disable", 0.001f);

}

}

}

EnemyController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class EnemyController : MonoBehaviour

{

Rigidbody2D enemyRigidbody;

PlayerController player;

public float xSpeed, ySpeed;

public GameObject bullet;

public float timeBetweenAttackLow = 0.5f;

public float timeBetweenAttackHigh = 2f;

float attackCools;

public int maxEnemyHealth;

int enemyHealth;

GameController cont;

public int amount;

Vector2 bounds;

public GameObject explosion;

// Start is called before the first frame update

void Start()

{

enemyRigidbody = GetComponent<Rigidbody2D>();

player = FindObjectOfType<PlayerController>();

cont = FindObjectOfType<GameController>();

attackCools = Random.Range(timeBetweenAttackLow, timeBetweenAttackHigh);

enemyHealth = maxEnemyHealth;

bounds = Camera.main.ScreenToWorldPoint(new Vector3(Screen.width, Screen.height, 0f));

}

// Update is called once per frame

void Update()

{

float x = 0f;

if (player != null)

{

if (player.transform.position.x > transform.position.x) //left

x = xSpeed;

else if (player.transform.position.x < transform.position.x) //right

x = -xSpeed;

enemyRigidbody.AddForce(new Vector2(x, -ySpeed) \* Time.deltaTime);

if (attackCools > 0)

attackCools -= Time.deltaTime;

else

Attack();

if (transform.position.y < -bounds.y)

{

cont.AddScore(-amount);

Destroy(gameObject);

}

}

}

void Attack()

{

Instantiate(bullet, transform.position, transform.rotation);

attackCools = Random.Range(timeBetweenAttackLow, timeBetweenAttackHigh);

}

public void TakeDamage(int damage)

{

enemyHealth -= damage;

if (enemyHealth <= 0)

Die();

}

void Die()

{

cont.AddScore(amount);

Instantiate(explosion, transform.position, Quaternion.Euler(0,0,0));

Destroy(gameObject);

}

}

EnemyBulletController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class EnemyBulletController : MonoBehaviour

{

Rigidbody2D bulletRigidbody;

public float speed;

public int damage = 1;

// Start is called before the first frame update

void Awake()

{

bulletRigidbody = GetComponent<Rigidbody2D>();

}

private void OnEnable()

{

bulletRigidbody.AddForce(Vector2.down \* speed);

Invoke("Disable", 5f);

}

private void Disable()

{

Destroy(gameObject);

}

// Update is called once per frame

void Update()

{

}

private void OnTriggerEnter2D(Collider2D collision)

{

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

{

collision.gameObject.GetComponent<PlayerController>().TakeDamage(damage);

Invoke("Disable", 0.001f);

}

}

}

GameController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SceneManagement;

public class GameController : MonoBehaviour

{

public GameObject[] enemies;

public float timeBetweenSpawnLow = 0.5f;

public float timeBetweenSpawnHigh = 3f;

float spawnCools;

Vector2 bounds;

Vector3 spawnPosition;

public Text scoreText;

int scores;

public bool gameOver;

public GameObject gameOverUI;

// Start is called before the first frame update

void Start()

{

bounds = Camera.main.ScreenToWorldPoint(new Vector3(Screen.width, Screen.height, 0f));

spawnCools = Random.Range(timeBetweenSpawnLow, timeBetweenSpawnHigh);

scores = 0;

scoreText.text = "Score: " + scores.ToString();

gameOver = false;

}

// Update is called once per frame

void Update()

{

if (spawnCools > 0)

spawnCools -= Time.deltaTime;

else

SpawnEnemy();

scoreText.text = "Score: " + scores.ToString();

if(gameOver && Input.anyKeyDown)

{

Restart();

}

}

void SpawnEnemy()

{

spawnPosition = new Vector3(Random.Range(-bounds.x + 1f, bounds.x - 1f), bounds.y + Random.Range(0.25f, 3f), 0f);

Instantiate(enemies[Random.Range(0,enemies.Length)], spawnPosition, Quaternion.Euler(0, 0, 180));

spawnCools = Random.Range(timeBetweenSpawnLow, timeBetweenSpawnHigh);

}

public void AddScore(int amount)

{

scores += amount;

}

void Restart()

{

SceneManager.LoadScene("SampleScene");

Time.timeScale = 1f;

}

}

ExplosionController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class ExplosionController : MonoBehaviour

{

public AnimationClip clip;

private void OnEnable()

{

Invoke("Disable", clip.length);

}

private void Disable()

{

Destroy(gameObject);

}

}