

Learn game development w/ U X

(2) WhatsApp

https://learn.unity.com/tutorial/challenge-4-soccer-scripting?uv=2022.3&courseId=5cd96c41edbc2a2ca6e8810f&pro

Course
Create with Code

Project
Getting Started

Project
Unit 1 - Player Control

Project
Unit 2 - Basic Gameplay

Project
Unit 3 - Sound and Effects

Project
Unit 4 - Gameplay Mechanics

Tutorial • Unit 4 - Introduction

Tutorial • Lesson 4.1 - Watch Where You're Going

Tutorial • Lesson 4.2 - Follow the Player

- When there are no more enemy balls, a new wave spawns with 1 more enemy

Materials

[Challenge 4 - Starter Files.zip](#)

Track your progress and get personalized recommendations.

Sign in with your Unity ID

1. Overview

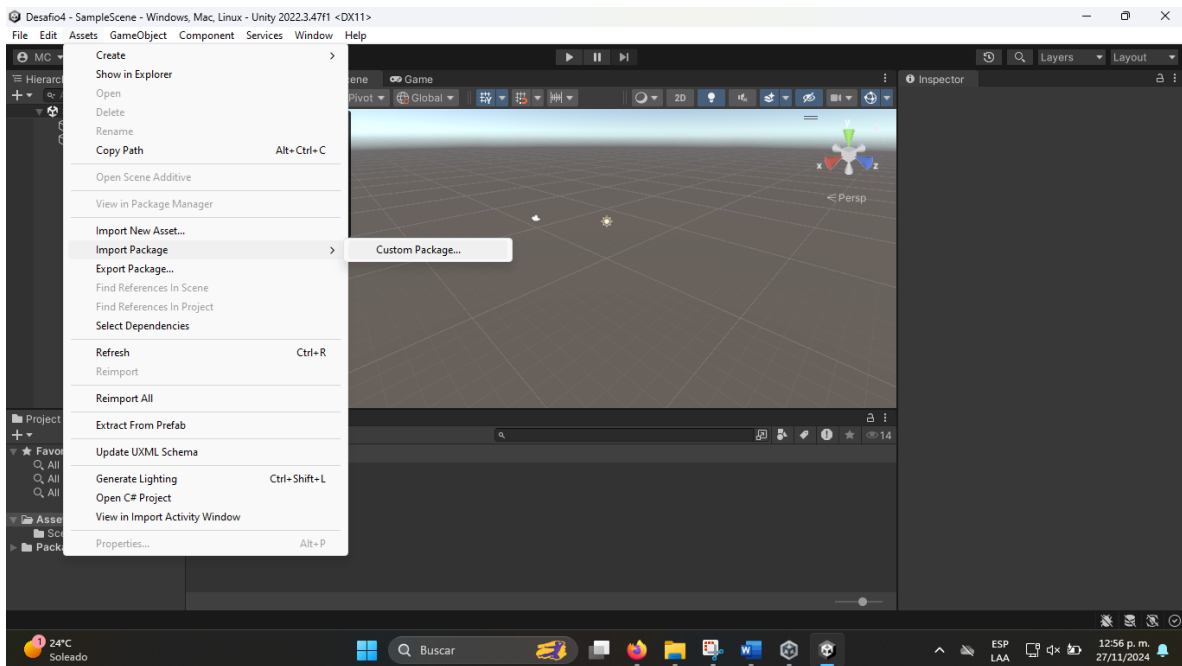
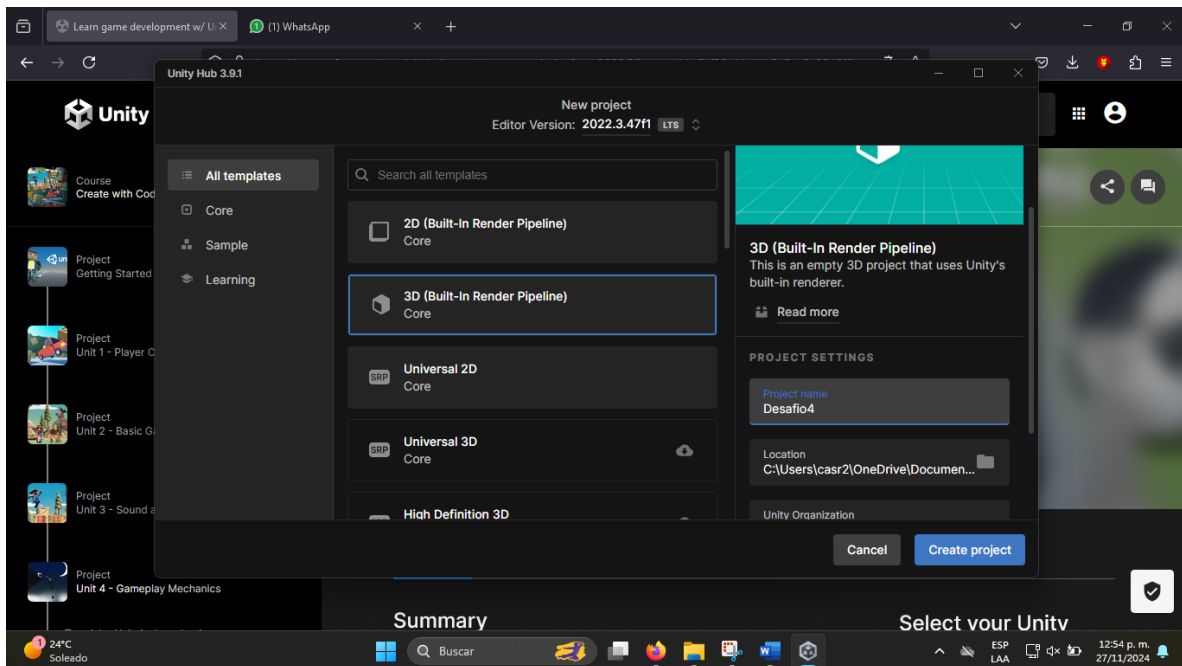
34

