SESSIÓ 3. (3)

Afegir enemics al joc

6. Obtenir punts en destruir enemics: Marcador.

Una de les maneres més senzilles de saber com avancem en el joc és mantenir un marcador. Sempre que destruim un enemic, la puntuació augmentarà.

6.1 Programar Interfaces d'usuari en Unity

Unity sempre a tingut un sistema propi per programar GUI però la dificultat d'ús i la ineficiència dels resultats han fet que, històricament, els desenvolupadors hagin preferit altres eines (p.e. NGUI, disponible en el Unity Asset Store).

En les últimes versions de Unity es proporciona una nova eina per desenvolupar interfícies d'usuari. Sembla que és molt millor que el que hi havia abans però, de moment, hi ha gent que la utilitza i d'altra que continua amb les eines de sempre.

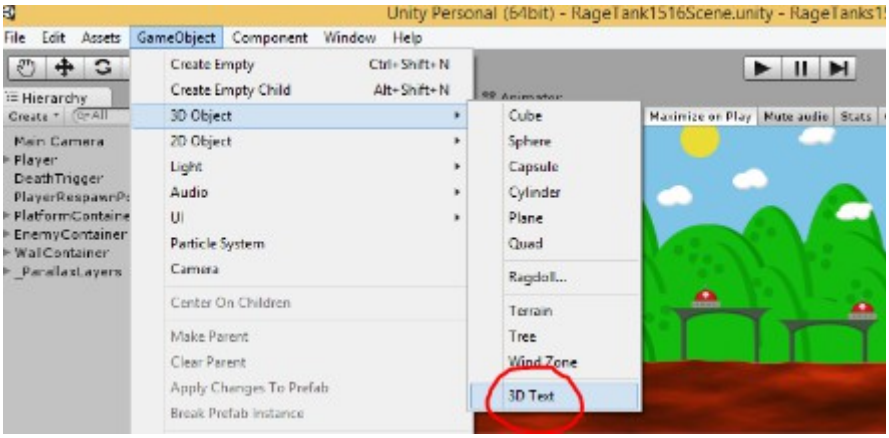
Per als nous desenvolupadors en Unity és interessant començar a familiaritzar-se amb el nou sistema de Unity per programar GUIs. Es pot trobar un tutorial interessant a:

https://www.raywenderlich.com/114700/introduction-unity-ui-part-1
https://www.raywenderlich.com/114764/introduction-unity-ui-part-2
https://www.raywenderlich.com/114828/introduction-unity-ui-part-3

En el projecte que estem treballant, amb poques necessitats de UI, sortirem del pas creant objectes i associant-los a la càmera. El resultat és prou acceptable.

