

PRÀCTICA 4: "SOLDIER"

REQUERIMENTS

Realitzar un vídeo joc des de zero, on el personatge és un soldat en primera persona i els enemics són kamikazes els quals podem esquivar o disparar-los en el cap per fer-los explotar i aconseguir 5 punts, cada cop que ens toquen perdem una vida.

En el marge superior dret tenim la puntuació obtinguda, en el marge inferior dret hi ha les vides com ens mostra la següent il·lustració, s'ha d'augmentar la velocitat dels enemics cada 60 segons amb la variable *TimeGame* com queda indicat, també he afegit un increment de dos enemics en el terreny per cada minut transcorregut, i ací augmenta el nivell de dificultat a mesura que anem arribant en el pou d'aigua del territori enemic.

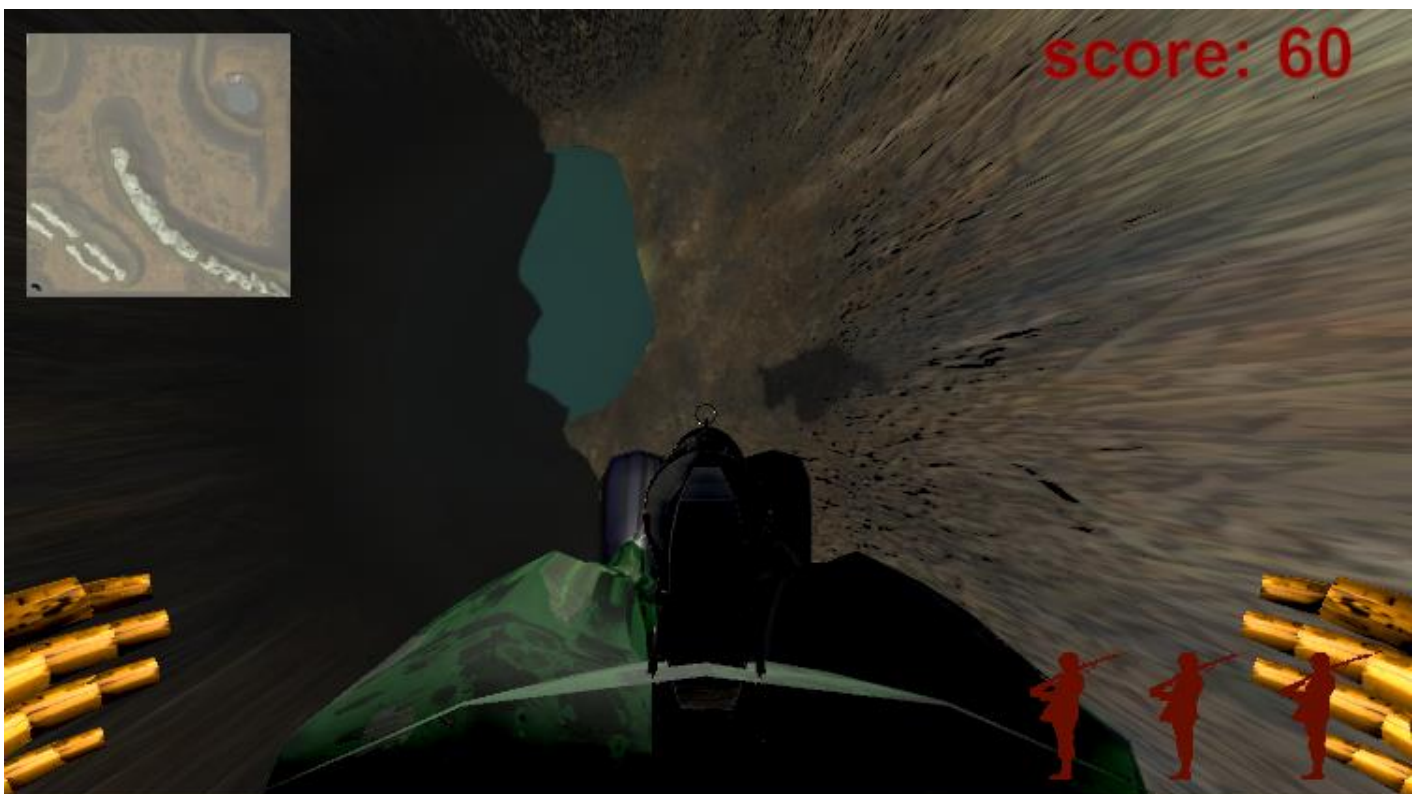


Com ens mostra en la següent captura, s'ha afegit un mapa, és una càmera en posició vertical on ens mostra el terreny, per poder localitzar la posició del soldat com la dels enemics, ubicat en la part superior esquerra, en el vídeo joc també hi ha la possibilitat d'aturar o tornar activar la partida amb la tecla "P",

com de sortir de l'escenari i torna a la pantalla principal amb la tecla "ENTER" (retorn).



