SESSIÓ 4. (1) v0

Preparar i completar el món del joc

Una vegada disposem dels personatges del joc, anem a preparar el món on es desenvoluparà.

1 Limitar l'espai de joc

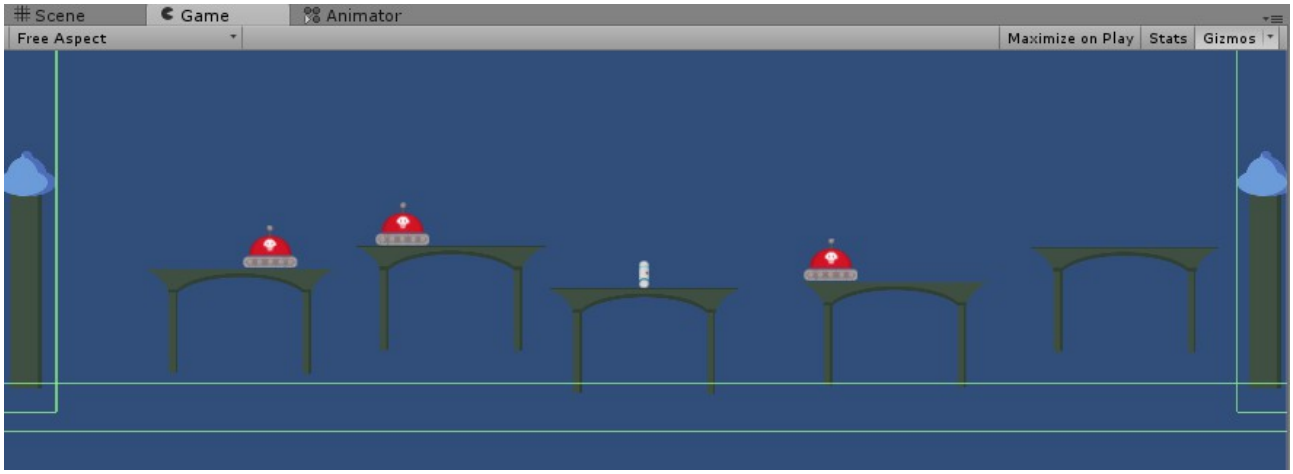
Primer de tot posem més plataformes, arrossegant el prefab **Platform** cap a l'escena les vegades que considerem adients. Ens assegurem que la distància entre plataformes permet que el **Player** pugui saltar d'una a l'altra sense problemes.

Situem enemics a les plataformes. Podem ajustar el prefab **Enemy**, de manera que la proporció de entre la mida dels enemics, la plataforma i el **Player** sigui l'adequada. També ajustem, si cal, la mida del **Player**.

Modifiquem la mida de l'objecte **DeathTrigger** de manera que cobreixi tot l'escenari per sota.

Per limitar l'espai pels costats utilitzarem un nou sprite: **Wall.png**. L'importem al joc, tal com hem fet amb les altres imatges (si cal, revisar els documents anteriors!), dins la carpeta **Assets/Textures/Scenery**.

Associem a **Wall** un **BoxCollider2D** amb mida Y:20, per evitar que el **Player** surti de l'escena per molt que salti i situem un Wall a cada costat de l'escena.



Per mantenir l'ordre, creem un objecte buit **WallContainer** i hi posem dintre els objectes **Wall**.

2 Posar imatges de fons. Parallax scrolling

Ara afegirem imatges de fons i, a més, gestionarem el fons implementant un Parallax scrolling.

L'efecte **Parallax** consisteix en que, quan hi ha desplaçament de càmara, els objectes mes propers a la càmara el mouen més ràpidament que els objectes del fons, més llunyans.

Com a imatges de fons tenim muntanyes inúvols: **Hill_Small.png**, **Hill_Tall.png**, **Cloud_1.png** i **Cloud_2.png**.