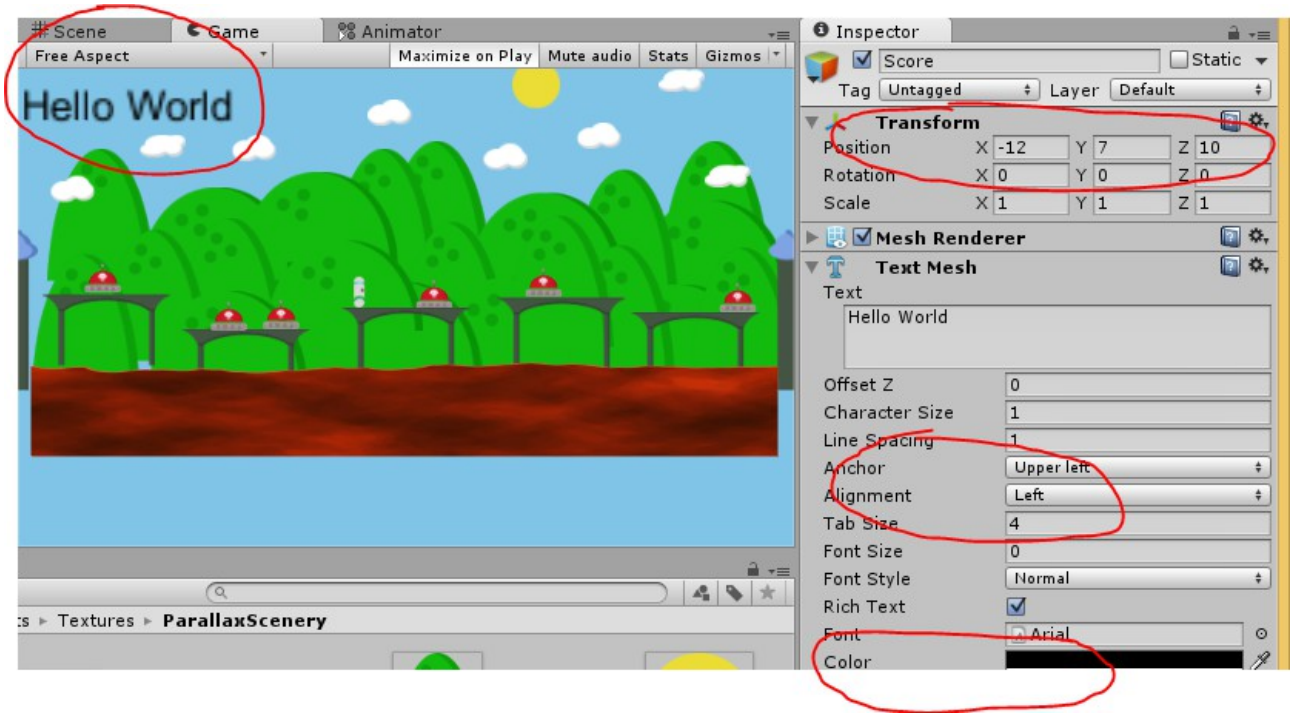
6.2 Afegir un marcador al joc: elements gràfics

Creem un objecte "**3D Text**" (`GameObject > Create Other > 3D Text`. **En Unity 5**: `GameObject > 3D Object > 3D Text`).



Situem el nou objecte dins la jerarquia com a **fill de Main Camera** i l'anomenem **Score**.

Establim el **Color** del text com negre, **Anchor** com Upper Left i **Alignment** com Left. Modifiquem el vector de **position** del component **Transform** com convingui perquè el text quedi a dalt a l'esquerra.



6.3 Afegir un marcador al joc: actualitzar valor visualitzat

Per aconseguir que el **Score** mostri la puntuació del jugador

- crearem un nou script, que anomenarem **ScoreWatcher** i que associarem a l'objecte **Score**.
- Completarem l'script **EnemyControllerScript**, de manera que generi l'event **enemyDied**.

El codi del nou script **ScoreWatcher** és:

```
using UnityEngine;
using System.Collections;

public class ScoreWatcher : MonoBehaviour
{
	public int currScore = 0;
	private TextMesh scoreMesh = null;

	void Start()
	{
		scoreMesh = gameObject.GetComponent<TextMesh>();
		scoreMesh.text = "0";
	}

	void OnEnable()
	{
		EnemyControllerScript.enemyDied += addScore;
	}

	void OnDisable()
	{
		EnemyControllerScript.enemyDied -= addScore;
	}

	void addScore(int scoreToAdd)
	{
		currScore += scoreToAdd;
		scoreMesh.text = currScore.ToString();
	}
}
```

Comentaris al codi:

- El mètode **Start()** inicialitza el valor del text.
- Al mètode **OnEnable()** l'objecte **Score** s'apunta a escoltar, per mitjà del mètode **addScore()**, l'event **enemyDied**, que generarà l'script **EnemyControllerScript**, de seguida que es detecti la mort d'un **Enemic**.
- Al mètode **OnDisable()** es realitza l'operació contraria a **OnEnable()**.
- El mètode **addScore()** el programem nosaltres (no s'hereda de **MonoBehaviour**) i és el **listener** que proporciona l'objecte **Score** per l'event **enemyDied**. La seva funció és actualitzar el marcador amb els punts que se li passen com a paràmetre.

