Programa Educativo: Entornos Virtuales y Negocios Digitales

Materia: Programación para Videojuegos I

Tutorial 03: Sonidos y Efectos

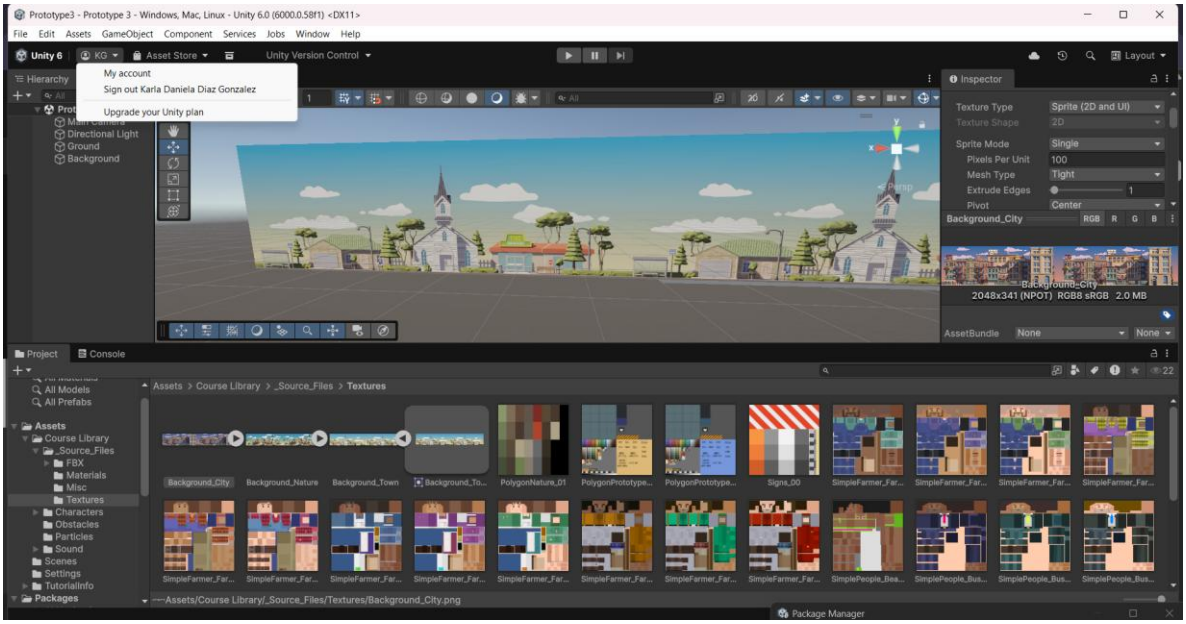
Alumna: Karla Daniela Diaz González

No Control: 12223100814

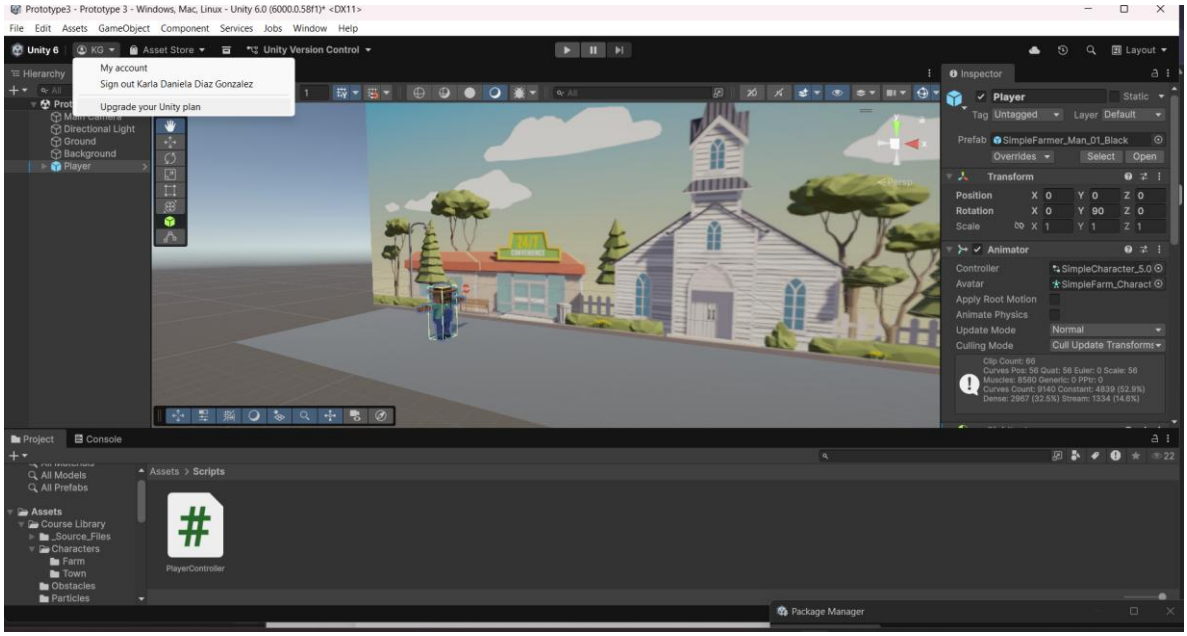
Grupo: GIEV3071

Profesor: Gabriel Barrón

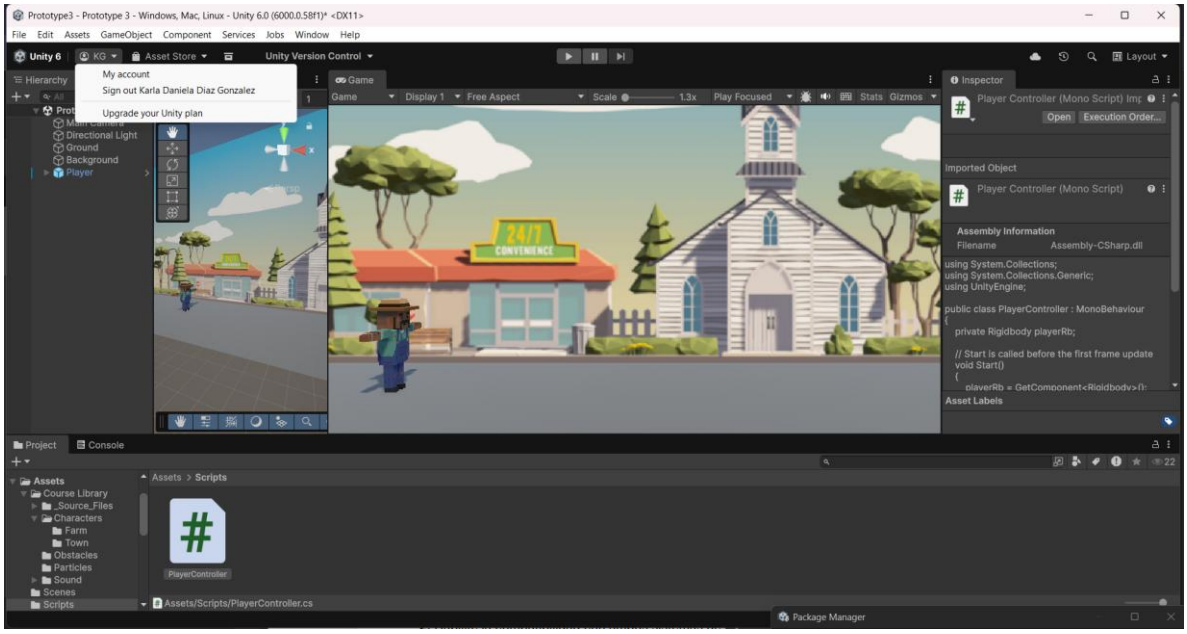
- Abra Unity Hub y cree un proyecto vacío " Prototype 3 " en el directorio de su curso con la versión correcta de Unity.
  Si no recuerda cómo hacerlo, consulte las instrucciones de la Lección 1.1 - Paso 1.
- Haga clic para descargar los archivos de inicio del Prototipo 3 , extraiga la carpeta comprimida e importe el paquete .unity a su proyecto.
  Si no recuerda cómo hacerlo, consulte las instrucciones de la Lección 1.1 - Paso 2.
- Abra la escena Prototipo 3 y elimine la Escena de muestra sin guardar
- Seleccione el objeto Fondo en la jerarquía, luego en el componente Sprite Renderer > Sprite, seleccione la imagen _City, _Nature o _Town

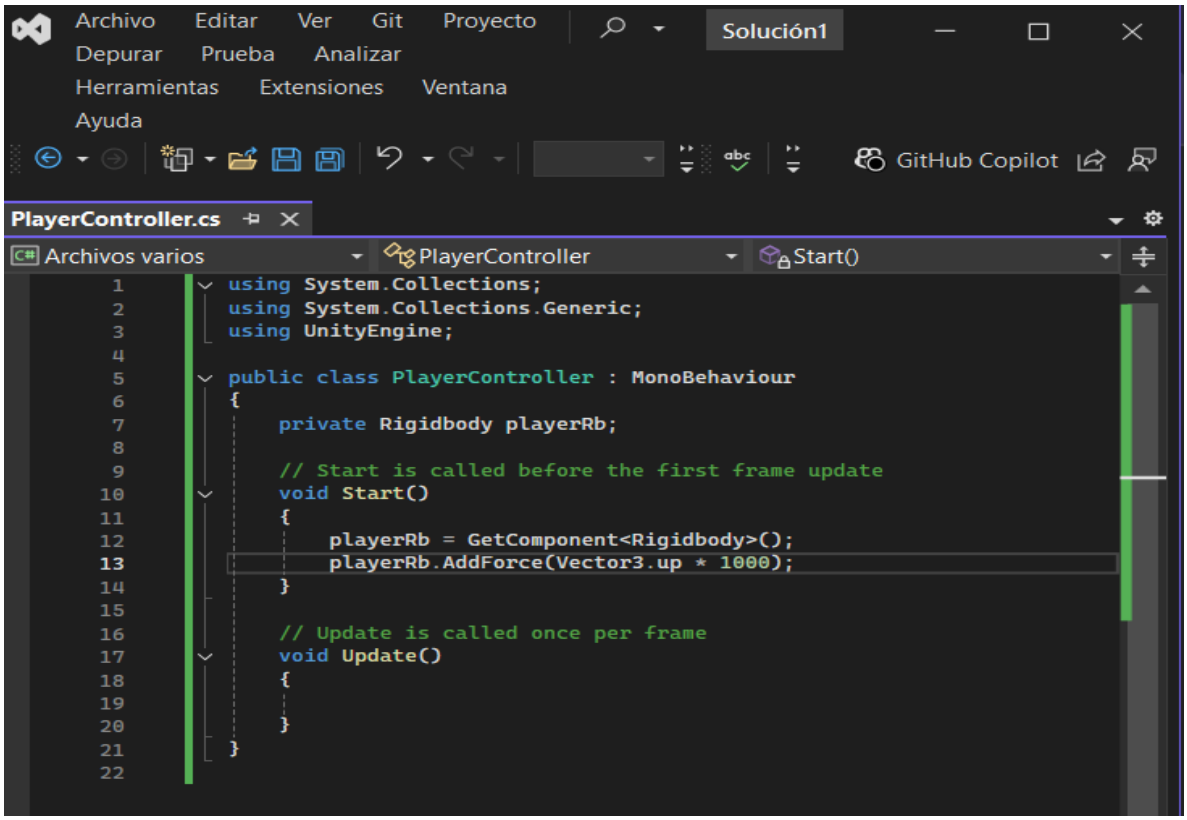


- Desde *Biblioteca de cursos* > *Personajes*, arrastre un personaje a la jerarquía, cámbiele el nombre a “Jugador” y luego rótelo en el eje Y para que mire hacia la derecha.
- Agregar un componente de cuerpo rígido
- Agregue un colisionador de cajas y luego edite los límites del colisionador
- Cree una nueva carpeta “ Scripts ” en Assets, cree un script “ PlayerController ” dentro y adjúntelo al reproductor.