El disparador de la metrallera, la pèrdua d'una vida o partida s'executarà amb la tecla "SPACE"; les tecles A, S, D i W són de desplaçament en l'escenari, finalment amb el *mouse* tenim 360° del camp visual del camp de batalla.



Per aconseguir un joc que no s'acaba mai, l'objectiu principal és d'arribar en el terreny dels enemics on tenen el seu pou d'aigua, quan hi entrem aconseguim una vida extra, 100 punts i començar en la posició inicial per seguir jugant amb un nivell de dificultat afegit.

FITXA TÈCNICA

Hi ha dos escenaris en el vídeo joc, l'escena principal, és la portada del vídeo joc i l'altre és l'escena on es realitza les interaccions del usuari amb el seu personatge per desenvolupar el vídeo joc i aniquilar els enemics, són les següent:

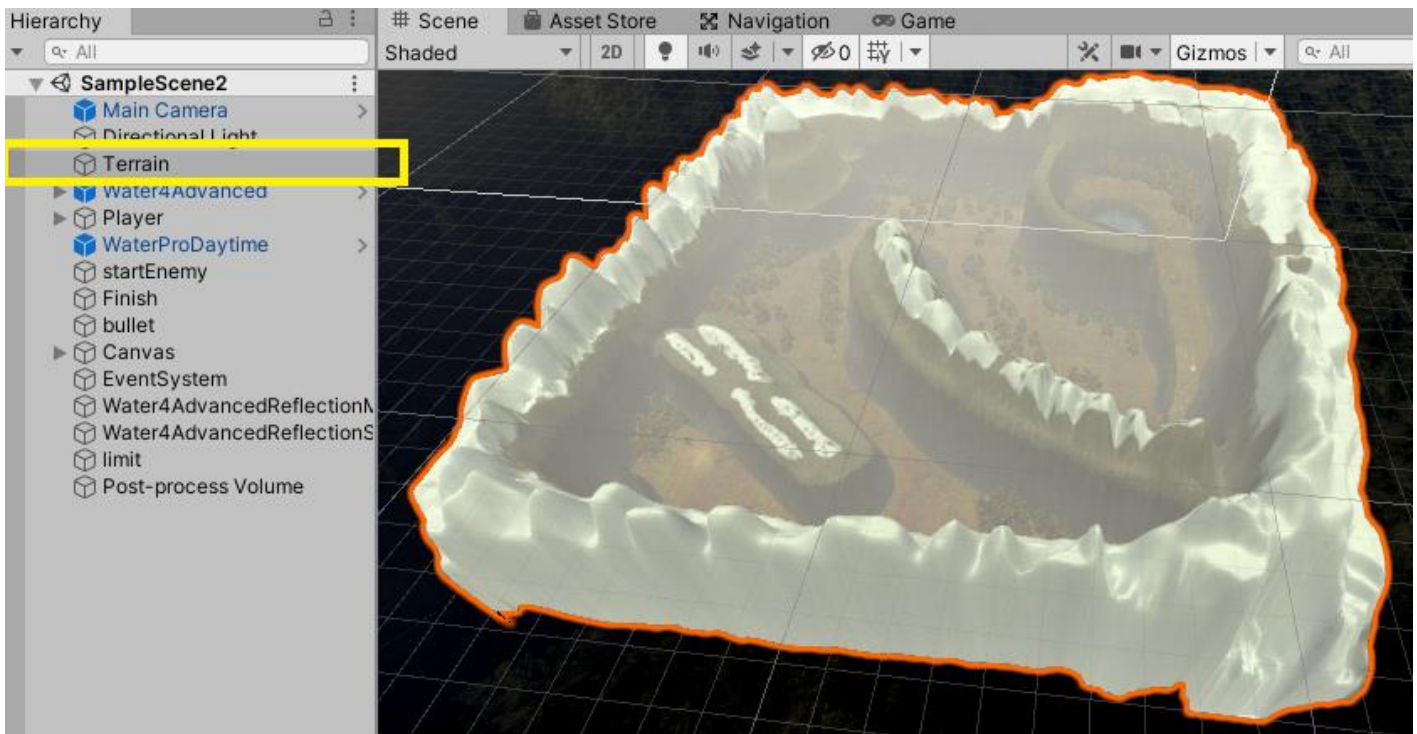
- **SampleScene**: portada del vídeo joc, tenim dues opcions entrar-hi, amb la tecla "space" o sortir de l'aplicació, amb la tecla "enter (retorn)", realitzat amb tipografia *Arial (bold)*, *R: 155; G:5; B:5*.