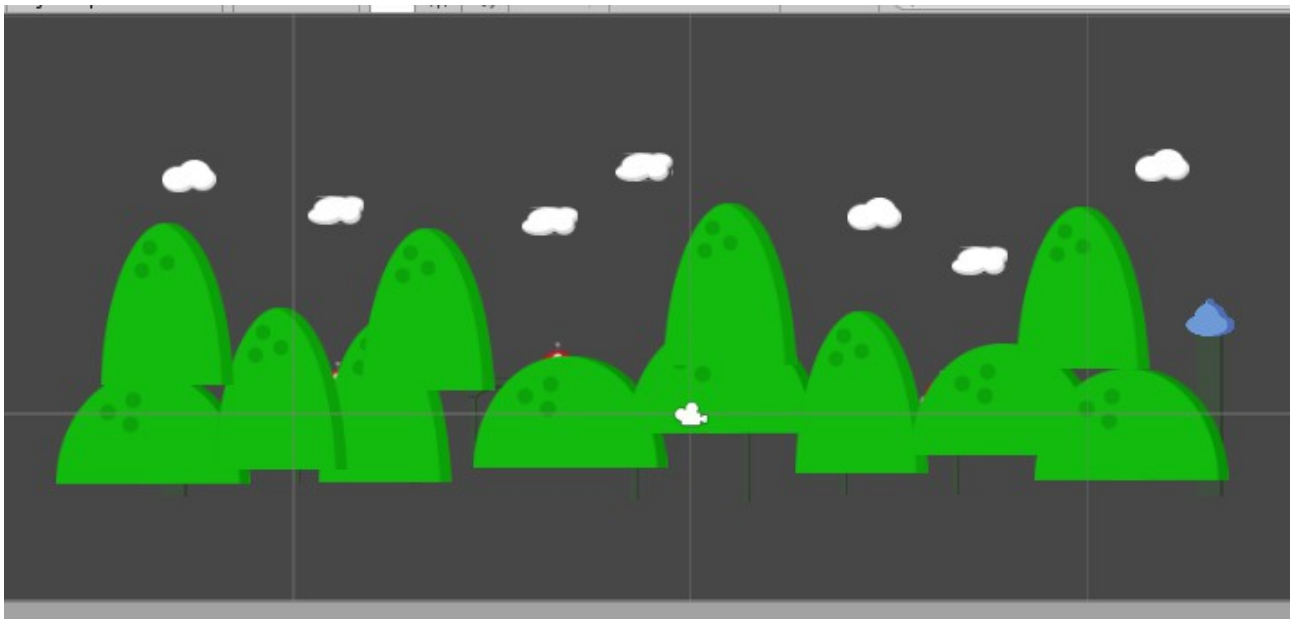
2.1 Crear estructura per definir diferents capes de Parallax.

Els objectes del fons es situaran en diferents Layers, per poder aplicar a cada una d'aquestes capes un moviment adaptat a la seva proximitat a la càmara.

Creem un objecte buit, que anomenem **_ParallaxLayers** i, dintre seu, hi creem les diferents "capes" (també objectes buits): _**CloudLayer**, **_NearHillsLayer** i **_FarHillsLayer**.

Importem a la carpeta **Assets/Textures/ParallaxScenery** els gràfics de les muntanyest i els núvols. Els situem on ens sembli bé, agrupant les muntanyes properes dins de l'objecte _NearHillsLayer, les muntanyes llunyanes dins de _FarHillsLayer i els núvols dins de _CloudLayer.

Encara no tenim el que volem perquè els nous objectes poden solapar-se amb els altres de manera inadequada:



2.2 Ordenar les capes de Parallax.

Per no barrejar incorrectament les imatges, anem a assignar cada objecte a una capa.