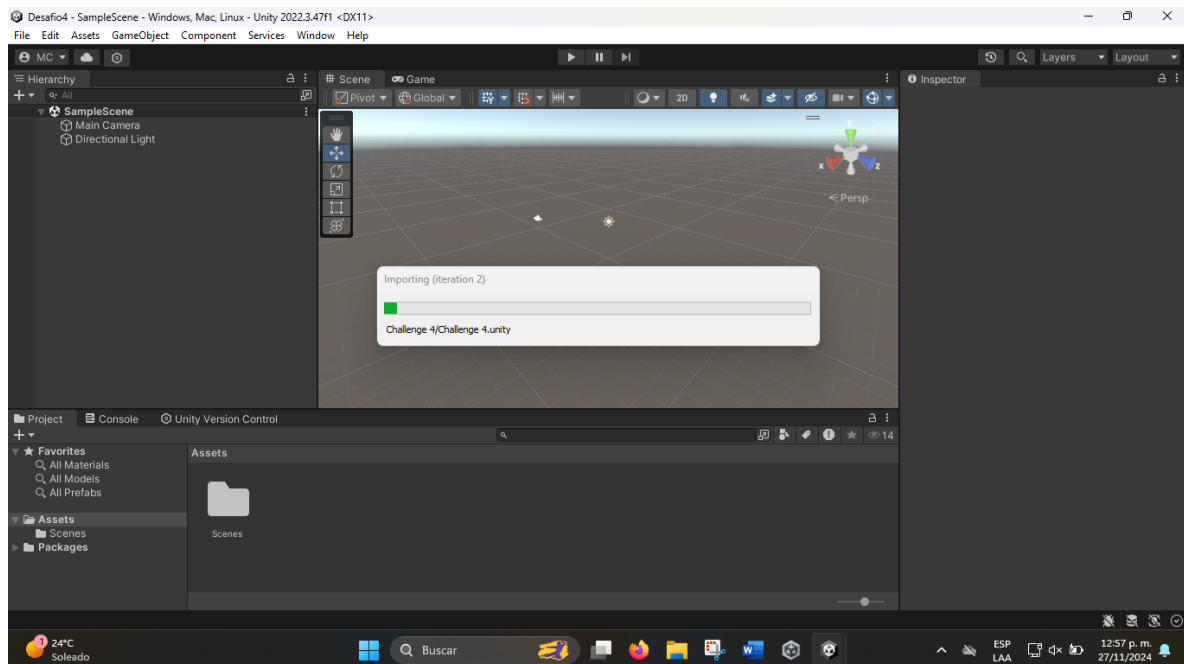
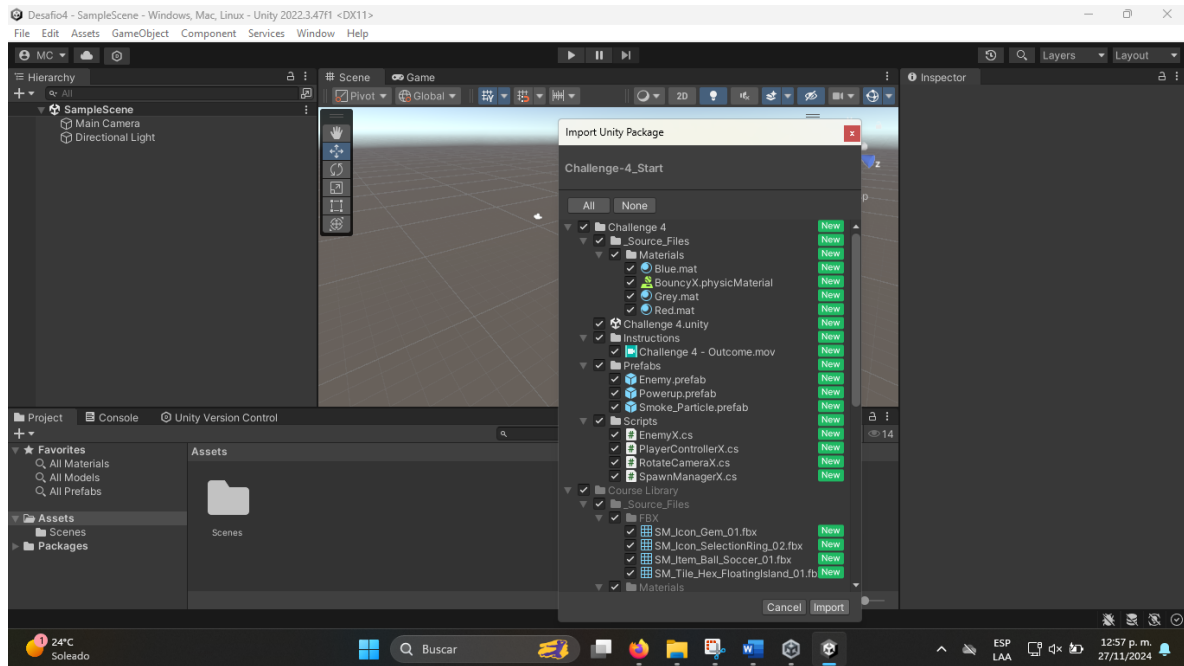
20°C
Soleado