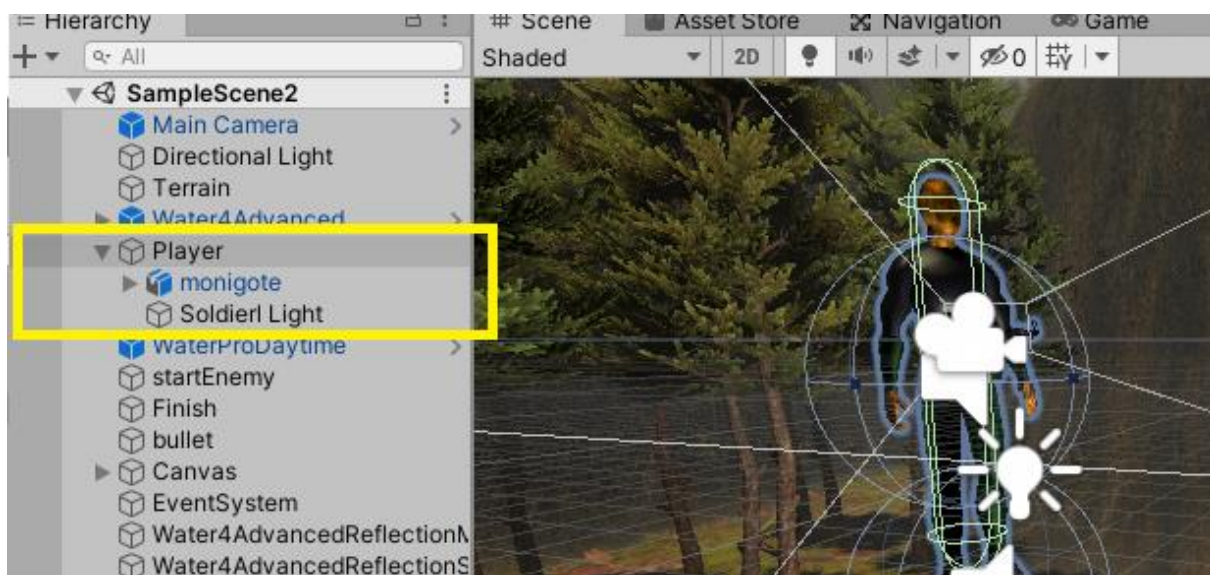
- **SampleScene2**: Interacció amb l'usuari pel soldat amb càmera de primera persona, una segona càmera a dalt a l'esquerra quadrada, en posició vertical del nostre terreny, visualització del mapa amb la localització del enemics i jugador.

Hi ha quatre *GameObject* principal són els següents:

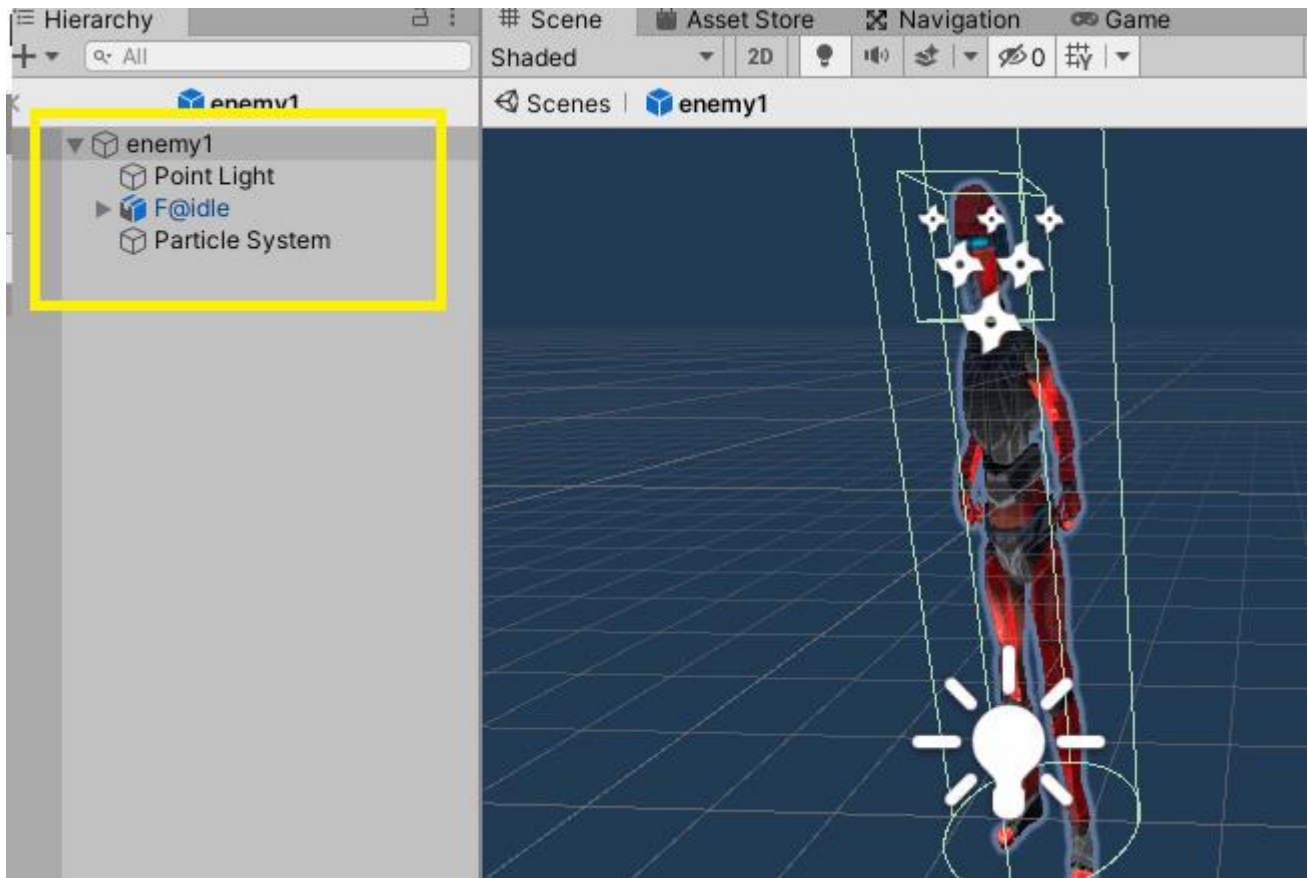
- **Terrain**: Són els límits del joc, realitzat amb una mida quadrada de 150x150 i un alçat (eix Y) de 100 màxim distribuïts els desnivells per tot el camp de batalla.



- **Player:** Hi ha dos *gameObject* fill, un és el *Soldier Light* que ens farà trobar la localització d'aquest en el mapa, i el més important un modelatge en 3D anomenat **monigote** ([sketchfab, "Monigote hombre"](https://sketchfab.com/3d-models/Monigote-hombre)), que conté una sèrie *gameObject* fills que són les parts del seu cos i la **càmera** principal, per realitzar les funcions en primera persona del personatge, i d'aquest l'últim *gameObject* fill que és l'arma una metrallera M 40 ([sketchfab, "gun"](https://sketchfab.com/3d-models/gun)). Hi ha també el component del script principal *MovePlayer*, on ens calcula la lògica de control del joc ací com les físiques dels moviments dels personatges, coalicions entre ells i la gravetat en el terreny.



- **startEnemy:** *gameObject* buit punt d'inici dels enemics on sortiran cada 5 segons aleatòriament un dels quatre enemics mitjançant una *Instantiate* (instància) els *Prefab*, enemy1, enemy2, enemy3 o enemy4. Els quals tenen el component del script, *followingPlayer*, del moviment dels enemics perseguint el personatge.



- **bullet:** és la bala, *gameObject* independent del jugador, degut als moviments de translació d'un punt fora de l'escena al punt indicat per l'objectiu de l'arma quan premem la tecla *space* o deixar de pitxer torna el seu punt inicial desenvolupat en el script *shootBallet*. Té un altre scripts associats *movingBullet*.

Altres *gameObject* secundaris com el llum direccionals, Canvas, *Prefab* de l'aigua del pou, límits si sortim de l'escenari, reflexes de l'aigua...