- En PlayerController.cs , declare una nueva variable *privada Rigidbody playerRb;*
- En *Start()* , inicialice *playerRb = GetComponent<Rigidbody>();*
- En *Start()* , use el método AddForce para hacer que el jugador salte al comienzo del juego.



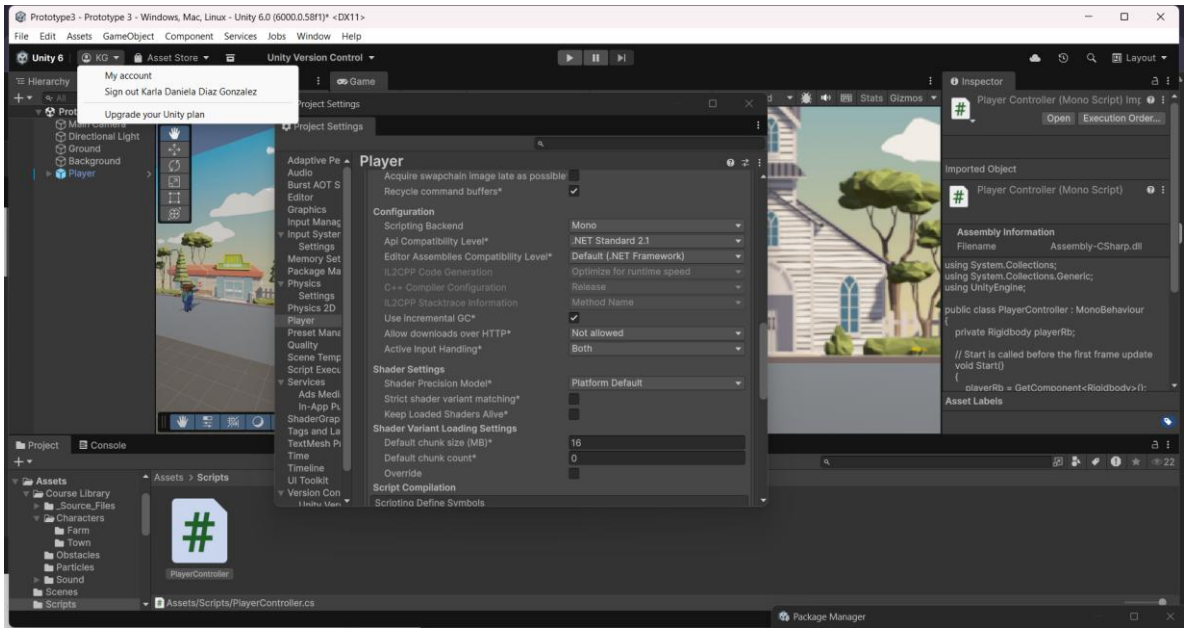


1. Abra la configuración del reproductor :

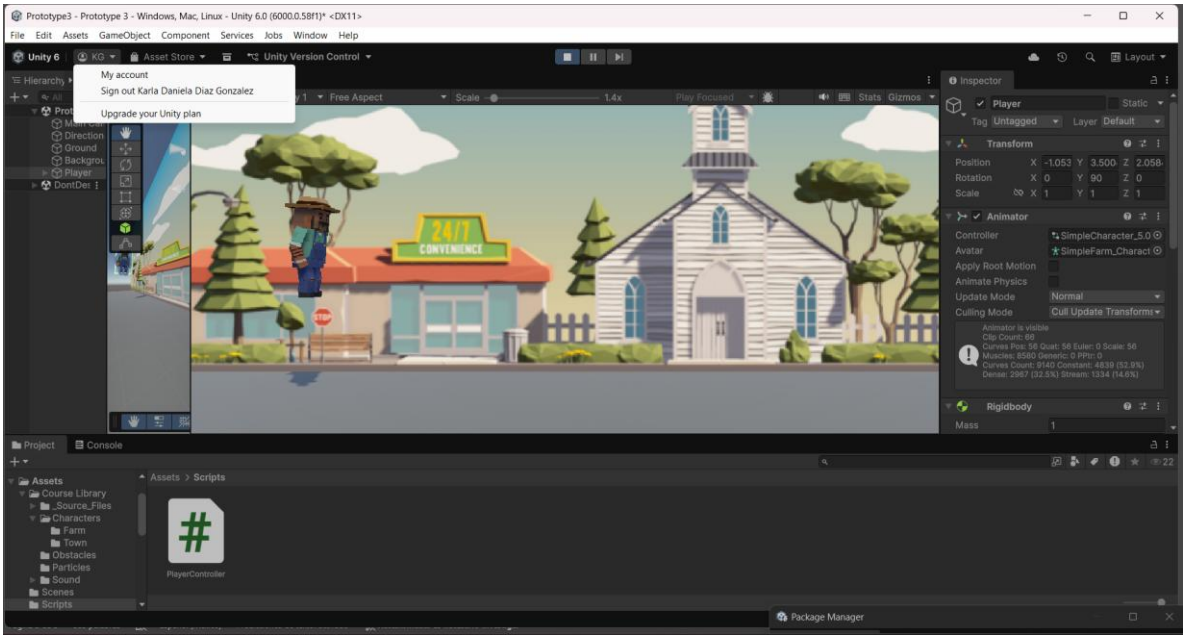
- Desde el menú principal, seleccione Editar > Configuración del proyecto, luego seleccione la categoría Reproductor en el panel más a la izquierda.

2. Habilite la compatibilidad con ambos sistemas de entrada:

- Encuentra la sección Configuración .
- En el menú desplegable Manejo de entrada activa , seleccione Ambos .
- Seleccione el botón Aplicar para confirmar el cambio.
- El Editor de Unity se reiniciará automáticamente para aplicar esta configuración. Tras reiniciarse, el proyecto estará configurado correctamente para gestionar la entrada tanto del Gestor de Entrada original como del nuevo Sistema de Entrada.



- En Update() agregue una declaración if-then que verifique si se presiona la barra espaciadora
- Corte y pegue el código AddForce de Start() en la declaración if
- Agregue el parámetro *ForceMode.Impulse* a la llamada *AddForce* , luego reduzca el valor del multiplicador de fuerza

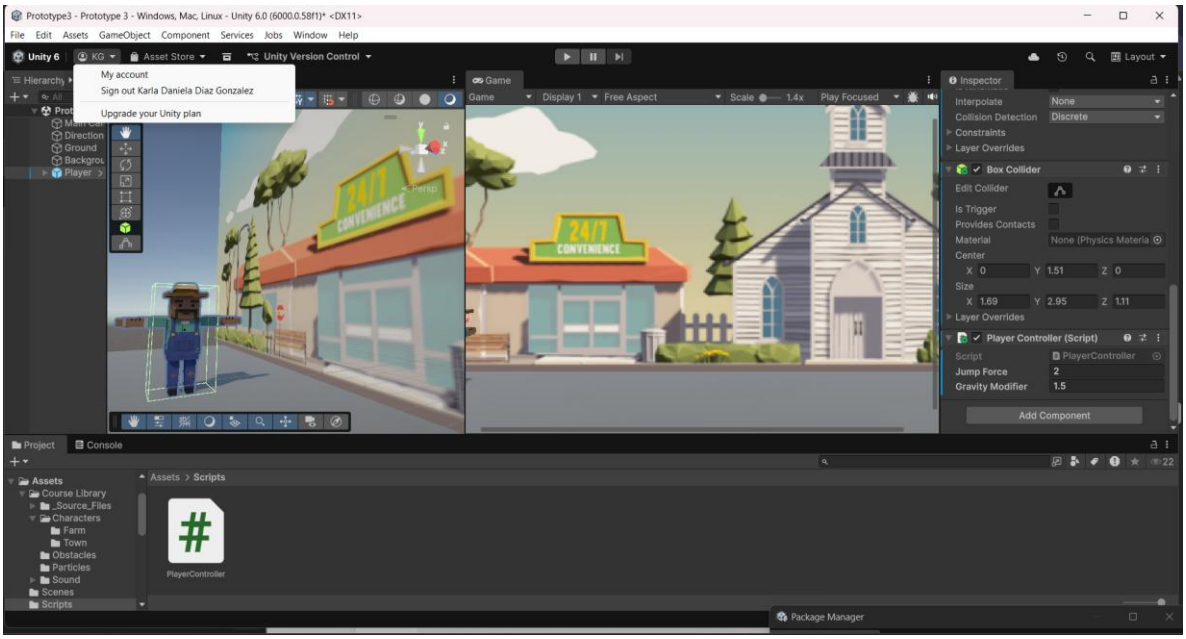


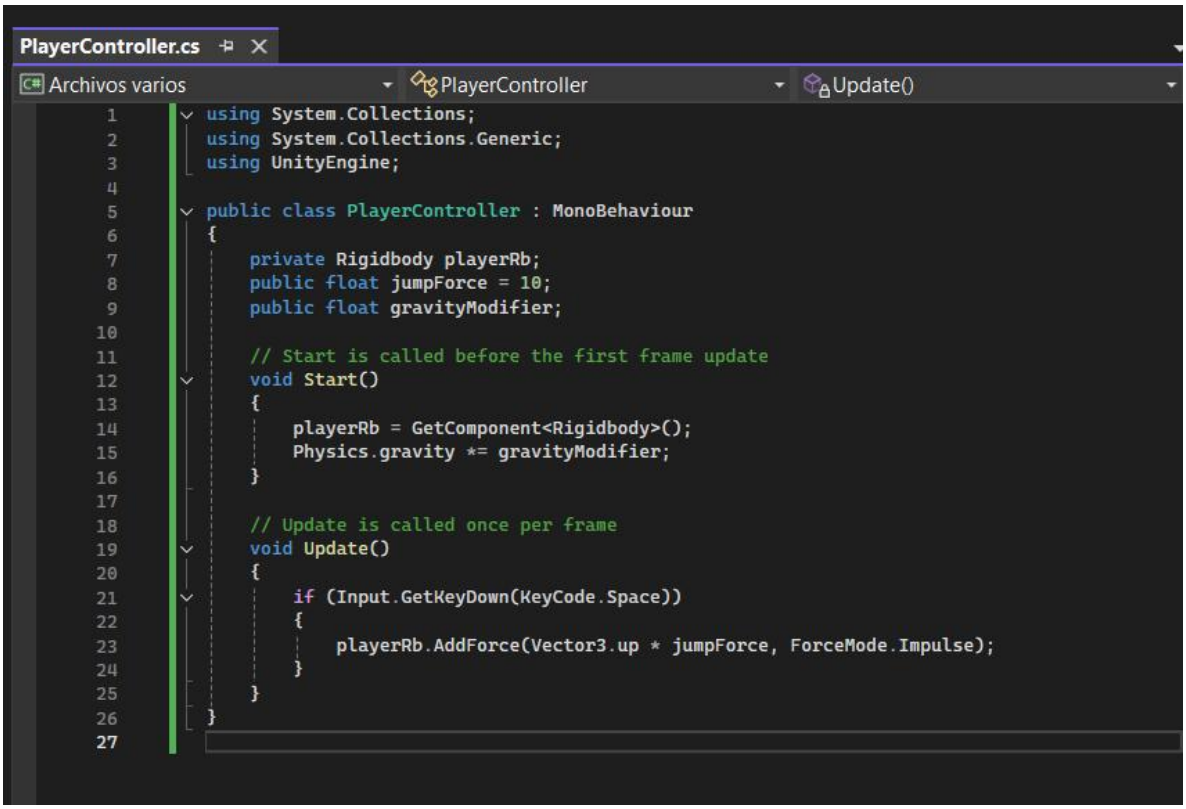
PlayerController.cs

Archivos varios | PlayerController | Update()