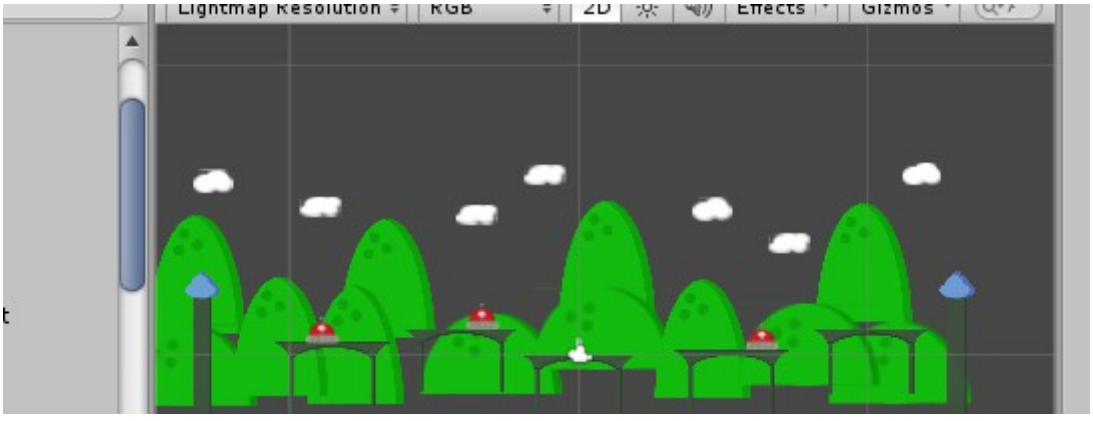
Si seleccionem un objecte qualsevol, veiem en el seu component **SpriteRenderer** l'atribut **Order in Layer**. De moment, tots tenen el mateix valor, 0.

Anem a canviar aquest valors agrupant els objectes, de la següent manera:

1. Núvols, Order in Layer 0.
2. Muntanyes properes, Order in Layer 1 i Muntanyes llunyanes, Order in Layer 0.
3. Prefab Platform, Order in Layer 2.
4. Prefab Enemy i Player Bullet, Order in Layer 2.
5. Player, Order in Layer 2.
6. Wall, Order in Layer 3.

Es pot canviar l'atribut **Order in Layer** d'un grup d'objectes, es poden seleccionar a la pestanya Hierarchy i canviar-los tots de cop.

Ara sí que hem aconseguit el que voliem:



2.3 Crear script ParallaxController

Aquest codi gestiona el moviment diferent de les capes.

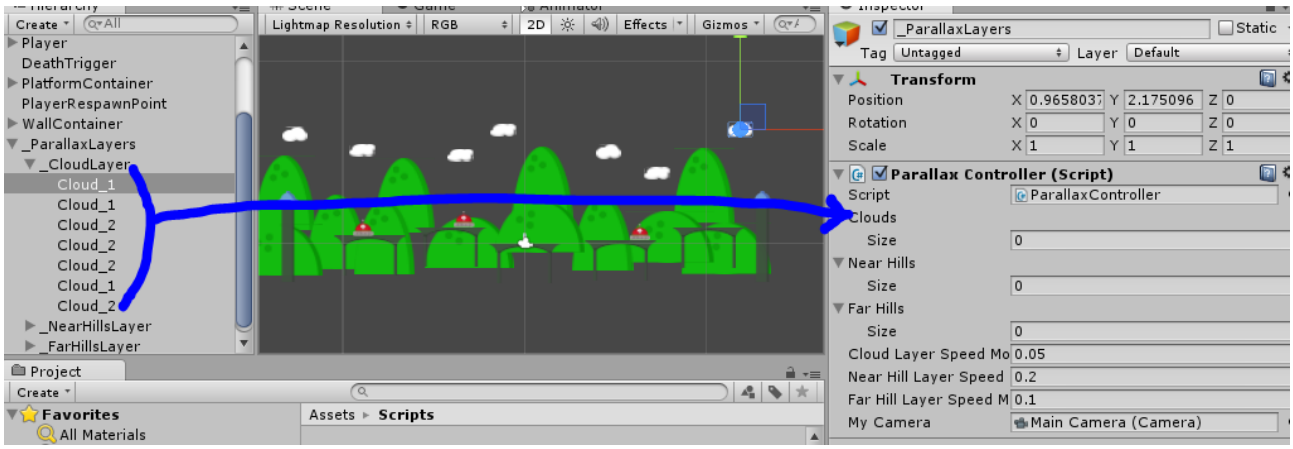
```
using UnityEngine;
using System.Collections;

public class ParallaxController : MonoBehaviour
{
   public GameObject[] clouds;
   public GameObject[] nearHills;
   public GameObject[] farHills;
   public float cloudLayerSpeedModifier;
   public float nearHillLayerSpeedModifier;
   public float farHillLayerSpeedModifier;
   public Camera myCamera;
   private Vector3 lastCamPos;

   void Start()
   {
      lastCamPos = myCamera.transform.position;
   }

   void Update()
   {
      Vector3 currCamPos = myCamera.transform.position;
      float xPosDiff = lastCamPos.x - currCamPos.x;

      adjustParallaxPositionsForArray(clouds,
                                    cloudLayerSpeedModifier, xPosDiff);
      adjustParallaxPositionsForArray(nearHills,
                                    nearHillLayerSpeedModifier, xPosDiff);
      adjustParallaxPositionsForArray(farHills,
                                    farHillLayerSpeedModifier, xPosDiff);
      lastCamPos = myCamera.transform.position;
   }

   void adjustParallaxPositionsForArray(GameObject[] layerArray,
                                    float layerSpeedModifier, float xPosDiff)
   {
      for(int i = 0; i < layerArray.Length; i++)
      {
            Vector3 objPos = layerArray[i].transform.position;
            objPos.x += xPosDiff * layerSpeedModifier;
            layerArray[i].transform.position = objPos;
      }
   }
}
```