El projecte s'ha realitzat amb 6 scripts i són els següent:

1. **movePlayer**, script principal on es realitzarà la lògica de control de puntuacions, vides, nombre d'enemics, dins del *Update ()*

```
// increment el nombre d'enemics cada 60 segons
timeGame += Time.deltaTime;
if (timeGame >= 60.0f)
{
    timeGame = 0.0f;
    if (enemy < 15)
    {
        enemy += 2;
        PlayerPrefs.SetInt("enemyReg", enemy);
    }
}
// creació d'enemics per cada 5 segons
time += Time.deltaTime;
if (time >= 5.0f)
{
    time = 0.0f;
    createEnemy();
}
// perdem una viada o la última vida
if (dead || finish) Time.timeScale = 0;
if (dead && Input.GetKeyDown(KeyCode.Space))
{
    UnityEngine.Random.InitState(Time.frameCount); // posició nova
de la sèrie Random
    Time.timeScale = 1;
    dead = false;
    GameObject.Find("lifeTx").GetComponent<Text>().enabled =
false;
}
if (finish && Input.GetKeyDown(KeyCode.Space))
{
    Time.timeScale = 1;
    SceneManager.LoadScene(0);
}
```

funció de crear enemies:

```
private void createEnemy()
{
    if (enemy > 0)
    {
        enemy -= 1;
        PlayerPrefs.SetInt("enemyReg", enemy);
        int nEnemy = UnityEngine.Random.Range(0, 4);
        Instantiate(enemyPref[nEnemy], enemyPost.transform.position,
Quaternion.identity);
    }
}
```


moviments físics del jugador, gravetat i coalicions d'aquest, controls d'aturada, final, avortament del joc... en el mètode *Update()*:

```
float x = Input.GetAxis("Horizontal");
float z = Input.GetAxis("Vertical");
Vector3 move = transform.right * x + transform.forward * z;
cc.Move(move * speed * Time.deltaTime);
// canvi de clip de moviment
if (x == 0 && z == 0)
{
    movingPlayer.SetBool("idle", true);
    movingPlayer.SetBool("run", false);
} else
{
    movingPlayer.SetBool("idle", false);
    movingPlayer.SetBool("run", true);
}
// simulació de la gravetat en el personatge
velocity.y += gravetat * Time.deltaTime;
cc.Move(velocity * Time.deltaTime);
if (velocity.y < 0) velocity.y = -2f;
```

2. **mouseLook**, script per moure la càmera de la visió del vídeo joc amb el *mouse*.

```
public class mouseLook : MonoBehaviour
{
    public float sensibilitat = 100;
    public Transform playerBody;

    public float xRotation;

    // Start is called before the first frame update
    void Start()
    {
        Cursor.lockState = CursorLockMode.Locked;
    }

    // Update is called once per frame
    void Update()
    {
        float mouseX = Input.GetAxis("Mouse X") * sensibilitat * Time.deltaTime;
        float mouseY = Input.GetAxis("Mouse Y") * sensibilitat * Time.deltaTime;

        playerBody.Rotate(Vector3.up * mouseX);
        xRotation = xRotation - mouseY;
        xRotation = Mathf.Clamp(xRotation, -90, 90);
        transform.localRotation = Quaternion.Euler(xRotation, 0, 0);
    }
}
```

```
}
```

3. **followingPlayer**, script mitjançant el *Nav Mesh Agent* els enemics detectaran el personatge i el perseguiran.

```
public class followingPlayer : MonoBehaviour
{
    [SerializeField] private Transform destination;
    [SerializeField] private NavMeshAgent myNash;
    [SerializeField] private Animator runningEnemy;
    [SerializeField] private float time, timeGame;
    [SerializeField] private bool enemyDead;
    private void Start()
    {
        destination = GameObject.Find("Player").GetComponent<Transform>();
        runningEnemy.SetBool("run", true);
        runningEnemy.SetBool("dead", false);
    }
    void Update()
    {
        // trobar el jugador per la Nav mesh
        myNash.SetDestination(destination.position);
        // augment de velocitat dels enemics cada 60 segons
        timeGame += Time.deltaTime;
        if (timeGame >= 60.0f)
        {
            timeGame = 0.0f;
            myNash.speed += 1;
        }
        // retart d'un segon per destruir l'enemic
        if (enemyDead)
        {
            time += Time.deltaTime;
            if (time >= 1.5f)
            {
                time = 0f;
                Destroy(gameObject);
            }
        }
    }
    public void OnTriggerEnter(Collider other)
    {
        if (other.tag == "Bala")
        {
            runningEnemy.SetBool("run", false);
            runningEnemy.SetBool("dead", true);
            enemyDead = true;
        }
    }
}
```