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerController : MonoBehaviour
6  {
7      private Rigidbody playerRb;
8
9      // Start is called before the first frame update
10     void Start()
11     {
12         playerRb = GetComponent<Rigidbody>();
13     }
14
15     // Update is called once per frame
16     void Update()
17     {
18         if (Input.GetKeyDown(KeyCode.Space))
19         {
20             playerRb.AddForce(Vector3.up * 10, ForceMode.Impulse);
21         }
22     }
23 }
24
```

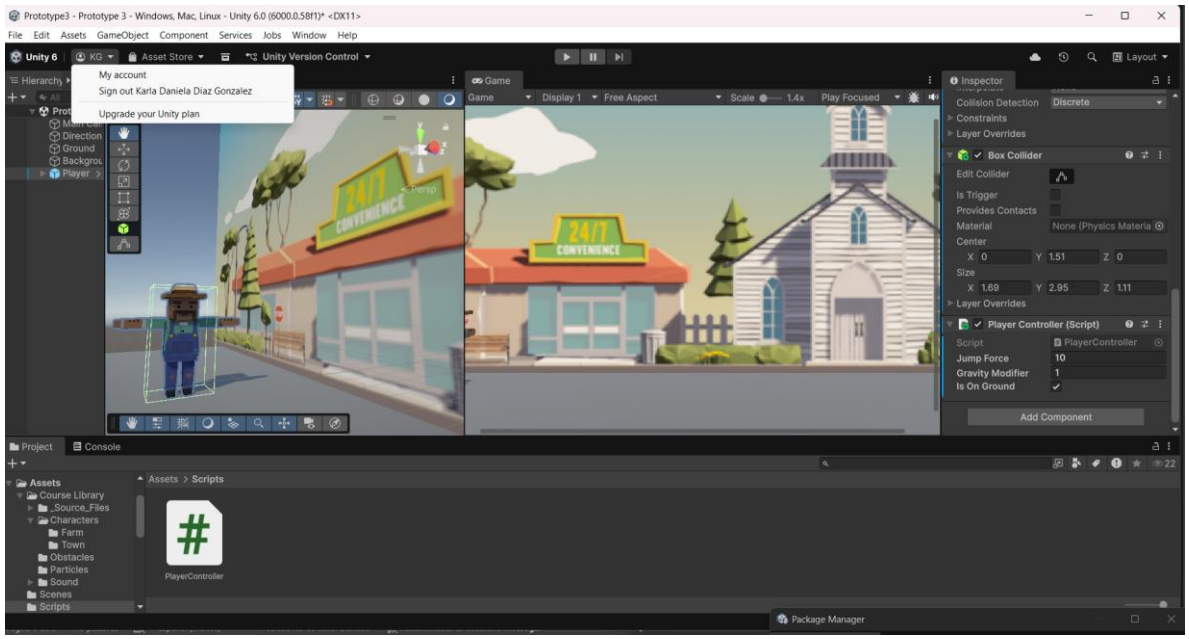
- Replace the hardcoded value with a new public float jumpForce variable
- Add a new public float gravityModifier variable and in Start(), add Physics.gravity *= gravityModifier;
- In the inspector, tweak the gravityModifier, jumpForce, and Ri





- Agregue una nueva variable pública bool isOnGround y configúrela como verdadera
- En la declaración if que hace que el jugador salte, establezca isOnGround = false y luego pruebe

- Agregue una condición && isOnGround a la declaración if
- Agregue un nuevo método void OnCollisionEnter , establezca isOnGround = true en ese método y luego pruebe

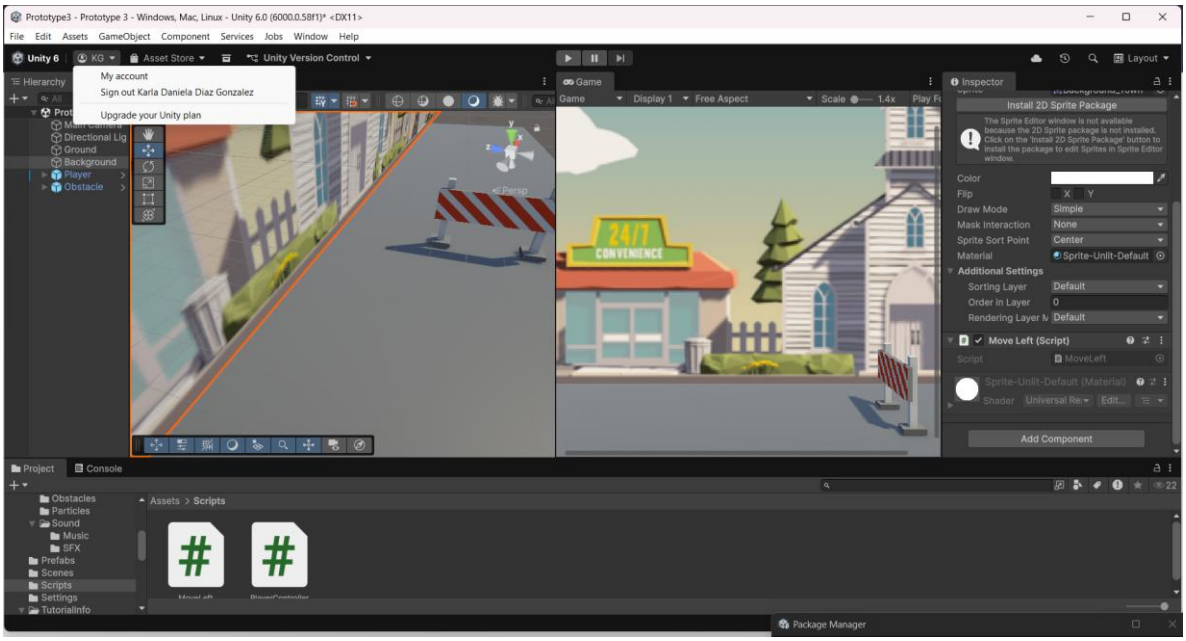


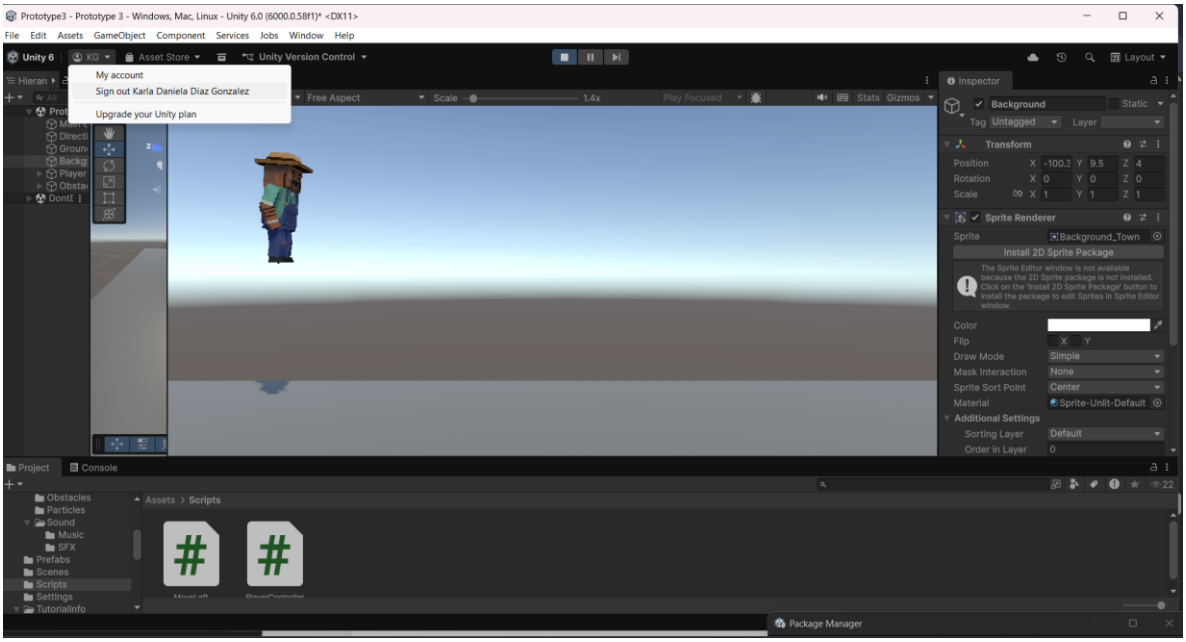
PlayerController.cs

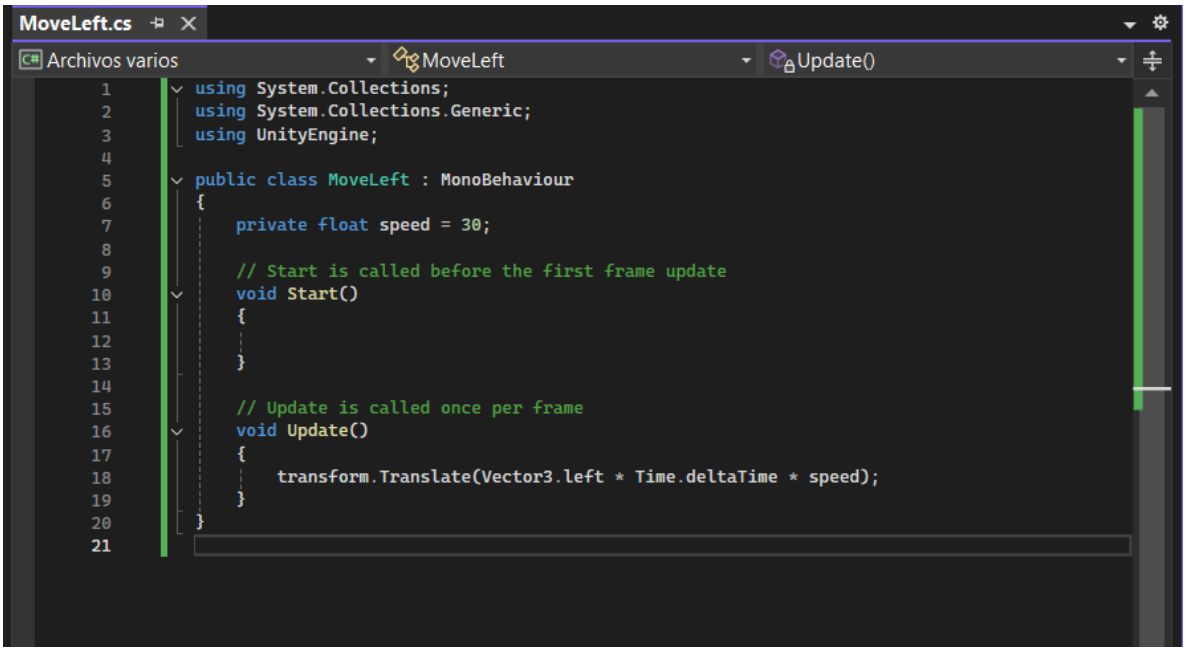
Archivos varios | PlayerController | Update()

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    private Rigidbody playerRb;
    public float jumpForce = 10;
    public float gravityModifier;
    public bool isOnGround = true;

    // Start is called before the first frame update
    void Start()
    {
        playerRb = GetComponent<Rigidbody>();
        Physics.gravity *= gravityModifier;
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space) && isOnGround)
        {
            playerRb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
            isOnGround = false;
        }
    }

    private void OnCollisionEnter(Collision collision)
    {
        isOnGround = true;
    }
}
```