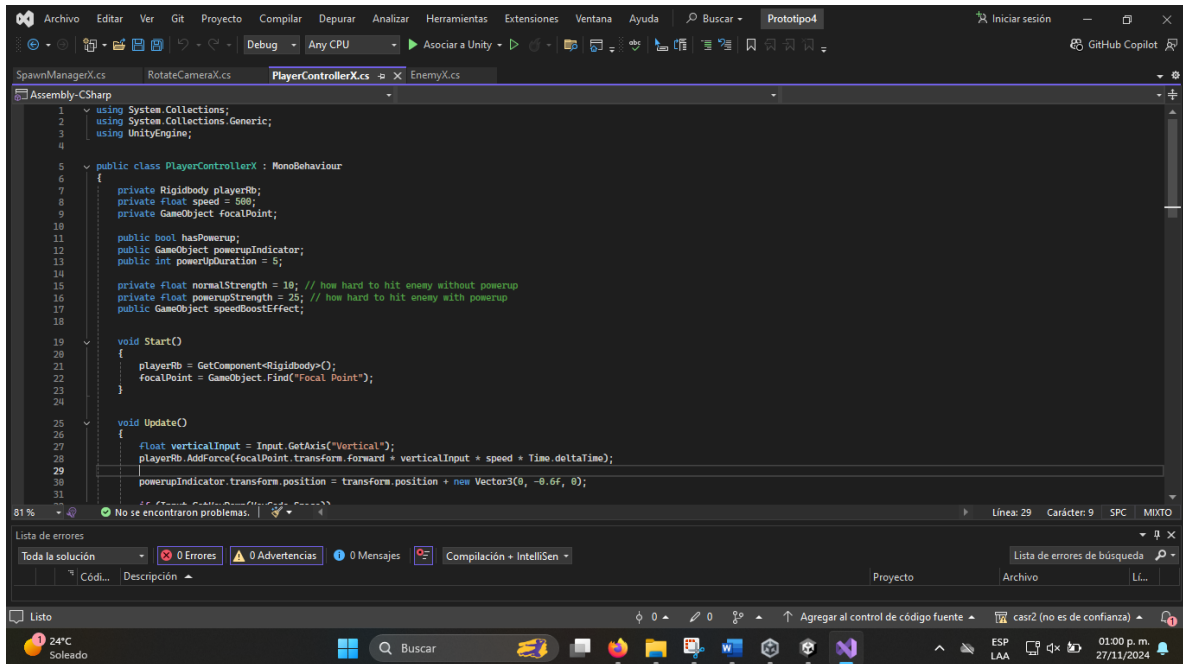
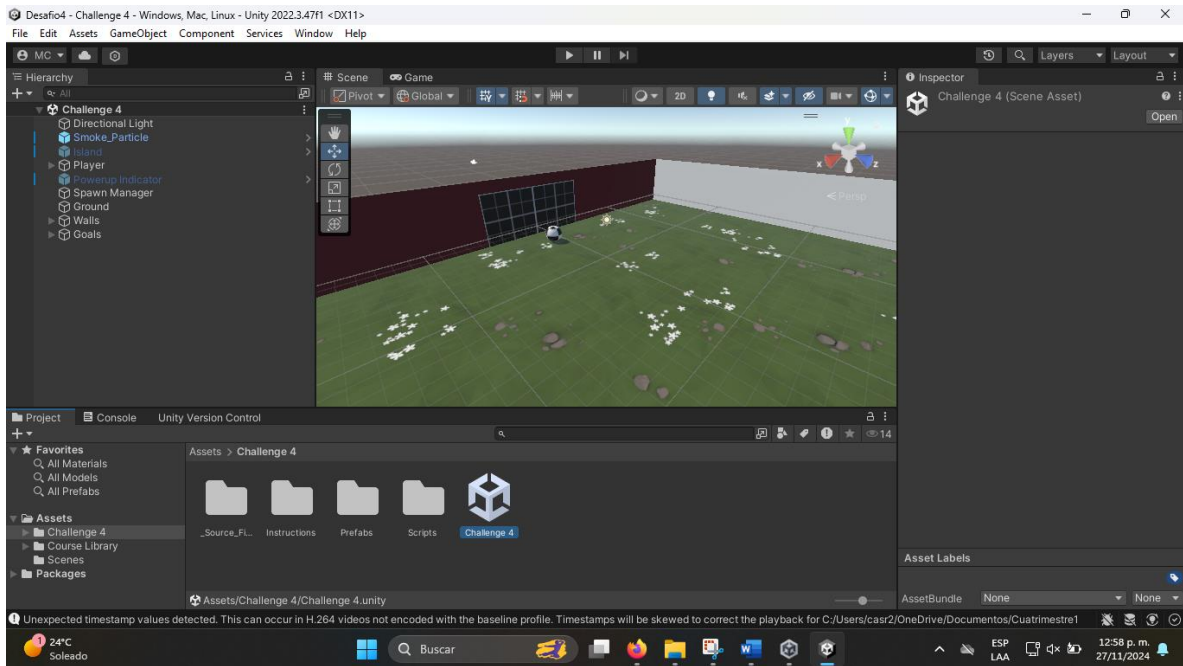
Buscar