4. **shootBallet**, script que ens farà la translació de la bala al punt on disparem amb la tecla *space*, realitzat amb el mètode *Physics.Raycast*.

```
public class shootBallet : MonoBehaviour
{
    // variables initialization
    public Ray bala;
    public Vector3 fwd;
    public RaycastHit objectiu;
    public Transform desti;

    void FixedUpdate()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            UnityEngine.Random.InitState(Time.frameCount); // posició nova de la sèrie
            Random
            GameObject.Find("bullet").GetComponent().enabled = false;
            GameObject.Find("Player").GetComponent().enabled = false;
            GameObject.Find("bullet").GetComponent().enabled = true;
            fwd = transform.TransformDirection(Vector3.forward);
            Physics.Raycast(transform.position, fwd, out objectiu);
            // imprimir per consola la distància del nostre tret
            //print("Trobat un objecte a la distància de " + objectiu.distance);
            desti.transform.position = new Vector3(objectiu.point.x, objectiu.point.y,
            objectiu.point.z);
        }else desti.transform.position = new Vector3(0f, 70f, 0f);
    }
}
```

5. **movingBullet**, detecta la coalició amb els enemics i ens donarà prioritat en el sons d'explosió amb ells, com activa el sistema de partícules d'ells.

```
public class movingBullet : MonoBehaviour
{
    [SerializeField] private int enemy;
    [SerializeField] private int score;

    public void OnTriggerEnter(Collider other)
    {
        if (other.tag == "Enemy")
        {
            score = PlayerPrefs.GetInt("scoreReg");
            score += 5;
            PlayerPrefs.SetInt("scoreReg", score);
            enemy = PlayerPrefs.GetInt("enemyReg");
            enemy += 1;
            PlayerPrefs.SetInt("enemyReg", enemy);
            GameObject.Find("bullet").GetComponent().enabled = false;
            GameObject.Find("Player").GetComponent().enabled = false;
            GameObject.Find("startEnemy").GetComponent().enabled = false;
            GameObject.Find("startEnemy").GetComponent().enabled = true;
            other.gameObject.GetComponentInChildren<ParticleSystem>().Play();
            other.gameObject.GetComponent<BoxCollider>().enabled = false;
        }
    }
}
```

```

    }
}
}

```

6. **mainStart**, ens farà la gestió de entrades i sortides del joc des de l'escena inici.

```

using UnityEngine;
using UnityEngine.SceneManagement;

public class mainStart : MonoBehaviour
{
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            SceneManager.LoadScene(1);
        }

        if (Input.GetKeyDown(KeyCode.Return))
        {
            Application.Quit(); // per build executables
            // SceneManager.LoadScene(0); // per buid webs.
        }
    }
}

```

REALITZACIÓ

Realitzar un terreny en el qual hi han dos nivells principals, la primera consta d'un 75% del terreny amb muntanyes, arbres i plantes; d'on començarà el jugador i el segon és on comencen els enemics i tenen el pou d'aigua, objectiu del jugador d'arribar i entrar-hi. El terreny consta d'un *width: 150; length: 150* i *height:100*, primer s'ha elevat tot el terreny a 20 i la segona elevació on hi ha el pou d'aigua a 35 els límits del terreny són muntanyes entre 60 -70 unitats que representen metres reals.