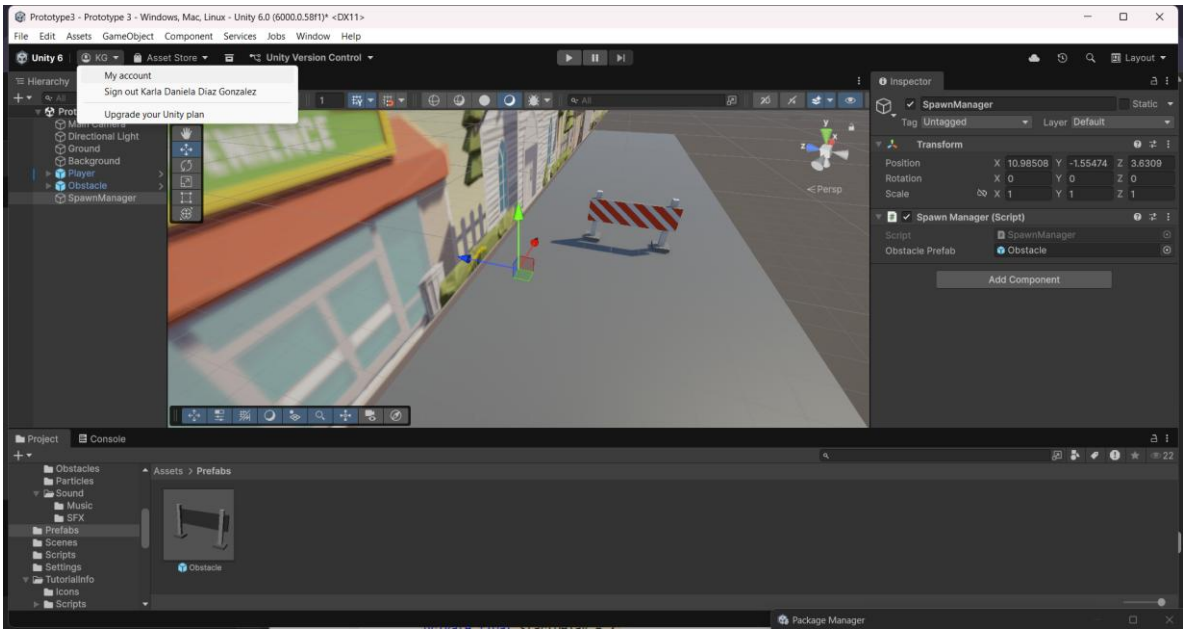
- Desde Biblioteca de cursos > Obstáculos , agrega un obstáculo, renómbralo “ Obstáculo ” y colócalo donde debería aparecer.
- Aplique un componente de cuerpo rígido y colisionador de caja , luego edite los límites del colisionador para que se ajusten al obstáculo
- Cree una nueva carpeta “ Prefabs ” y arrástrela para crear un nuevo Prefab original
- Crea un nuevo script “ MoveLeft ”, aplícalo al obstáculo y ábrelo
- En MoveLeft.cs, escribe el código para trasladarlo a la izquierda según la variable de velocidad
- Aplicar el script MoveLeft al fondo

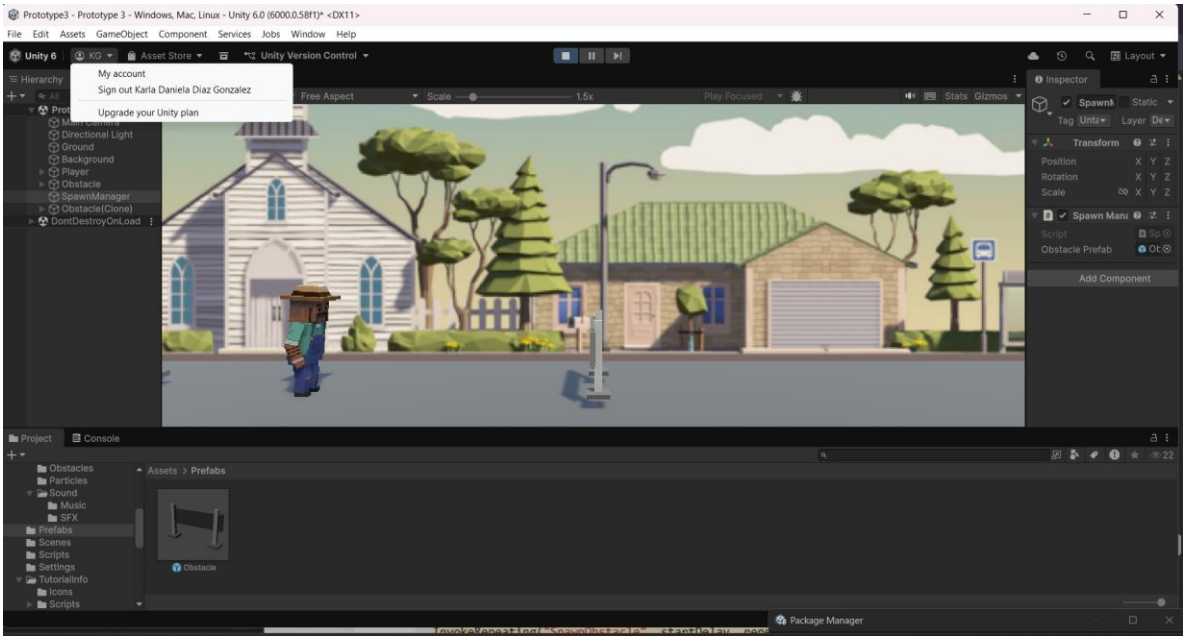

Prototype3 - Prototype 3 - Windows, Mac, Linux - Unity 6.0 (6000.0.58f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
KG
Asset Store
Unity Version Control
Layout
My account
Sign out Karla Daniela Diaz Gonzalez
Upgrade your Unity plan
Hierarchy
Game
Display 1
Free Aspect
Scale
1.4x
Inspector
Install 2D Sprite Package
Color
Flip
Draw Mode
Simple
Mask Interaction
None
Sprite Sort Point
Center
Material
Sprite-Unlit-Default
Additional Settings
Sorting Layer
Default
Order in Layer
Move Left (Script)
Script
MoveLeft
Add Component
Project
Console
Assets > Scripts
Package Manager


Prototype3 - Prototype 3 - Windows, Mac, Linux - Unity 6.0 (6000.0.58f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
KG
Asset Store
Unity Version Control
Layout
My account
Sign out Karla Daniela Diaz Gonzalez
Upgrade your Unity plan
Free Aspect
Scale
1.4x
Play Focused
Stats
Gizmos
Inspector
Background
Static
Tag
Untagged
Layer
Transform
Position
Rotation
Scale
Sprite Renderer
Sprite
Background_Town
Install 2D Sprite Package
Color
Flip
Draw Mode
Simple
Mask Interaction
None
Sprite Sort Point
Center
Material
Sprite-Unlit-Default
Additional Settings
Sorting Layer
Default
Order in Layer
Project
Console
Assets > Scripts
Package Manager



- Cree un nuevo objeto vacío “ Spawn Manager ” y luego aplíquele un nuevo script SpawnManager.cs
- En SpawnManager.cs , declare un nuevo GameObject público obstáculoPrefab; y luego asigne su prefab a la nueva variable en el inspector
- Declara un nuevo spawnPos privado de Vector3 en tu ubicación de aparición
- En Start() , cree una instancia de un nuevo prefabricado de obstáculo, luego elimine su prefabricado de la escena y pruebe


Prototype3 - Prototype 3 - Windows, Mac, Linux - Unity 6.0 (6000.0.58f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
KG
Asset Store
Unity Version Control
My account
Sign out Karla Daniela Diaz Gonzalez
Upgrade your Unity plan
Layout
Hierarchy
Directional Light
Ground
Background
Player
Obstacle
SpawnManager
Persp
Inspector
SpawnManager
Static
Tag Untagged
Layer Default
Transform
Position X 10.98508 Y -1.55474 Z 3.6309
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Spawn Manager (Script)
Script SpawnManager
Obstacle Prefab Obstacle
Add Component
Project
Console
Obstacles
Particles
Sound
Music
SFX
Prefabs
Scenes
Scripts
Settings
TutorialInfo
Icons
Scripts
Assets > Prefabs
Obstacle
Package Manager

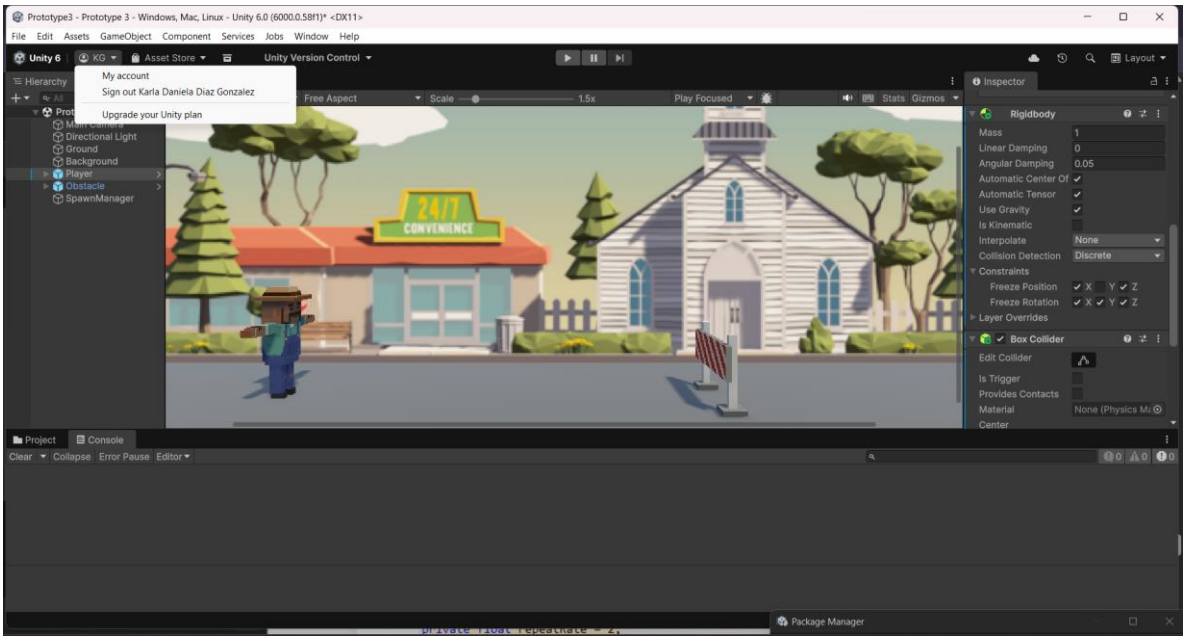

Prototype3 - Prototype 3 - Windows, Mac, Linux - Unity 6.0 (6000.0.58f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
KG
Asset Store
Unity Version Control
My account
Sign out Karla Daniela Diaz Gonzalez
Upgrade your Unity plan
Free Aspect
Scale
1.5x
Play Focused
Stats
Gizmos
Layout
Hierarchy
Directional Light
Ground
Background
Player
Obstacle
SpawnManager
Obstacle(Clone)
DontDestroyOnLoad
Inspector
Transform
Position
Rotation
Scale
Add Component
Project
Console
Obstacles
Particles
Sound
Music
SFX
Prefabs
Scenes
Scripts
Settings
TutorialInfo
Icons
Scripts
Assets > Prefabs
Obstacle
Package Manager