ESP LAA

12:26 p. m.
27/11/2024







```
53 IEnumerator PowerUpCooldown()
54 {
55     yield return new WaitForSeconds(powerUpDuration);
56     hasPowerup = false;
57     powerupIndicator.SetActive(false);
58 }
59
60 // If Player collides with enemy
61 #pragma warning disable 0168
62 private void OnCollisionEnter(Collision other)
63 {
64     if (other.gameObject.CompareTag("Enemy"))
65     {
66         Rigidbody enemyRigidbody = other.gameObject.GetComponent<Rigidbody>();
67         Vector3 awayFromPlayer = other.gameObject.transform.position - transform.position;
68
69         if (hasPowerup) // if have powerup hit enemy with powerup force
70         {
71             enemyRigidbody.AddForce(awayFromPlayer * powerupStrength, ForceMode.Impulse);
72         }
73         else // if no powerup, hit enemy with normal strength
74         {
75             enemyRigidbody.AddForce(awayFromPlayer * normalStrength, ForceMode.Impulse);
76         }
77     }
78 }
79
80
81
82
83
84 }
```

81% No se encontraron problemas. Línea: 66 Carácter: 13 SPC MIXTO

Lista de errores: 0 Errores, 0 Advertencias, 0 de 5 Mensajes. Compilación: IntelliSense

```
18 public GameObject player;
19
20 // Update is called once per frame
21 // Message de Unity (0 referencias)
22 void Update()
23 {
24     //int length = GameObject.FindGameObjectsWithTag("Powerup").Length;
25     //enemyCount = length;
26
27     //if (enemyCount != 0)
28     //{
29         return;
30     //}
31     //SpawnEnemyWave(waveCount);
32     enemyCount = GameObject.FindGameObjectsWithTag("Enemy").Length;
33     if (enemyCount == 0)
34     {
35         SpawnEnemyWave(waveCount);
36     }
37 }
38
39
40
41 // Generate random spawn position for powerups and enemy balls
42 // referencia
43 Vector3 GenerateSpawnPosition ()
44 {
45     float xPos = Random.Range(-spawnRangeX, spawnRangeX);
46     float zPos = Random.Range(spawnZMin, spawnZMax);
47     return new Vector3(xPos, 0, zPos);
48 }
49
```

81% No se encontraron problemas. Línea: 32 Carácter: 62 SPC CRLF

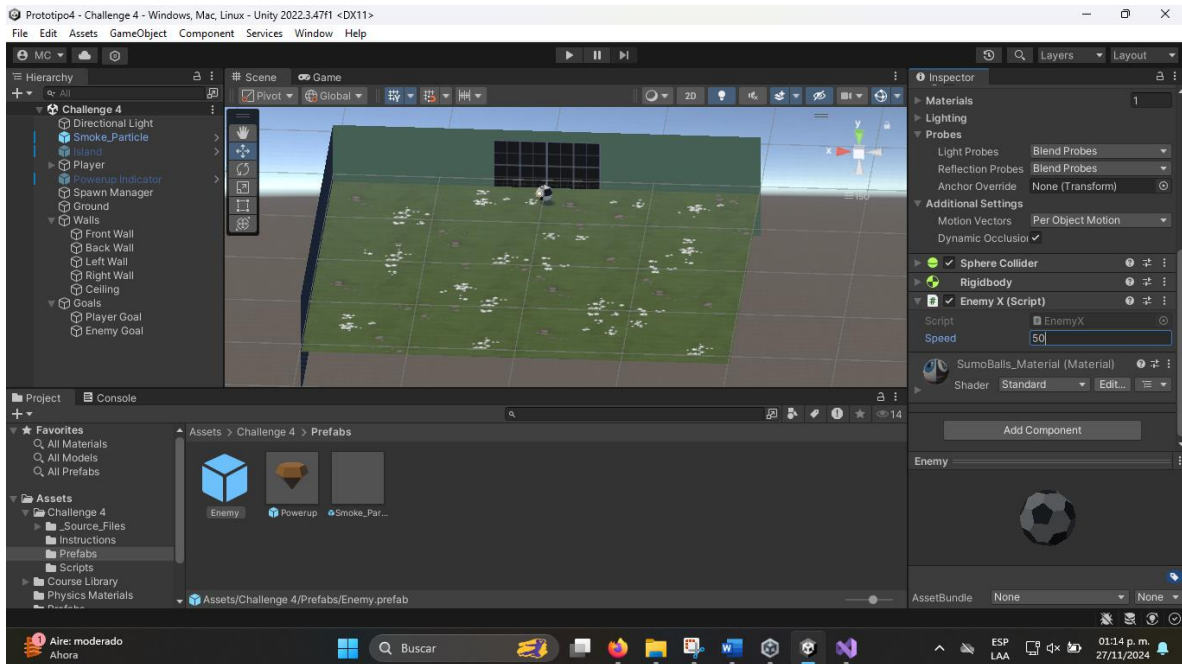
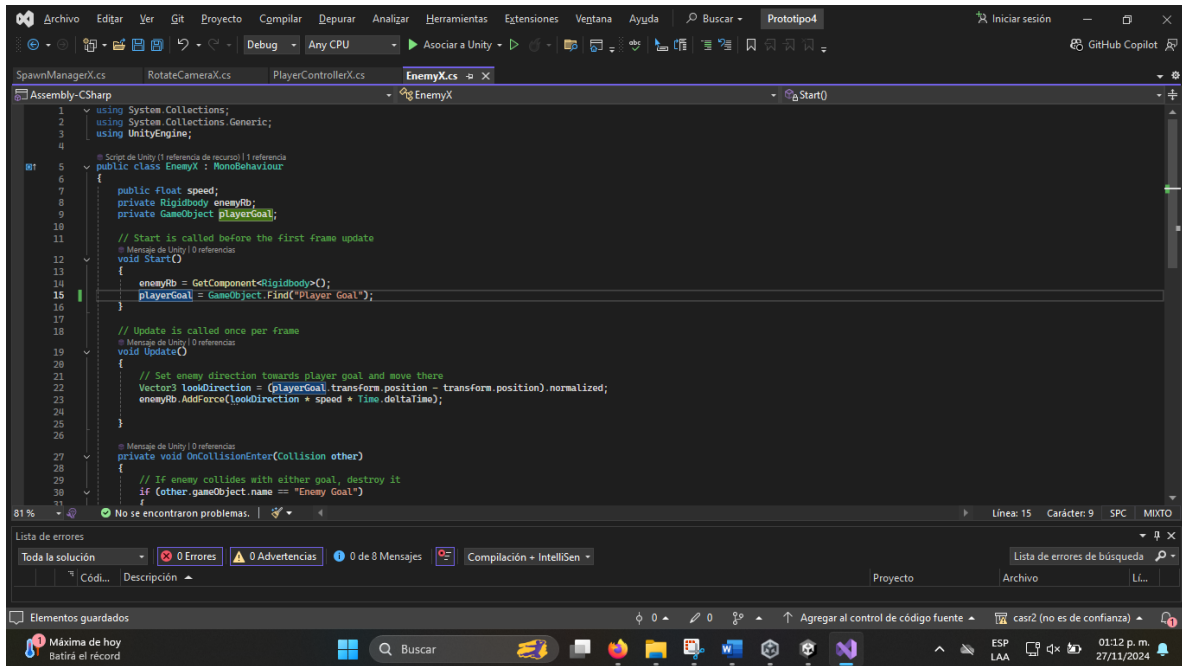
Lista de errores: 0 Errores, 0 Advertencias, 0 de 8 Mensajes. Compilación: IntelliSense

