

# PROJECTE FINAL (SURVIVAL)

## Introducció:

Videojoc de guerrers medievals, acabar amb els enemics amb l'espasa, ells ens poden matar i treure una vida, tenim 3 vides en cada partida. també hi han els cilindres per destrossar, van apareixen per tot el terreny a mesura que els fem explotar amb una coalició de la nostra espasa.



Poden jugar dos jugadors en equip o un jugador individual, opció escollida per l'usuari en la pantalla *option*, com també podem escollir de baixar o pujar la resolució i el volum de l'aplicació.



Finalment una última pantalla on tenim les millors puntuacions obtingudes. Els registres són guardats en una base de dades Sqlite3 per les versions executables d'escriptori a 64 bits, i els registres de l'aplicació webGL en memòria de l'equip, com no la utilitat del llenguatge C# *"PlayerPrefeb"*.



## **Instruccions:**

Cada cop que destruïm un cilindre tenim 10 punts, els enemics en el primer nivell venen 1 per jugador, són 5 punts si acabes amb ells, però si et maten perds una vida, tens un màxim de tres vides.

Per aconseguir pujar de nivell has de fer al menys 75 punts, el qual baixarà el pont i un cop creuat augmentarà de nivell. Evitar caure en el riu perquè perds una vida.

Cada nivell superat augmentarà el nombre d'enemics per jugador, també 50 punts més en l'anotació de cada jugador, i si tens menys de 3 vides, aconsegueixes una vida més.

Els jugadors es desplaçaran pel terreny amb les tecles següents:

Jugador 1: ←, →, ↑, ↓ (esquerra, dreta, endavant i enrere).

Jugador2: per correlació amb el jugador 1 són A, D, W i S.

Tenim les tecles de saltar i 3 atacs, són les següent:

Jugador1: [-] atac directa, [.] atac amb gir, [,] per saltar i finalment la combinació de les tecles [,] i [.] és un atac amb salt.

Jugador2: [C] atac directa, [V] atac amb gir, [B] per saltar i la combinació de [B] i [V] tenim l'atac amb salt.

## **Realització:**

Per realitzar un videojoc *arcade* (aventures), primer m'he proposat assolir uns objectius, una taula de registres de les millors anotacions (1 pantalla per ell, en *Unity* és una escena).

El segon objectiu és de poder escollir jugar individual o en equip, o sigui 1 o 2 jugadors, juntament amb els controls de resolució i volum un altre escena de *Unity*.

Sumem la portada del joc i el desenvolupament de cada partida, són quatre escenes, que tenim de coordinar entre elles per botons clicant amb el *mouse* i tecles del teclat, s'ha triat les següent tecles per les versions executables d'escriptori:

Escape: Sortir del programa.

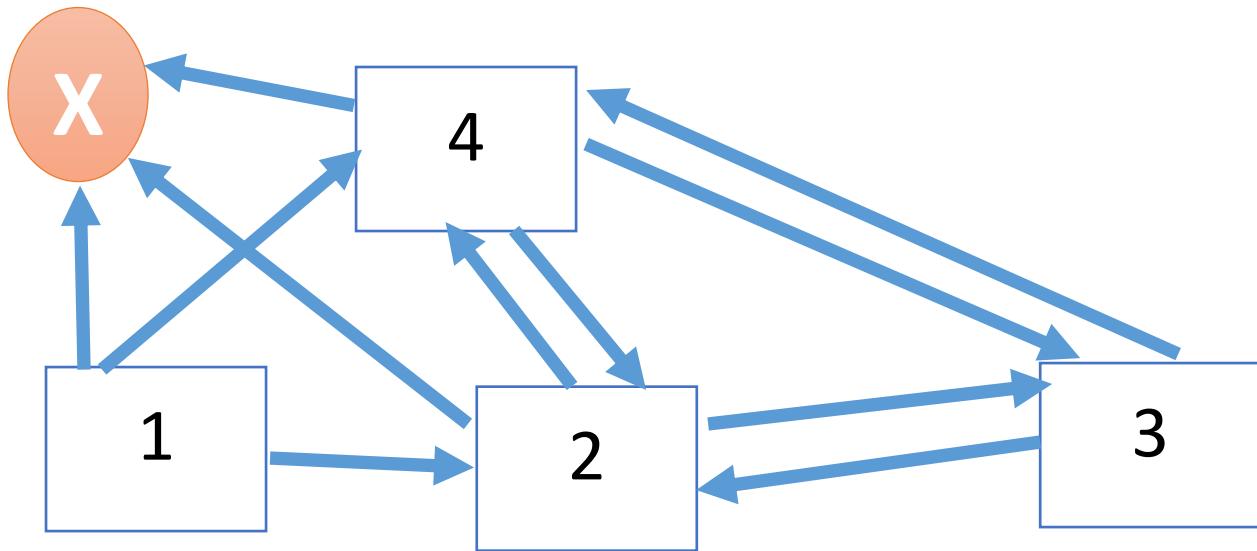
Space: Començar la partida.