Aquest script s'ha d'associar a l'objecte **_ParallaxLayers**. Aleshores, des del Object Inspector anem a assignar valors als atributs públics de l'script:

Assignem l'objecte **Main Camera** a l'atribut **My Camera**.

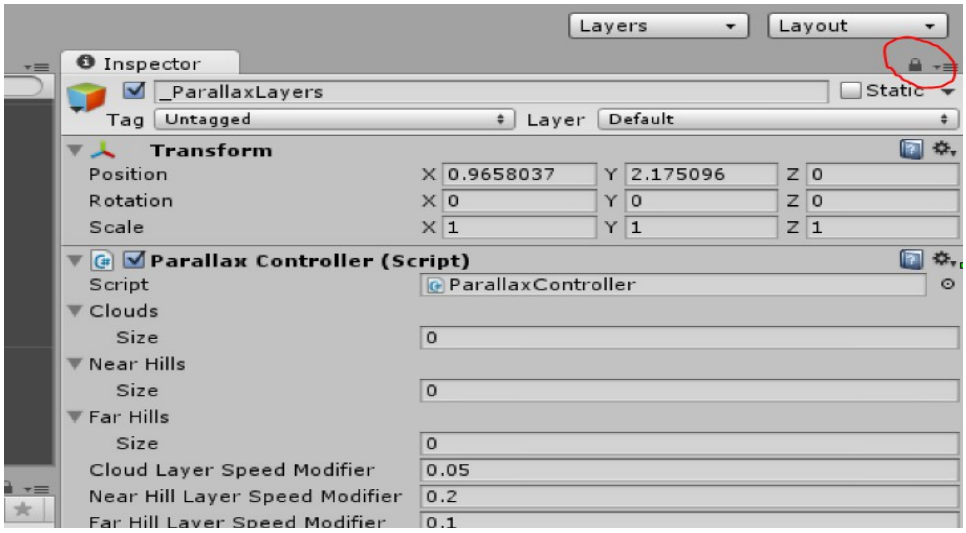
Als altres tres, els **LayerSpeedModifier**, els assignem un valor numèric adequat, sabent que com més gran sigui, més ràpid es mourà quan es desplaci la càmera.
Provem amb
nearHillLayerSpeedModifier = 0,2
farHillLayerSpeedModifier = 0,1 i
cloudLayerSpeedModifier = 0,05

Ara només cal afegir a l'array corresponent de l'script els objectes que hem situat a cada capa. Comencem amb els núvols:



Es tracta simplement d'arrossegar cada núvol a l'array de l'script.