This screenshot shows the Visual Studio Code editor with the file `PlayerControllerX.cs` open. The code implements logic for a player's powerup system. It includes a `OnTriggerEnter` method that checks for collisions with powerups, adds force to the player's movement, and activates a powerup. A coroutine `PowerupCooldown` is used to manage the duration of the powerup. The interface at the bottom shows 0 errors and 0 warnings. The Windows taskbar at the bottom displays the system clock as 01:06 p.m. on 27/11/2024.

```
30 powerupIndicator.transform.position = transform.position + new Vector3(0, -0.6f, 0);
31
32 if (Input.GetKeyDown(KeyCode.Space))
33 {
34     playerRb.AddForce(focalPoint.transform.forward * powerupStrength, ForceMode.Impulse);
35     Instantiate(speedBoostEffect, transform.position, speedBoostEffect.transform.rotation);
36 }
37
38 // If Player collides with powerup, activate powerup
39 // Mensaje de Unity 1.0 referencias
40 private void OnTriggerEnter(Collider other)
41 {
42     if (other.gameObject.CompareTag("Powerup"))
43     {
44         StartCoroutine(PowerupCooldown());
45         Destroy(other.gameObject);
46         hasPowerup = true;
47         powerupIndicator.SetActive(true);
48     }
49
50 // Coroutine to count down powerup duration
51 // 1 referencia
52 IEnumerator PowerupCooldown()
53 {
54     yield return new WaitForSeconds(powerupDuration);
55     hasPowerup = false;
56     powerupIndicator.SetActive(false);
57 }
58
59 // If Player collides with enemy
60 // Mensaje de Unity 1.0 referencias
```

This screenshot shows the Visual Studio Code editor with the file `SpawnManagerX.cs` open. The code implements a wave-based enemy spawning system. The `SpawnEnemyWave` method iterates through a list of enemies to spawn, instantiates them at specific positions, and updates their speed. It also includes a `ResetPlayerPosition` method to move the player back to the start of the level. The interface at the bottom shows 0 errors and 0 warnings. The Windows taskbar at the bottom displays the system clock as 01:08 p.m. on 27/11/2024.

```
50 void SpawnEnemyWave(int enemiesToSpawn)
51 {
52     for (int i = 0; i < enemiesToSpawn; i++)
53     {
54         GameObject enemy = Instantiate(enemyPrefab, GenerateSpawnPosition(), enemyPrefab.transform.rotation);
55         enemy.GetComponent<Enemy>().speed += waveCount * enemySpeedIncrement;
56     }
57     waveCount++;
58     ResetPlayerPosition();
59 }
60
61 // Move player back to position in front of own goal
62 // 1 referencia
63 void ResetPlayerPosition ()
64 {
65     player.transform.position = new Vector3(0, 1, -7);
66     player.GetComponent<Rigidbody>().velocity = Vector3.zero;
67     player.GetComponent<Rigidbody>().angularVelocity = Vector3.zero;
68 }
69
70
71 }
```



Archivo Editar Ver Git Proyecto Compilar Depurar Analizar Herramientas Extensiones Ventana Ayuda Prototipo4 Iniciar sesión