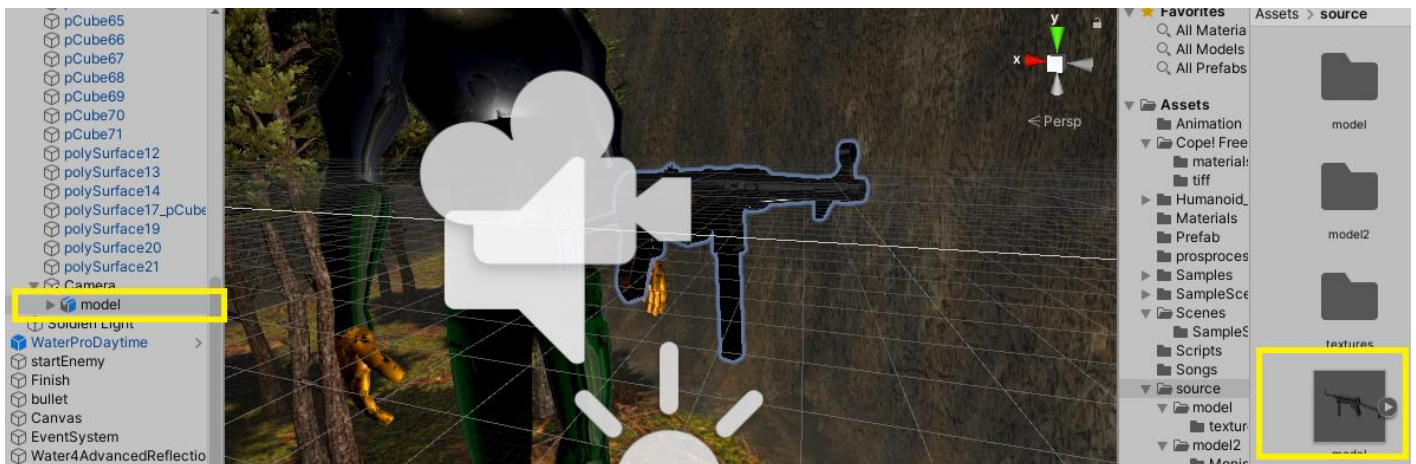
Mesh Resolution (On Terrain Data)	
Terrain Width	150
Terrain Length	150
Terrain Height	100
Detail Resolution Per Patch	32
Detail Resolution	1024

Les textures empleats són quatre, la herba de colors verds i carbasses per el terra, per les roques el color marró fosc per les

montanyes el gris i la neu per els pics de muntanya són els següents:



La realització del personatge en primera persona ha sigut un soldat prefabricat, tret de la [sketchfab, "Monigote hombre"](#) afegim un *gameObject* fill del personatge, la càmera principal del joc, i d'aquesta un altre *gameObject* fill la metrallera trobada en l'enllaç següent: [sketchfab, "gun"](#), s'han aplicat textures al personatge amb materials, de colors verd fosc i blau marí, i color carn pel cos.