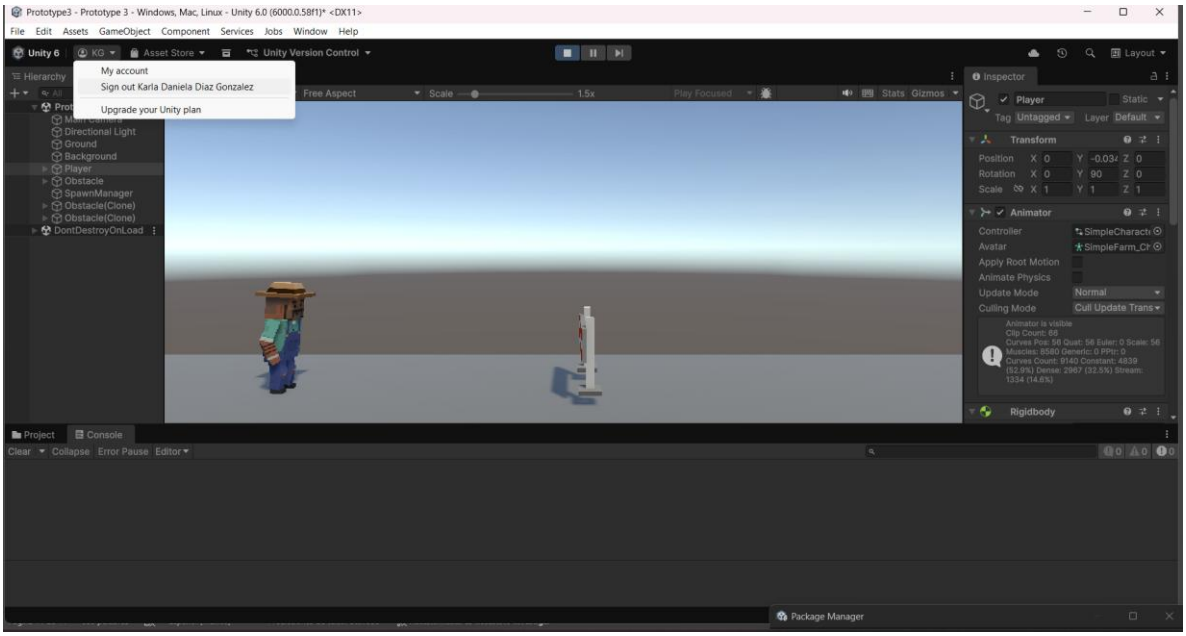
SpawnManager.cs | MoveLeft.cs

Archivos varios | SpawnManager | Update()

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class SpawnManager : MonoBehaviour
6  {
7      public GameObject obstaclePrefab;
8      private Vector3 spawnPos = new Vector3(25, 0, 0);
9
10     // Start is called before the first frame update
11     void Start()
12     {
13         Instantiate(obstaclePrefab, spawnPos, obstaclePrefab.transform.rotation);
14     }
15
16     // Update is called once per frame
17     void Update()
18     {
19
20     }
21 }
22
```

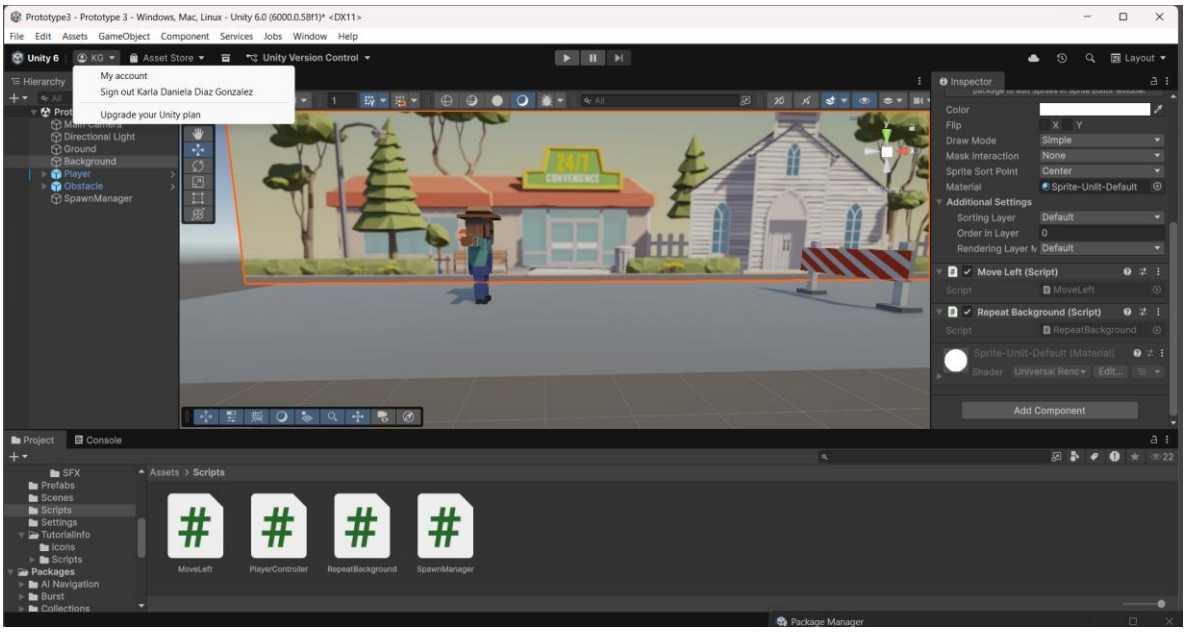
- Cree un nuevo método vacío SpawnObstacle , luego mueva la llamada Instantiate dentro de él
- Crear nuevas variables flotantes para startDelay y repeatRate
- Haz que tus obstáculos se generen en intervalos usando el método InvokeRepeating()
- En el componente Cuerpo rígido del jugador, expanda Restricciones y luego Congele todas las posiciones excepto la Y


Prototype3 - Prototype 3 - Windows, Mac, Linux - Unity 6.0 (6000.0.58f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
KG
Asset Store
Unity Version Control
Layout
My account
Sign out Karla Daniela Diaz Gonzalez
Upgrade your Unity plan
Hierarchy
Directional Light
Ground
Background
Player
Obstacle
SpawnManager
Free Aspect
Scale
1.5x
Play Focused
Stats
Gizmos
24/7
CONVENIENCE
Inspector
Rigidbody
Mass
Linear Damping
Angular Damping
Automatic Center Of
Automatic Tensor
Use Gravity
Is Kinematic
Interpolate
Collision Detection
Constraints
Freeze Position
Freeze Rotation
Layer Overrides
Box Collider
Edit Collider
Is Trigger
Provides Contacts
Material
Center
Project
Console
Clear
Collapse
Error Pause
Editor
Package Manager


Prototype3 - Prototype 3 - Windows, Mac, Linux - Unity 6.0 (6000.0.58f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
KG
Asset Store
Unity Version Control
Layout
My account
Sign out Karla Daniela Diaz Gonzalez
Upgrade your Unity plan
Hierarchy
Directional Light
Ground
Background
Player
Obstacle
SpawnManager
Obstacle(Clone)
Obstacle(Clone)
DontDestroyOnLoad
Free Aspect
Scale
1.5x
Play Focused
Stats
Gizmos
Inspector
Player
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
Animator
Controller
Avatar
Apply Root Motion
Animate Physics
Update Mode
Normal
Culling Mode
Cull Update Trans
Rigidbody
Project
Console
Clear
Collapse
Error Pause
Editor
Package Manager

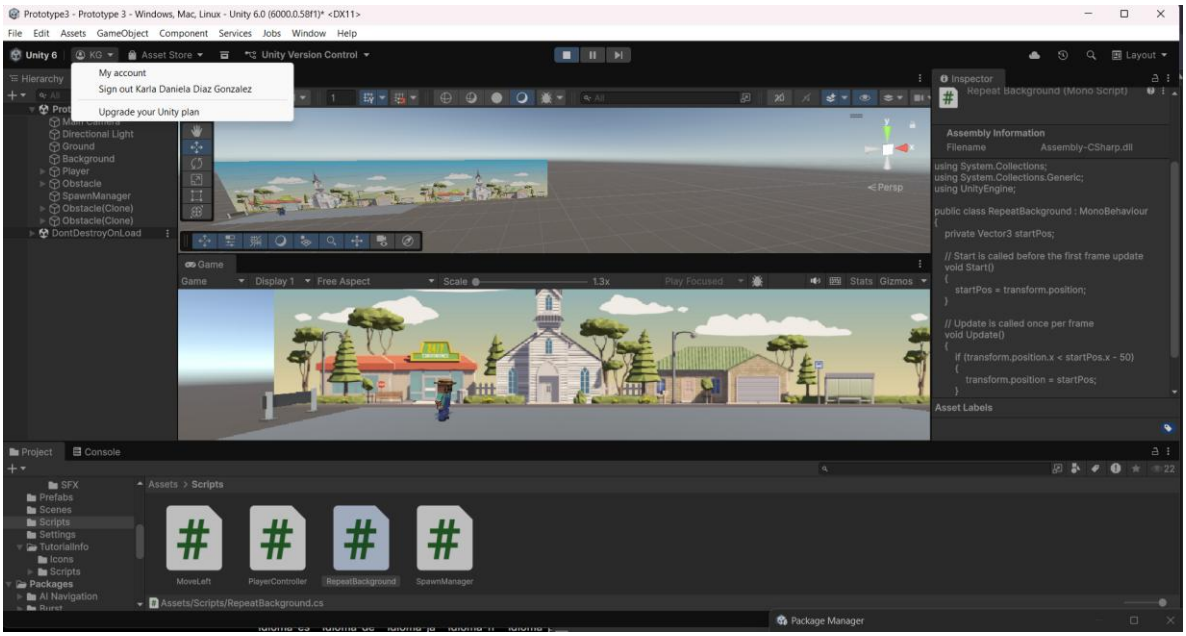
```
SpawnManager.cs | MoveLeft.cs
Archivos varios | SpawnManager | Update()