Debug Any CPU Asociar a Unity

SpawnManagerX.cs PlayerControllerX.cs RotateCameraX.cs EnemyX.cs

Assembly-CSharp PlayerControllerX

```
12 public GameObject powerupIndicator;
13 public int powerupDuration = 5;
14
15 private float normalStrength = 10; // how hard to hit enemy without powerup
16 private float powerupStrength = 25; // how hard to hit enemy with powerup
17 float turboBoost = 10f;
18
19 // Mensaje de Unity | 0 referencias
20 void Start()
21 {
22     playerRb = GetComponent<Rigidbody>();
23     focalPoint = GameObject.Find("Focal Point");
24 }
25
26 // Mensaje de Unity | 0 referencias
27 void Update()
28 {
29     float verticalInput = Input.GetAxis("Vertical");
30     playerRb.AddForce(focalPoint.transform.forward * verticalInput * speed * Time.deltaTime);
31
32     powerupIndicator.transform.position = transform.position + new Vector3(0, -0.6f, 0);
33
34     if (Input.GetKeyDown(KeyCode.Space))
35     {
36         playerRb.AddForce(focalPoint.transform.forward * turboBoost, ForceMode.Impulse);
37     }
38
39     // If Player collides with powerup, activate powerup
40     // Mensaje de Unity | 0 referencias
41     private void OnTriggerEnter(Collider other)
42     {
43         if (other.gameObject.CompareTag("Powerup"))
44         {
45             StartCoroutine(PowerupCoolDown());
46         }
47     }
48
49 // 81 % No se encontraron problemas. Línea: 16 Carácter: 78 SPC MIXTO
```

PowerShell para desarrolladores

+ PowerShell para desarrolladores

PS C:\Users\casr2\OneDrive\Documentos\Cuatrimestre10\Juegos\Prototipo4>

PowerShell para desarrolladores Lista de errores

Elementos guardados

26°C Soledad

Buscar

ESP LAA 01:21 p. m. 27/11/2024

Archivo Editar Ver Git Proyecto Compilar Depurar Analizar Herramientas Extensiones Ventana Ayuda Prototipo4 Iniciar sesión

Debug Any CPU Asociar a Unity

SpawnManagerX.cs PlayerControllerX.cs RotateCameraX.cs EnemyX.cs

Assembly-CSharp PlayerControllerX

```
6 private Rigidbody playerRb;
7 private float speed = 500;
8 private GameObject focalPoint;
9
10 public bool hasPowerup;
11 public GameObject powerupIndicator;
12 public int powerupDuration = 5;
13
14 private float normalStrength = 10; // how hard to hit enemy without powerup
15 private float powerupStrength = 25; // how hard to hit enemy with powerup
16 float turboBoost = 10f;
17
18 public ParticleSystem turbosSmoke;
19
20 // Mensaje de Unity | 0 referencias
21 void Start()
22 {
23     playerRb = GetComponent<Rigidbody>();
24     focalPoint = GameObject.Find("Focal Point");
25 }
26
27 // Mensaje de Unity | 0 referencias
28 void Update()
29 {
30     float verticalInput = Input.GetAxis("Vertical");
31     playerRb.AddForce(focalPoint.transform.forward * verticalInput * speed * Time.deltaTime);
32
33     powerupIndicator.transform.position = transform.position + new Vector3(0, -0.6f, 0);
34
35     if (Input.GetKeyDown(KeyCode.Space))
36     {
37         playerRb.AddForce(focalPoint.transform.forward * turboBoost, ForceMode.Impulse);
38         turbosSmoke.Play();
39     }
40
41     // If Player collides with powerup, activate powerup
42     // Mensaje de Unity | 0 referencias
43     private void OnTriggerEnter(Collider other)
44     {
45         if (other.gameObject.CompareTag("Powerup"))
46         {
47             StartCoroutine(PowerupCoolDown());
48         }
49     }
50
51 // 81 % No se encontraron problemas. Línea: 24 Carácter: 53 SPC MIXTO
```