(Si un altre event s'ha de reflexar en el marcador, ni haurà prou d'apuntar el mètode **addScore()** com a escoltador d'aquest event).

Les modificacions a **EnemyControllerScript** són:

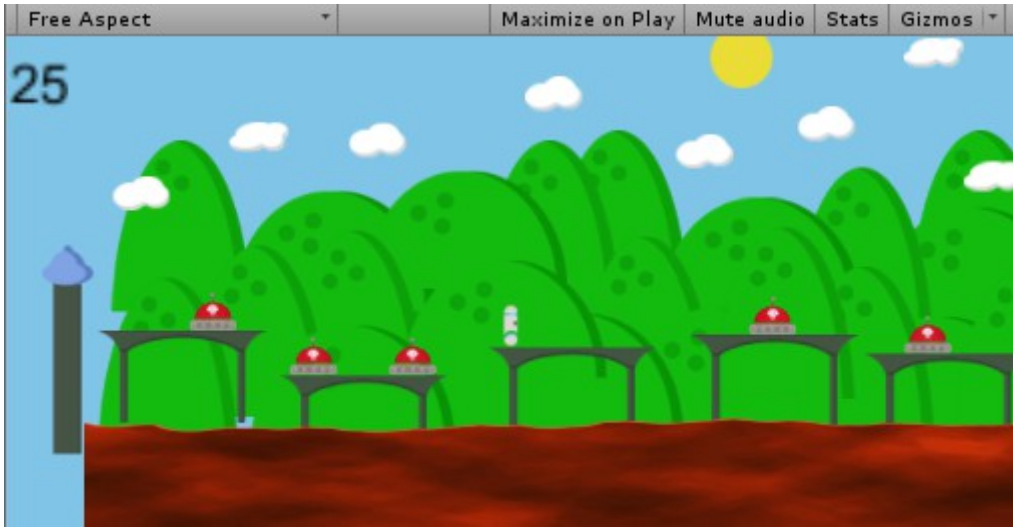
1. Declarar el delegate **enemyEventHandler** (signatura de mètode, que encaixa amb **addScore()**) i l'event **enemyDied**, que podrà ser atés per qualsevol mètode amb signatura igula a la definida pel delegate **enemyEventHandler**. Incloure al principi de l'script el següent codi:

```
// Delegat i event que permetran als objectes del joc saber quan mor un Enemic
public delegate void enemyEventHandler(int scoreMod);
public static event enemyEventHandler enemyDied;
```

2. Just **abans de destruir l'objecte Enemy**, generar l'event **enemyDied**, perquè se n'assabentin el objectes del joc que hi estiguin interessats. Es passa la puntuació que correspon.

```
// Generar event enemyDied i donar una puntuacio de 25 punts.
if(enemyDied != null)
      enemyDied(25);
```

Com a resultat, per cada enemic eliminat, se sumen 25 punts al marcador.



EXPERIMENT

Executar el joc tal com està en aquest moment, amb els últims canvis.
Revisar el codi i entendre el mecanisme utilitzat.

7. Regenerar enemics.

Per fer el joc més interessant, implementarem un sistema de regeneració d'enemics. Si no ho fessim, seria massa fàcil guanyar ...

Afegirem un nou element visual, un vòrtex, des del qual s'incorporaran nous enemics al joc. La creació d'enemics dependrà de la mort dels enemics actuals i també tindrà un component aleatori.

7.1 Creació dels objectes "vòrtex": primer objecte

A la carpeta **Textures** hi creem una nova carpeta **Vortex**. Hi importem els sprites **Vortex_Back**, **Vortex_Center** i **Vortex_Front**, que es troben al zip de Recursos.