ATENCIÓ: si no fem res addicional, quan seleccionem un núvol per arrossegar-lo, l'Object Inspector mostrarà informació sobre el núvol i no sobre **_CloudLayer**, que és el que volem. Per això hem de bloquejar les dades de **_CloudLayer** a l'Object Inspector, clicant la icona del cadenat:



Fem el mateix amb les muntanyes de cada capa. **En acabar** d'arrossegar els objectes als arrays s'ha de tenir la precaució de **desbloquejar l'Object Inspector**.

EXPERIMENT

Executar el joc tal com està en aquest moment, amb els últims canvis.

Comprovar que el paisatge es mou segons les capes que hem definit. Experimentar amb diferents valors per canviar la velocitat.

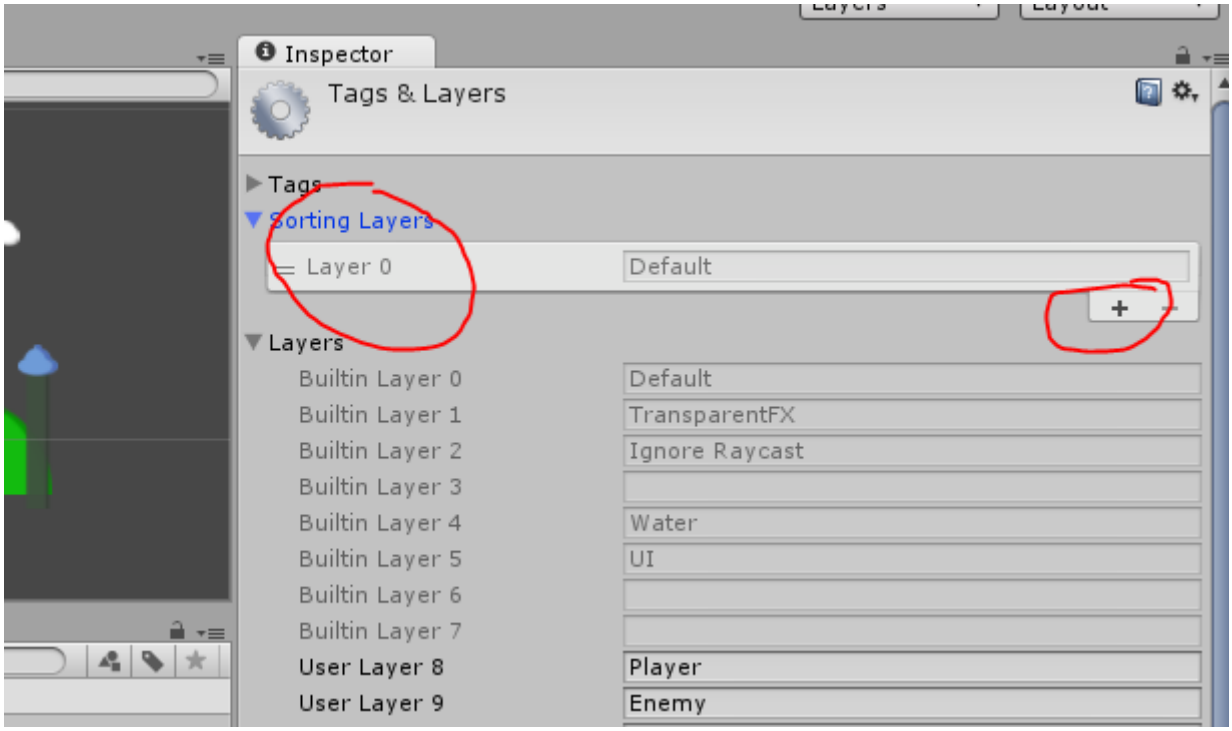
Veure què passa amb el sistema de partícules que apareix en destuir un enemic.