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class SpawnManager : MonoBehaviour
6  {
7      public GameObject obstaclePrefab;
8      private Vector3 spawnPos = new Vector3(25, 0, 0);
9      private float startDelay = 2;
10     private float repeatRate = 2;
11
12     // Start is called before the first frame update
13     void Start()
14     {
15         InvokeRepeating("SpawnObstacle", startDelay, repeatRate);
16     }
17
18     // Update is called once per frame
19     void Update()
20     {
21
22     }
23
24     void SpawnObstacle()
25     {
26         Instantiate(obstaclePrefab, spawnPos, obstaclePrefab.transform.rotation);
27     }
28 }
29
```

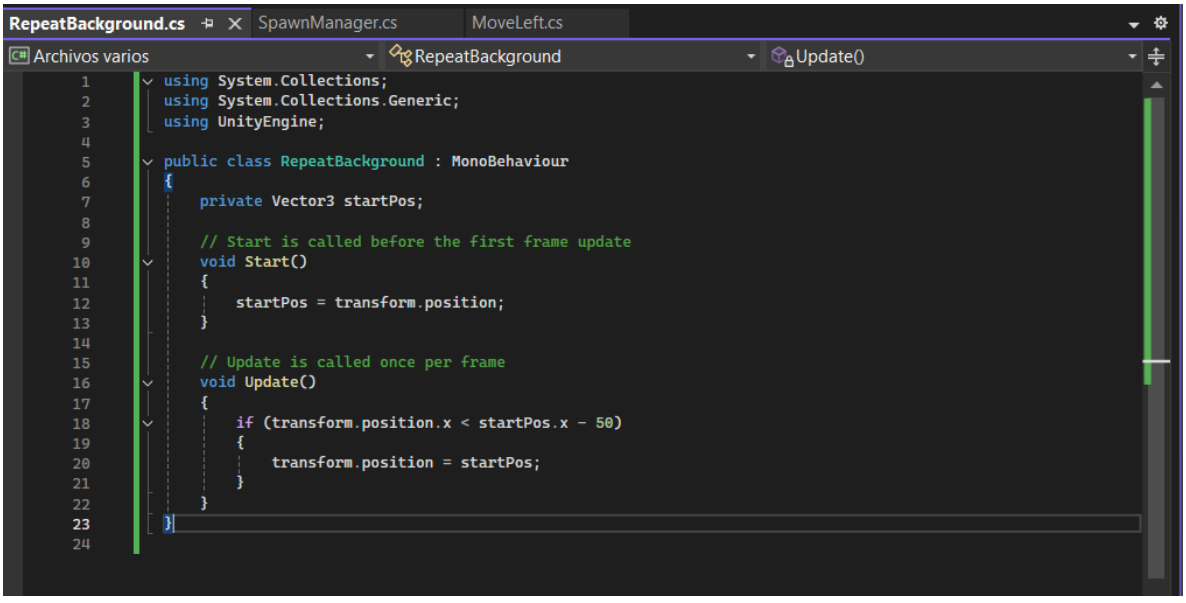
- Cree un nuevo script llamado RepeatBackground.cs y adjúntelo al objeto de fondo



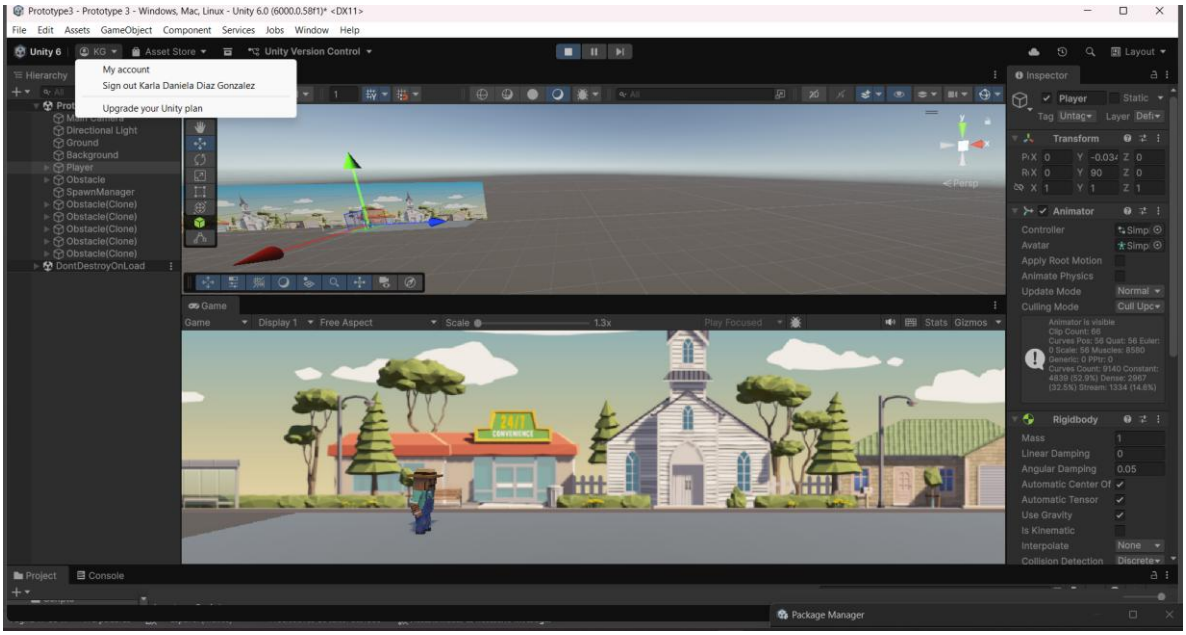
- Declarar una nueva variable privada Vector3 startPos;
- En Start() , establezca la variable startPos en su posición inicial real asignándole = transform.position;

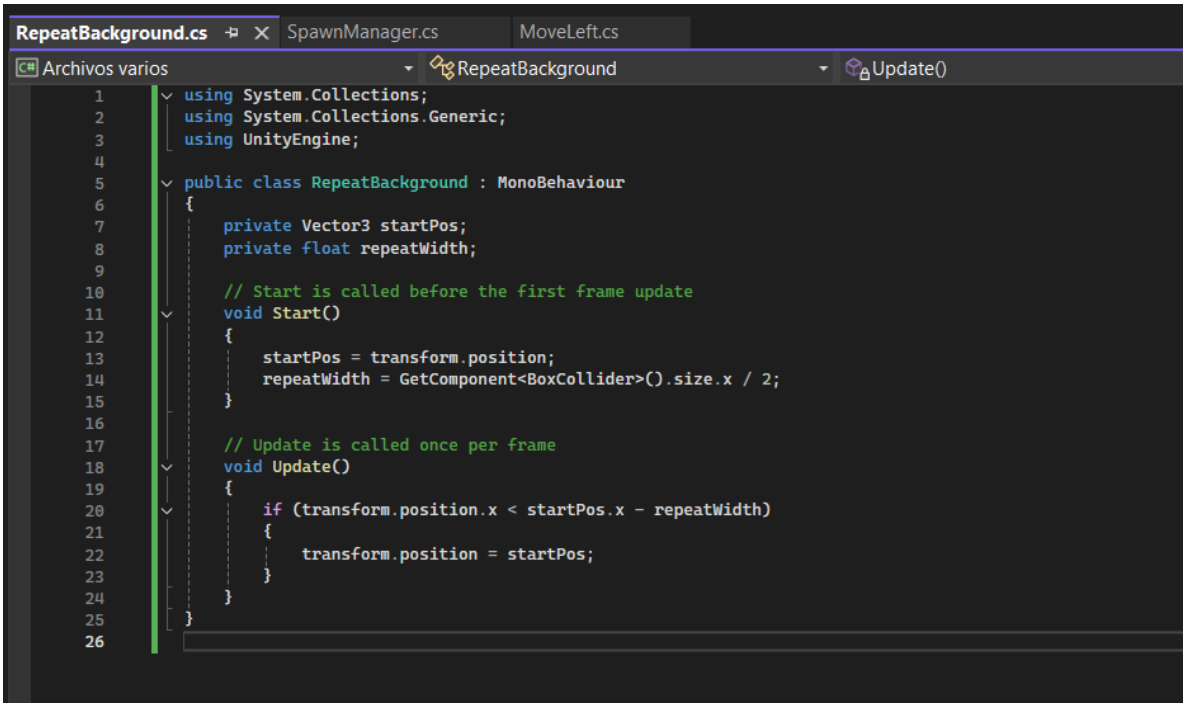
- En Update() , escriba una declaración if para restablecer la posición si se mueve una cierta distancia





- Agregue un componente Box Collider al fondo
- Declarar una nueva variable privada float repeatWidth
- En Start() , obtenga el ancho del colisionador de cajas, dividido por 2
- Incorporar la variable repeatWidth en la función repeat

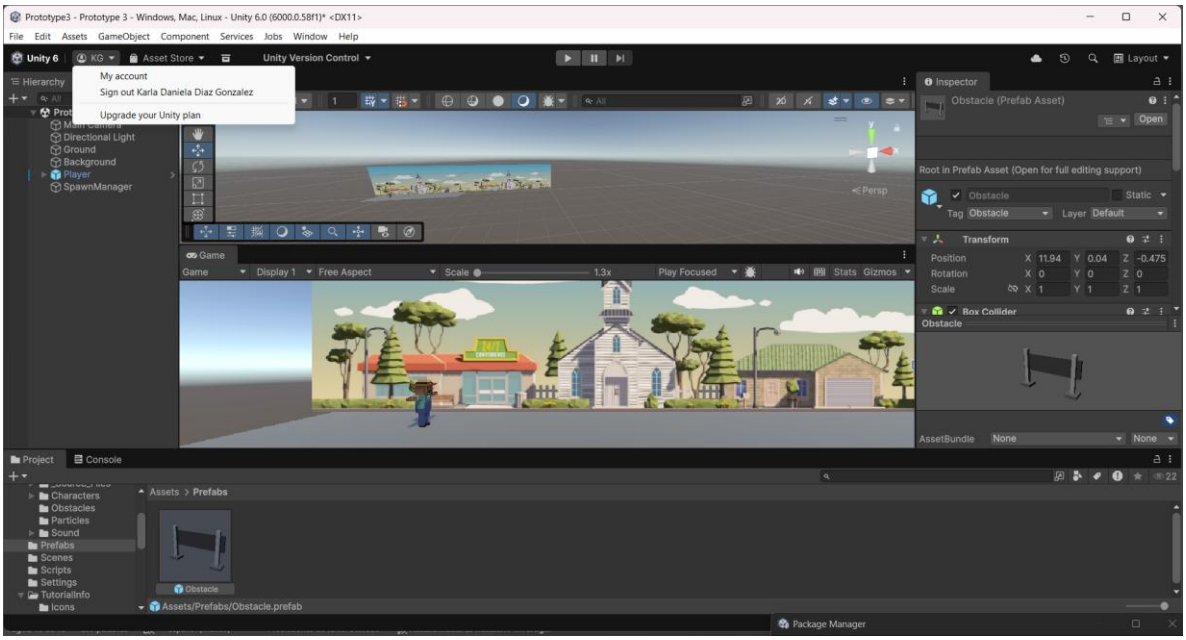


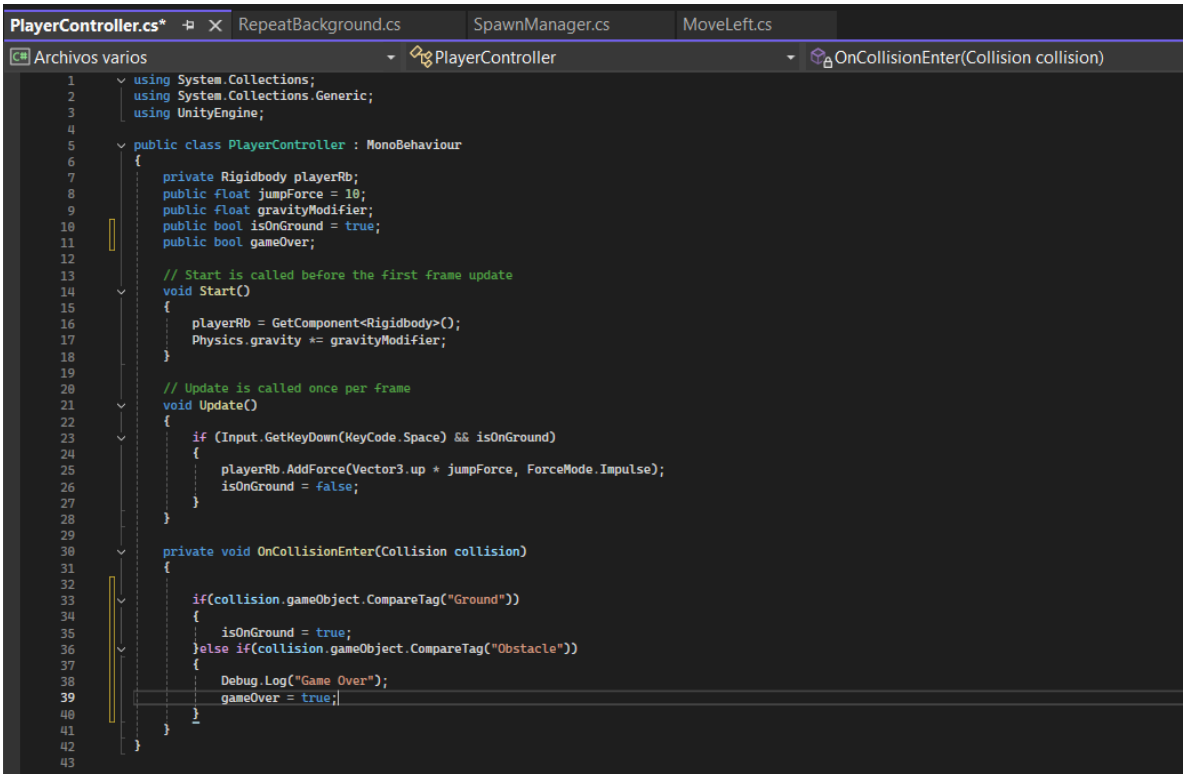