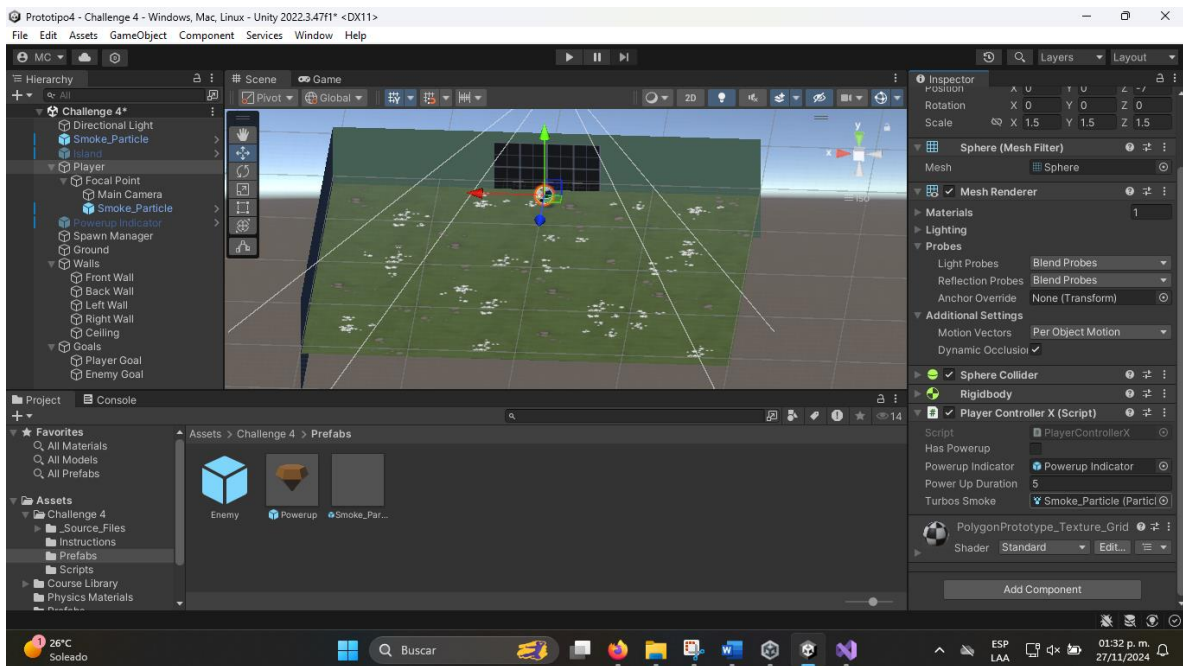
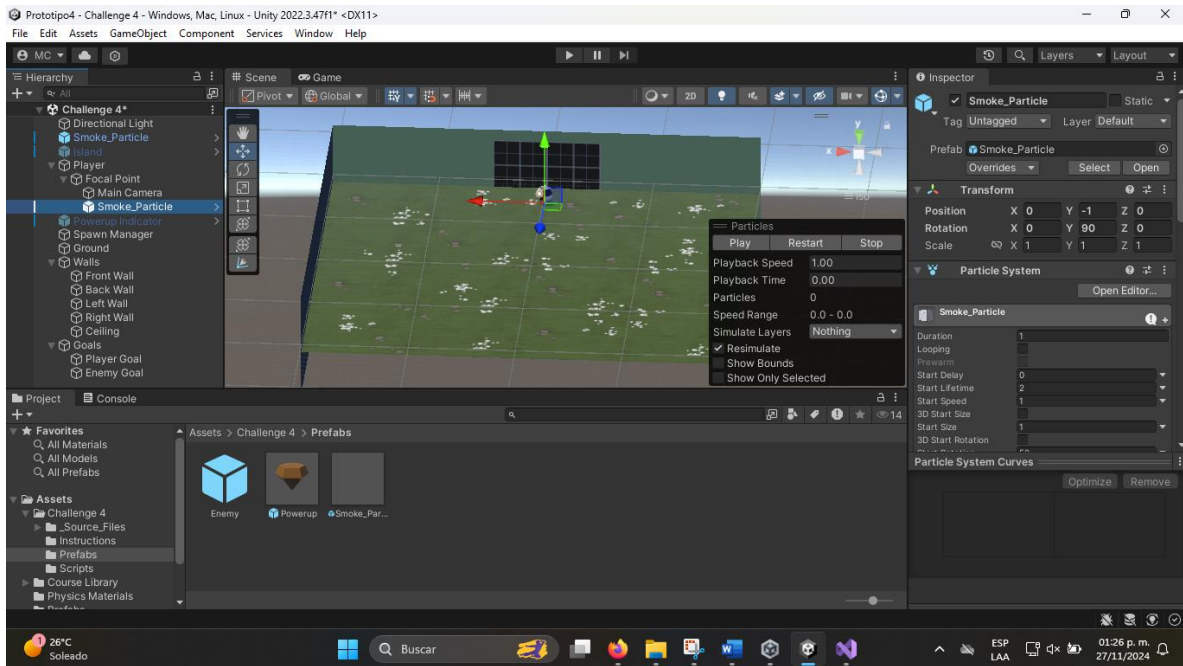
PowerShell para desarrolladores Lista de errores