Return: Acabar partida, o tornar a una pantalla anterior.

La primera lletra majúscula de cada opció ens enviaria a la pantalla destí.

Pel disseny de cada pantalla volem tenir unides entre elles de tal manera que podem començar partida o sortir de l'aplicació en cada una d'elles o al menys torna a una pantalla anterior, finalment el resultat ha sigut en el proper diagrama de blocs.

1. Pantalla d'inici
2. Pantalla d'opcions
3. Pantalla d'anotacions
4. Pantalla desenvolupament del joc



Per tal de tenir l'opció de modificar el nombre de jugadors per jugar la partida, des de la taula d'anotacions no anem a la sortida del programa sinó que anem una pantalla enrere, en la pantalla d'opcions del joc, o sigui de la pantalla 3 a la 2.

Per realitzar la primera pantalla, portada del joc prèviament instal·larem *TextMeshPro* del nostre *Package Manager*. Ara en la configuració del *Canvas*, podem usar totes les eines de TMP disponibles, com ara el Text del títol i de les opcions que podem accedir-hi des de 3 punts diferents, clicant els botons o texts amb el *mouse* i des de el teclat, amb la primera tecla de l'opció. Per accedir-hi a l'opció desitjada cridarem a la funció següent editada en llenguatge C# del propi editor *Visual Studio*.

```
public void PlayGame(int escena)
{
    SceneManager.LoadScene(escena);
}
```

On cada escena, pantalla del joc té un nombre determinat en la *Build Setting*, el qual ha de concedir amb el paràmetre *integer escena*, i ací carregar la propera escena amb la llibreria *SceneManagement*. Cal destacar també de fer una animació amb temps d'espera abans de carregar l'escena amb la següent funció específica *IEnumerator* del llenguatge C#.

```
IEnumerator StartGame()  
{  
    yield return new WaitForSeconds(6);  
    anim.SetTrigger("fade");  
}
```

Cridada des del mètode *start()*:

```
private void Start()  
{  
    Time.timeScale = 1;  
    StartCoroutine(StartGame());  
}
```

En la segona pantalla usarem més utilitats de la llibreria TMP (*TextMeshPro*) a part del Text del títol i opció de anar a la pantalla d'anotacions, són la *Toggle* (per modificar el nombre de jugadors), *Slider* (per modificar el volum del videojoc) i el *Dropdown* (opció per canviar de resolució de pantalla).

Per fer el canvi de nombre de jugadors amb la utilitat *Toggle* usem un *boolean* per fer l'intercanvi.

```
public bool players;
```



Dins del mètode *Update()* tenim el següent *script*:

```
    if (GameObject.Find("OnePlayer").GetComponent<Toggle>().isOn
&& players)
    {
        GameObject.Find("TwoPlayer").GetComponent<Toggle>().isOn
= false;
        players = !players;
    }

    if (GameObject.Find("TwoPlayer").GetComponent<Toggle>().isOn
&& !players)
    {
        GameObject.Find("OnePlayer").GetComponent<Toggle>().isOn
= false;
        players = !players;
    }
```

Per realitzar la modificació del volum serà amb la variable,

```
public AudioSource musicaPrincipal;
```

Tan sols afegir en el *Update()*.

```
void Update()
{
    musicaPrincipal.volume = PlayerPrefs.GetFloat("volume");
```

Aquesta variable serà modificada per varies funcions on estan relacionades amb els paràmetres de control del component *AudioSource* del objecte corresponent, com ara la següent que es mostra a continuació.

```
public void plusVolum()
{
    float music=controladormusica.value;
    if (music <= 0.95f) music += 0.05f;
    controladormusica.value = music;
    PlayerPrefs.SetFloat("volume", music);
}
```

Item per la funció *lessVolum()*.

## **Enllaç del projecte:**

Projecte s'obre amb el maquinari d'*Unity*, versió 2020.3.2f1:

<https://drive.google.com/drive/folders/1T9NX49-zYIPqtutogiQEaO7yisDWieKL?usp=sharing>

Aplicació web, construït amb WebGL:

<https://gilbertvs.github.io/survival/>

i en el servidor simmer.io:

<https://simmer.io/@Equalitzator/survival>

Alumne: Gilbert Viader i Sauret.

Curs: Disseny de videojocs amb Unity.

Domini: <http://denviader.gq/>