2.4 Últims ajustos.

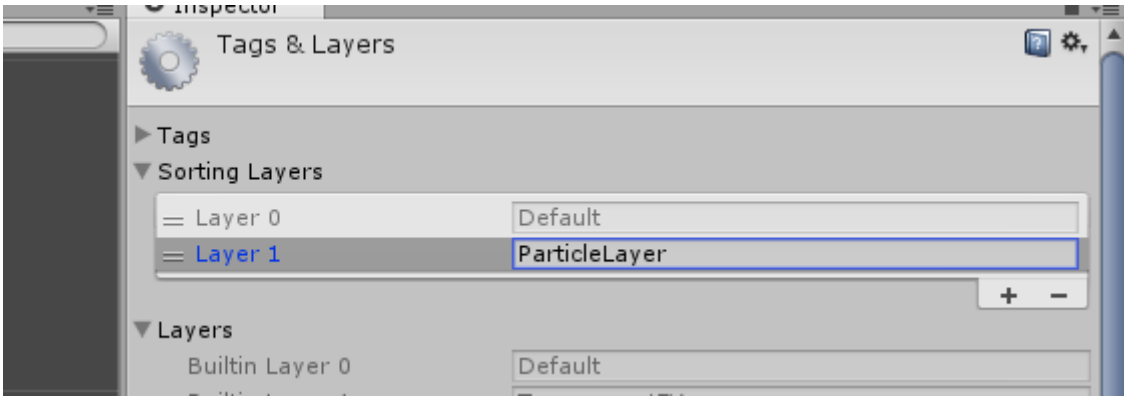
Per defecte, els sprites es renderitzen després que les partícules. Això té com a efecte que les partícules es veuen darrera de tots els sprites de l'escena.

Per arreglar això utilitzarem **Sorting Layers**, per canviar l'ordre de renderitzat.

Anem a la finestra de **Tags & Layers** i expandim el menú desplegable **Sorting Layers**.



Hi afegim un altre Layer (quedarà com a Layer1) que anomenem **ParticleLayer**. Aquest Layer es renderitzarà després del Layer 0 (Default Layer).



Però això per si sol no farà res. Ara cal associar el sistema de partícules al Layer **ParticleLayer**. Ho fem en un nou script que anomenem **ParticleLayering** i que associarem al prefab **EnemyDeathFX**.

```
using UnityEngine;
using System.Collections;

public class ParticleLayering : MonoBehaviour
{
      public string sortLayerString = "";

      void Start ()
      {
         particleSystem.renderer.sortingLayerName = sortLayerString;
      }
}
```

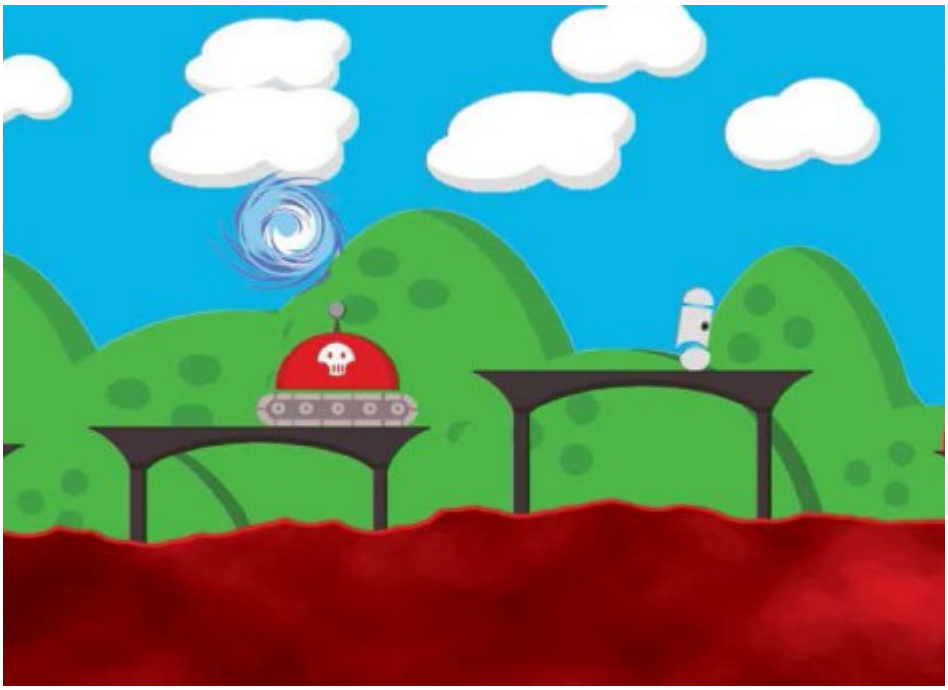
Des de l'object inspector, assignem a **Sort Layer String** el nou layer que hem creat **ParticleLayer**.