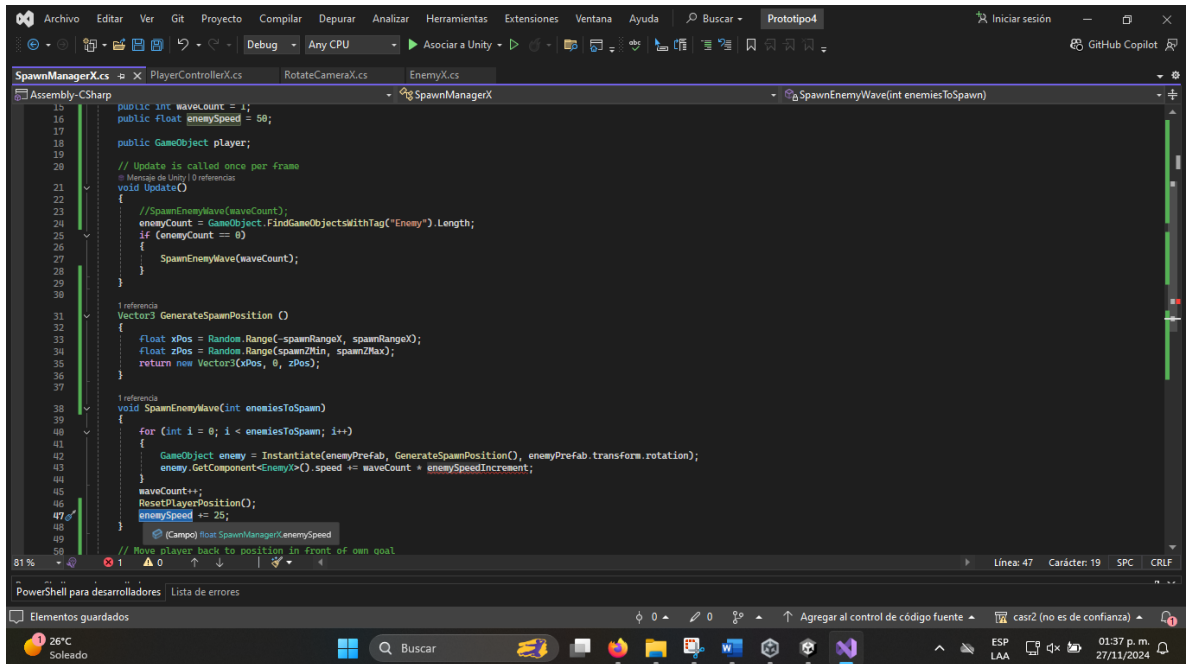
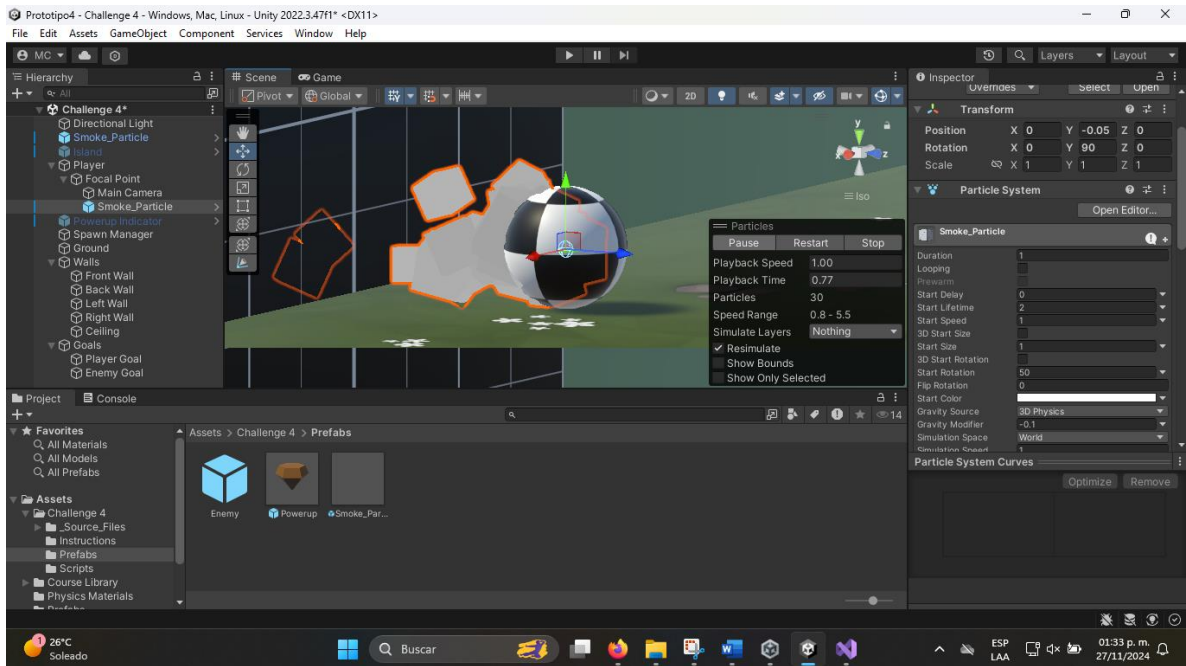
Elementos guardados

26°C Soledad

Buscar

ESP LAA 01:24 p. m. 27/11/2024





Archivo Editar Ver Git Proyecto Compilar Depurar Analizar Herramientas Extensiones Ventana Ayuda Prototipo4

Debug Any CPU Asociar a Unity

SpawnManagerX.cs PlayerControllerX.cs RotateCameraX.cs EnemyX.cs

Assembly-CSharp EnemyX

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 [Script de Unity (1 referencia de recurso) | 1 referencia]
6 public class EnemyX : MonoBehaviour
7 {
8     public float speed;
9     private Rigidbody enemyRb;
10    private GameObject playerGoal;
11    private SpawnManagerX spawnManagerX;
12
13    // Start is called before the first frame update
14    // Mensaje de Unity | 0 referencias
15    void Start()
16    {
17        enemyRb = GetComponent<Rigidbody>();
18        playerGoal = GameObject.Find("Player Goal");
19        spawnManagerX = GameObject.Find("Spawn Manager").GetComponent<SpawnManagerX>();
20        speed = spawnManagerX.enemySpeed;
21    }
22
23    // Update is called once per frame
24    // Mensaje de Unity | 0 referencias
25    void Update()
26    {
27        // Set enemy direction towards player goal and move there
28        Vector3 lookDirection = (playerGoal.transform.position - transform.position).normalized;
29        enemyRb.AddForce(lookDirection * speed * Time.deltaTime);
30    }
31
32    // Mensaje de Unity | 0 referencias
33    private void OnCollisionEnter(Collision other)
34    {
35        // If enemy collides with either goal, destroy it
36        if (other.gameObject.name == "Enemy Goal")
37        {
38            Destroy(gameObject);
39        }
40    }
41}
```

81% No se encontraron problemas. Línea: 17 Carácter: 1 SPC MIXTO

PowerShell para desarrolladores Lista de errores

Elementos guardados

EUR/CNY +0.66%

Buscar

ESP LAA 01:42 p. m. 27/11/2024

Prototipo4 - Challenge 4 - Windows, Mac, Linux - Unity 2023.4.7f1* <DX11>

File Edit Assets GameObject Component Services Window Help