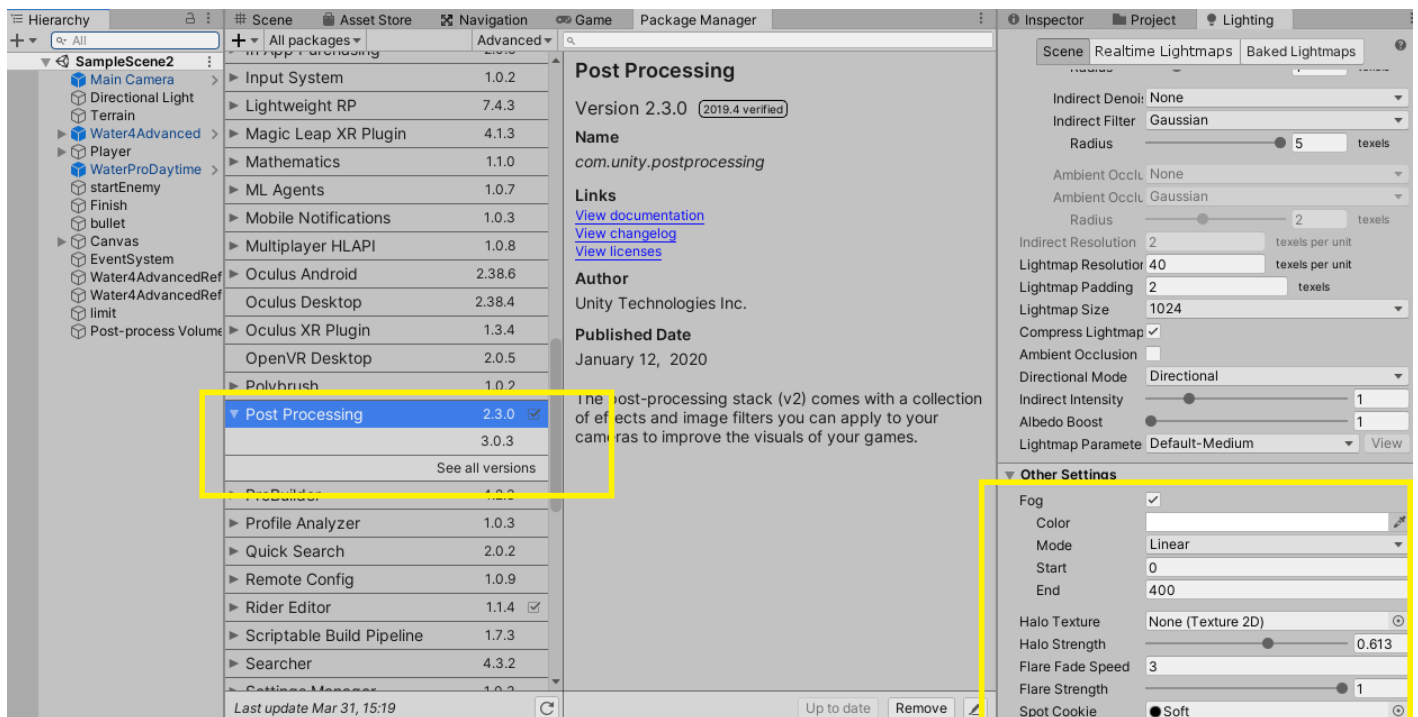


Per realitzar els enemics kamikaze, he usat de el paquet *Assets Store Unity*, [Humanoid Avatar Rig Free](#) on també hi ha el clip d'animacions d'aquest, com els materials que són compost de quatre gammes de colors, cada un d'ells en un *prefab* els quals seran instanciat en un *gameObject* anomenat "StartEnemy", i aniran sortint aleatòriament per la *class random* i controlant el nombre d'aquest per una variable *integer*, que anirà augmentant cada 60 segons. La realització de detecció i persecució dels enemics sobre el jugador configurat amb *Nav Mesh Agent*, increment de velocitat cada minut de l'eina *Navigation*.

Per realitzar l'aigua del pou, els arbres, les llums *Flade* s'ha usat el paquet *Assets Store Unity*, [Standard Assets](#), i el decorat del terreny, fet amb un *cube(Map)* del paquet *Assets Store Unity*, [Cope! Free Skybox Pack](#).



Les animacions i sistemes de partícules específic en cada enemic realitzades amb les eines *Animator* i *Particle System* amb una durada d'un segon, una elevada emissió, la figura d'una esfera i mides incrementals, colors determinats corresponent a cada enemic, el *Box Collider*, s'ha reduït a la mida dels caps dels enemics, component que pot haver-hi coalició amb el *Character Collider* de la *bullet* (bala) o el *Player* (jugador), per tal de donar-hi més dificultat en la partida.



Per acabar he afegit la boira (Fox) component de *Lighting Setting*, també he importat el paquet *Post Processing* del

Manager Assets per realitzar efectes visuals en el vídeo joc i aconseguir una millor qualitat en harmonia del terreny amb el *Cube(Map)*.



Són afegits els efectes visual del *Post Processing* de *Auto Exposure* i *Color Grading* per donar abrillantament i color en el vídeo joc, en canvi *chromatic aberration* i de *Vignette* tan sols per quan ens toquen els enemics *Kamikaze* i perdem una vida canviarem les *intensity* respectives des de el *script movePlayer*.

Declarem el objecte de *PostProcess* i les dues variables dels efectes en la funció *Start()* afegim el següent:

```
public PostProcessVolume ppv;  
private ChromaticAberration ChrAb = null;  
private Vignette vigneta = null;  
  
// Start is called before the first frame update  
void Start()  
{  
    ppv.profile.TryGetSettings(out ChrAb);  
    ppv.profile.TryGetSettings(out vigneta);  
}
```

Ara en el *OnTriggerEnter* quan faci la coalició amb l'enemic canviem la intensitat de les dos efectes:

```
if (other.tag == "Enemy")
{
    ChrAb.intensity.value = 1f;
    vigneta.intensity.value = 0.5f;

    ....
}
```

Finalment només ens cal tornar a posar els valors per defecte quan tornem a la partida amb la tecla *Space*.

```
if (dead && Input.GetKeyDown(KeyCode.Space))
{
    ChrAb.intensity.value = 0f;
    vigneta.intensity.value = 0f;

    ....
}
```

El projecte està en un fitxer comprimit "practica04.zip" en següent enllaç:

<https://drive.google.com/drive/folders/1PPK11LApV55uiWJMxbY-EDC6yckqWCS7?usp=sharing>

Enllaç del joc "Soldier" executable en versió web:

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

Alumne : **Gilbert Viader Sauret**

Curs: **Disseny de videojocs amb Unity**

Barcelona, 31 de març 2021