EXPERIMENT

Executar el joc tal com està en aquest moment, amb els últims canvis.

Veure què passa amb el sistema de partícules que apareix en destuir un enemic.

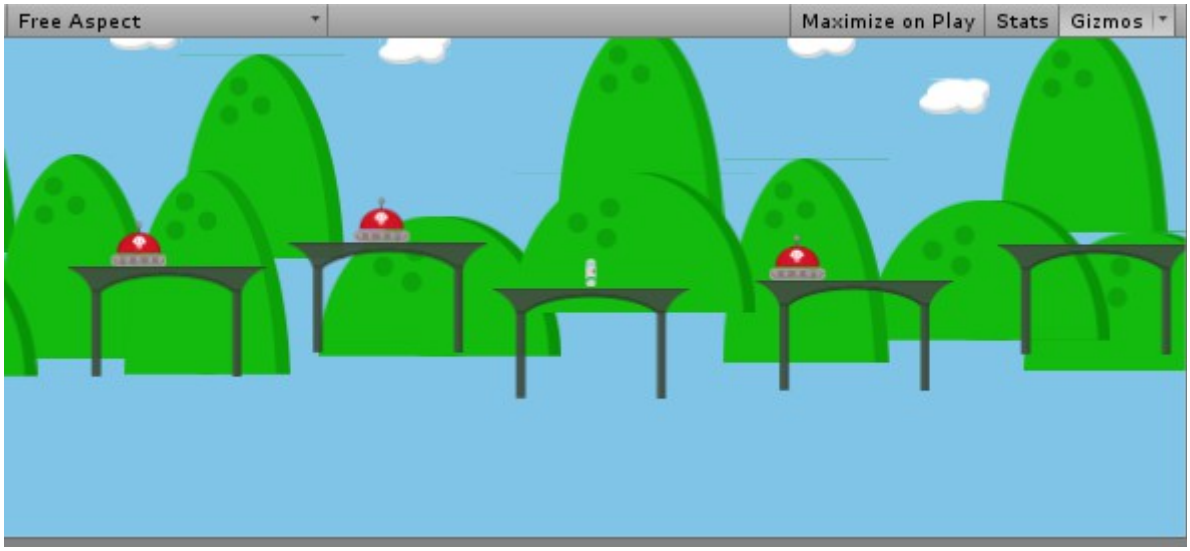
L'aspecte que ha de tenir l'escena del joc és aquest:



Per donar color blau al cel podem situar un objecte al fons o, de manera més senzilla, canviem el color de fons de la càmera:

Seleccionem l'objecte **Main Camera** i, a l'object inspector, editem el seu atribut **Background**. Provem amb els valors R:128, G:197, B:232 i A:0.

Ja hem fet la meitat de la feina:



EXPERIMENT

Executar el joc tal com està en aquest moment, amb els últims canvis.

Activitats

1. Acabar de construir l'escena. Utilitzar l'sprite **Lava.png** per crear una nova capa a **_ParallaxLayers** que es comporti adequadament. Afegir també **Sun.png** a la capa dels núvols.

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