Creem un nou objecte buit que anomenem **Vortex** i li afegim com a fills els tres sprites. Assignarem als tres objectes la posició X:0, Y:0, Z:0 i com a valor de **Order in Layer** 10, 11 i 12 respectivament.

7.2 Generació d'enemics a través del "vòrtex"

La generació d'enemics a través del Vòrtex la realitzarà el nou script **EnemyRespawner**, que associarem a l'objecte **Vortex.** El codi de l'script és:

```
using UnityEngine;
using System.Collections;

public class EnemyRespawner : MonoBehaviour
{
	public GameObject spawnEnemy = null;
	float respawnTime = 0.0f;

	void OnEnable()
	{
		EnemyControllerScript.enemyDied += scheduleRespawn;
	}

	void OnDisable()
	{
		EnemyControllerScript.enemyDied -= scheduleRespawn;
	}

	void scheduleRespawn(int enemyScore)
	{
	// Randomly decide if we will respawn or not
		if(Random.Range(0,10) < 5)
		return;
		respawnTime = Time.time + 4.0f;
	}

	void Update()
	{
		if(respawnTime > 0.0f)
		{
			if(respawnTime < Time.time)
```

```
                {
                        respawnTime = 0.0f;
                        GameObject newEnemy =
                        Instantiate(spawnEnemy) as GameObject;
                        newEnemy.transform.position = transform.position;
                }
        }
    }
}
```

A la variable pública **spawnEnemy** se li ha d'assignar el **prefab del Enemy**, des del Object Inspector.

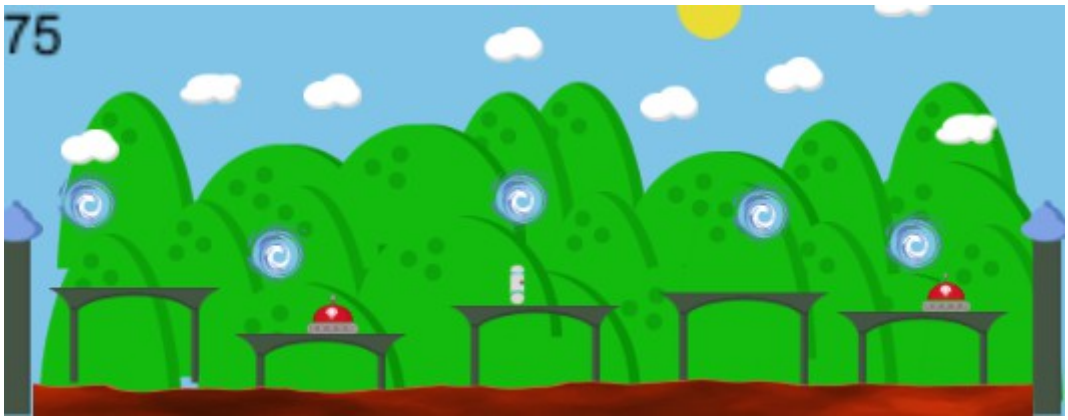
EXPERIMENT

Executar el joc tal com està en aquest moment, amb els últims canvis.
Revisar el codi i entendre el mecanisme utilitzat.

Activitats

1. Implementar una animació sobre el Vòrtex fent-lo rotar sobre l'eix Z. (Es pot fer amb un script).

2. **Convertir l'objecte Vortex en prefab**, esborrar l'objecte Vortex original i posar uns quants Vortex sobre el terreny del joc. Probar el joc.



3. Comentar el codi i explicar el funcionament de l'script **EnemyRespawner**. Pensar com es podria modificar el codi perquè la regeneració d'enemics fos més/menys freqüent.

4. Implementar el marcador del joc utilitzant l'eina de generació de GUI de Unity. Explicar els passos que s'han seguit.

5. Fer un llistat amb els conceptes introduits en aquest document, indicar la pàgina i explicar què són i per a què s'utilitzen.