```
RepeatBackground.cs | SpawnManager.cs | MoveLeft.cs
Archivos varios | RepeatBackground | Update()

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class RepeatBackground : MonoBehaviour
6  {
7      private Vector3 startPos;
8      private float repeatWidth;
9
10     // Start is called before the first frame update
11     void Start()
12     {
13         startPos = transform.position;
14         repeatWidth = GetComponent<BoxCollider>().size.x / 2;
15     }
16
17     // Update is called once per frame
18     void Update()
19     {
20         if (transform.position.x < startPos.x - repeatWidth)
21         {
22             transform.position = startPos;
23         }
24     }
25 }
26
```

- En el inspector, agregue una etiqueta “ Suelo ” al prefabricado Suelo y una etiqueta “Obstáculo” al prefabricado Obstáculo
- En PlayerController, declare un nuevo juego público bool gameOver;
- En OnCollisionEnter , agregue la declaración if-else para probar si el jugador chocó con el “Suelo” o un “Obstáculo”.

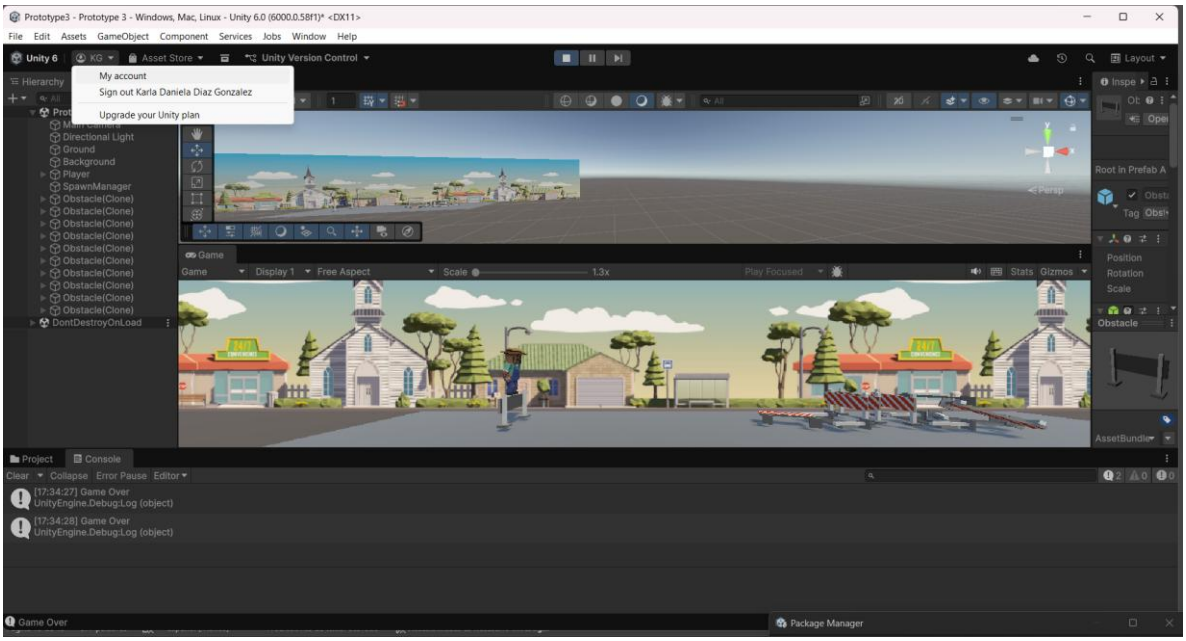
- Si chocan con el “Suelo”, establezca isOnGround = true , y si chocan con un “Obstáculo”, establezca gameOver = true

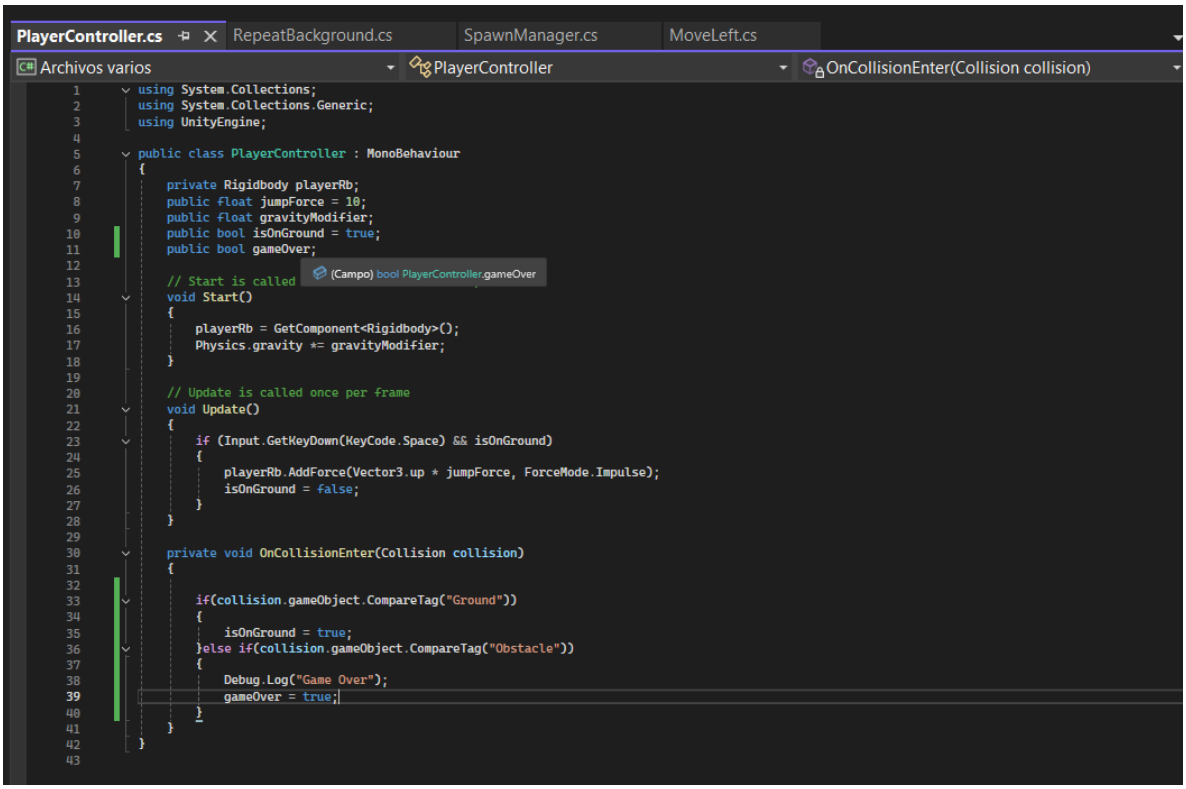




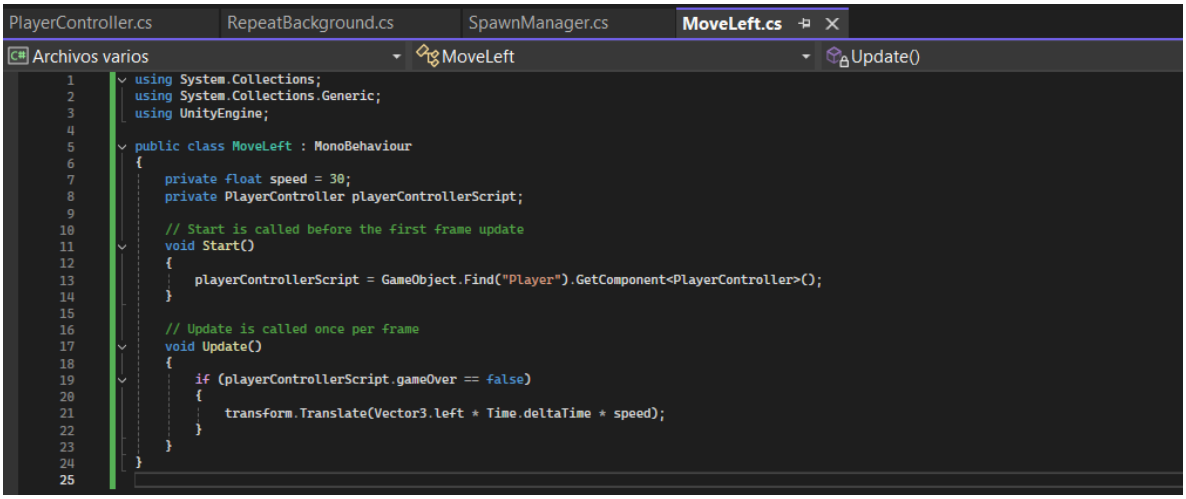
- En MoveLeft.cs , declare un nuevo PlayerController privado playerControllerScript;

- En Start() , inicialícelo encontrando el reproductor y obteniendo el componente PlayerController
- Envuelva el método de traducción en una declaración if que verifique si el juego no ha terminado



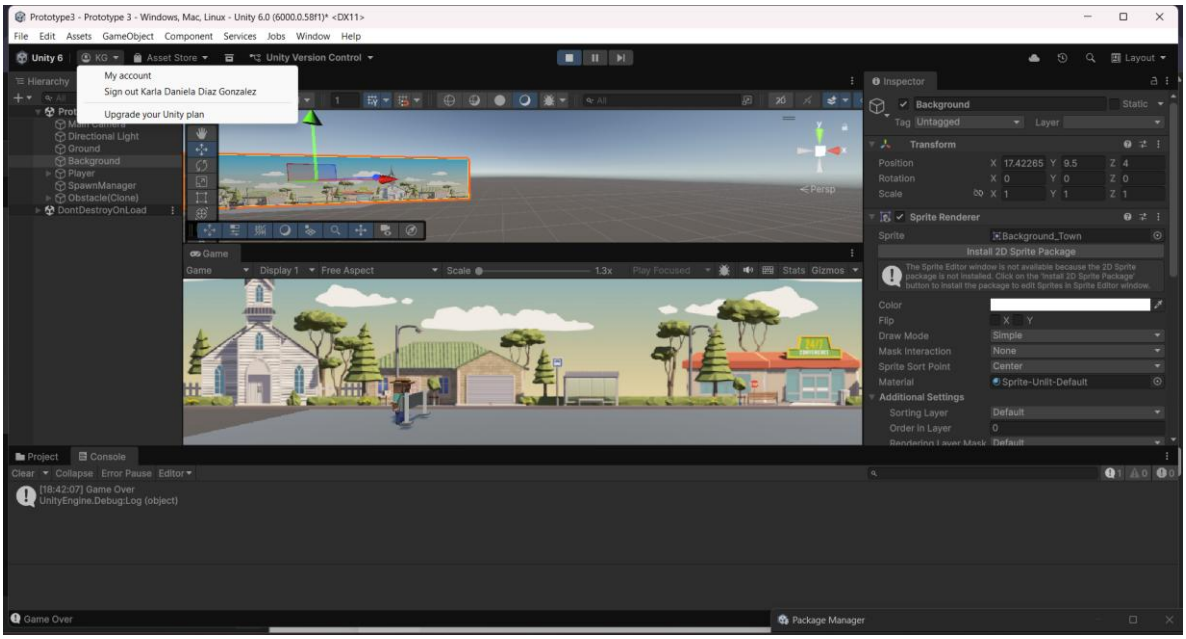


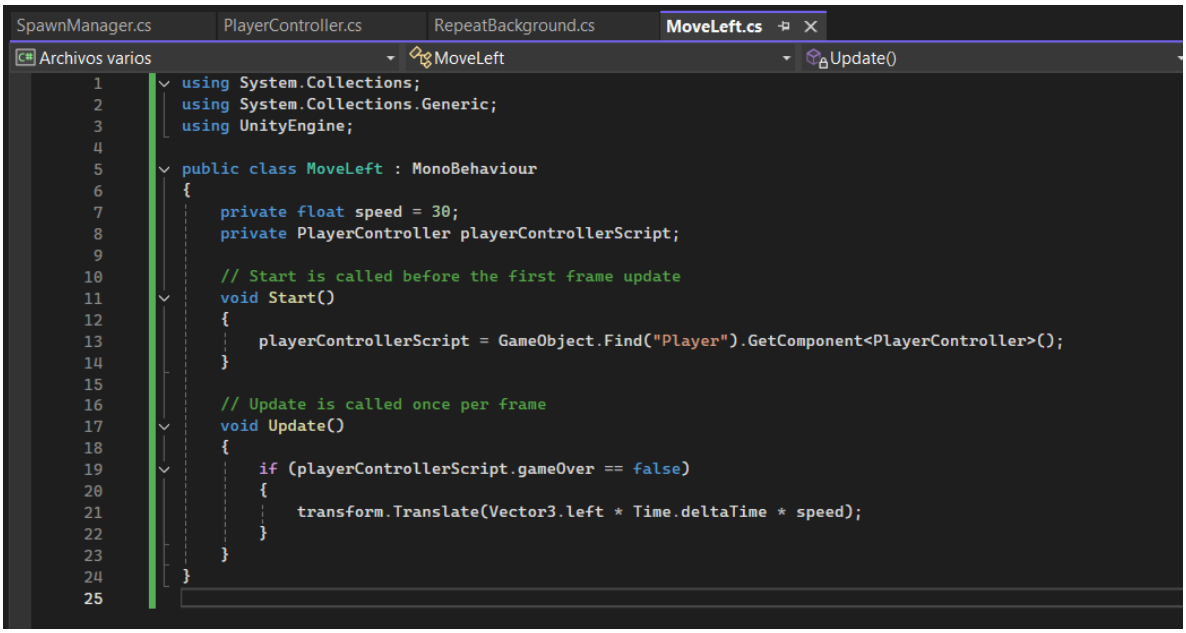

```cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    private Rigidbody playerRb;
    public float jumpForce = 10;
    public float gravityModifier;
    public bool isOnGround = true;
    public bool gameOver;

    // Start is called
    void Start()
    {
        playerRb = GetComponent<Rigidbody>();
        Physics.gravity *= gravityModifier;
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space) && isOnGround)
        {
            playerRb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
            isOnGround = false;
        }
    }

    private void OnCollisionEnter(Collision collision)
    {

        if(collision.gameObject.CompareTag("Ground"))
        {
            isOnGround = true;
        }else if(collision.gameObject.CompareTag("Obstacle"))
        {
            Debug.Log("Game Over");
            gameOver = true;
        }
    }
}
```

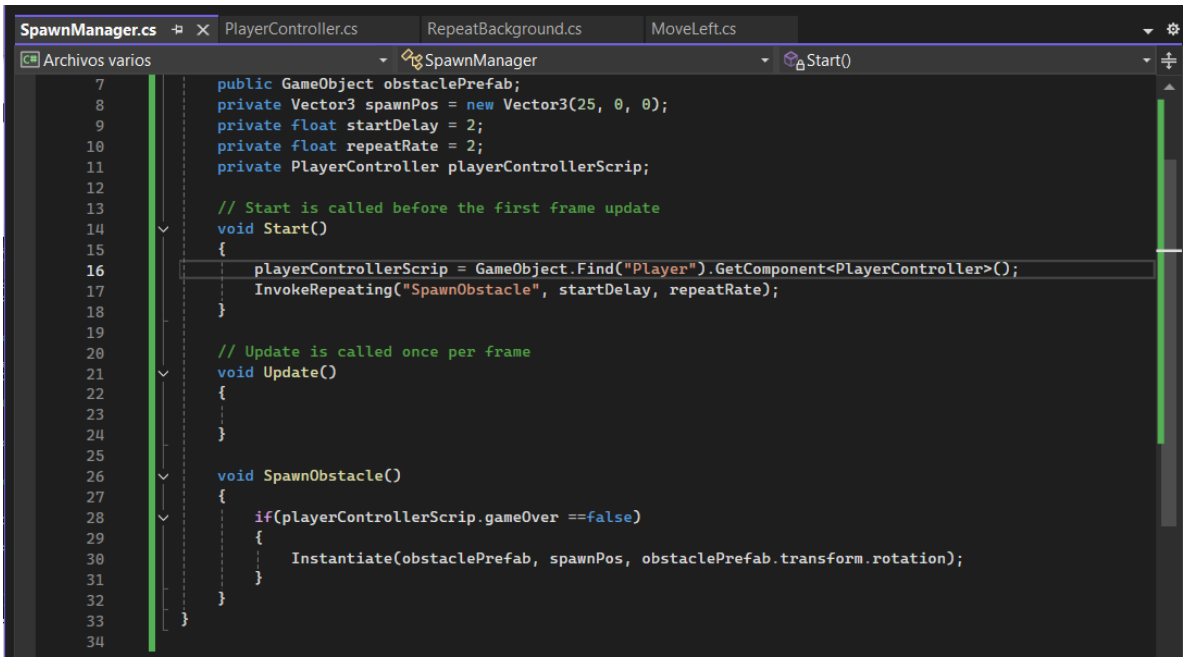



```cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MoveLeft : MonoBehaviour
{
    private float speed = 30;
    private PlayerController playerControllerScript;

    // Start is called before the first frame update
    void Start()
    {
        playerControllerScript = GameObject.Find("Player").GetComponent<PlayerController>();
    }

    // Update is called once per frame
    void Update()
    {
        if (playerControllerScript.gameOver == false)
        {
            transform.Translate(Vector3.left * Time.deltaTime * speed);
        }
    }
}
```

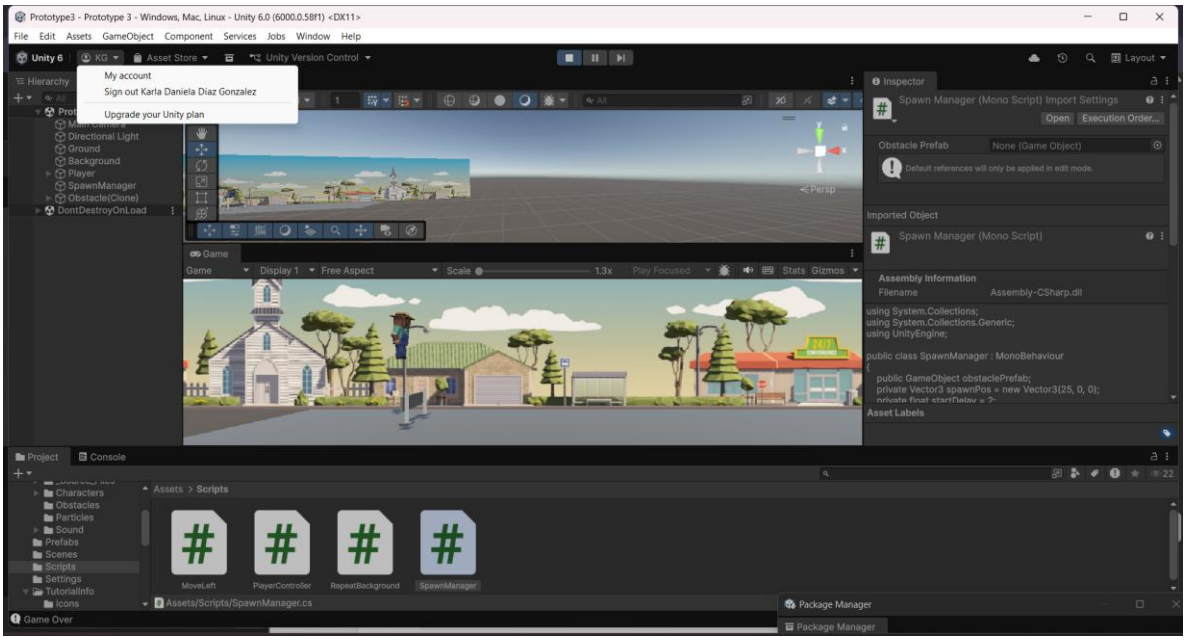
- En SpawnManager.cs , obtenga una referencia al playerControllerScript usando la misma técnica que utilizó en MoveLeft.cs
- Agregue una condición para instanciar objetos solo si gameOver == false


Prototype3 - Prototype 3 - Windows, Mac, Linux - Unity 6.0 (6000.0.58f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
Asset Store
Unity Version Control
My account
Sign out Karla Daniela Diaz Gonzalez
Upgrade your Unity plan
Hierarchy
Directional Light
Ground
Background
Player
SpawnManager
Obstacle(Clone)
DontDestroyOnLoad
Persp
Game
Display 1
Free Aspect
Play Focused
Stats
Gizmos
Inspector
Background
Static
Tag Untagged
Layer
Transform
Position
Rotation
Scale
Sprite Renderer
Sprite
Background_Town
Install 2D Sprite Package
Color
Flip
Draw Mode
Simple
Mask Interaction
None
Sprite Sort Point
Center
Material
Sprite-Unlit-Default
Additional Settings
Sorting Layer
Default
Order in Layer
Project
Console
Clear
Collapse
Error Pause
Editor
[18:42:07] Game Over
UnityEngine.Debug:Log (object)
Game Over
Package Manager


SpawnManager.cs
PlayerController.cs
RepeatBackground.cs
MoveLeft.cs
Archivos varios
MoveLeft
Update()
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class MoveLeft : MonoBehaviour
{
private float speed = 30;
private PlayerController playerControllerScript;
// Start is called before the first frame update
void Start()
{
playerControllerScript = GameObject.Find("Player").GetComponent<PlayerController>();
}
// Update is called once per frame
void Update()
{
if (playerControllerScript.gameOver == false)
{
transform.Translate(Vector3.left * Time.deltaTime * speed);
}
}
}



- En MoveLeft , en Update() ; escriba una declaración if para destruir obstáculos si su posición es menor que una variable leftBound
- Añade cualquier comentario que necesites para que tu código sea más legible.

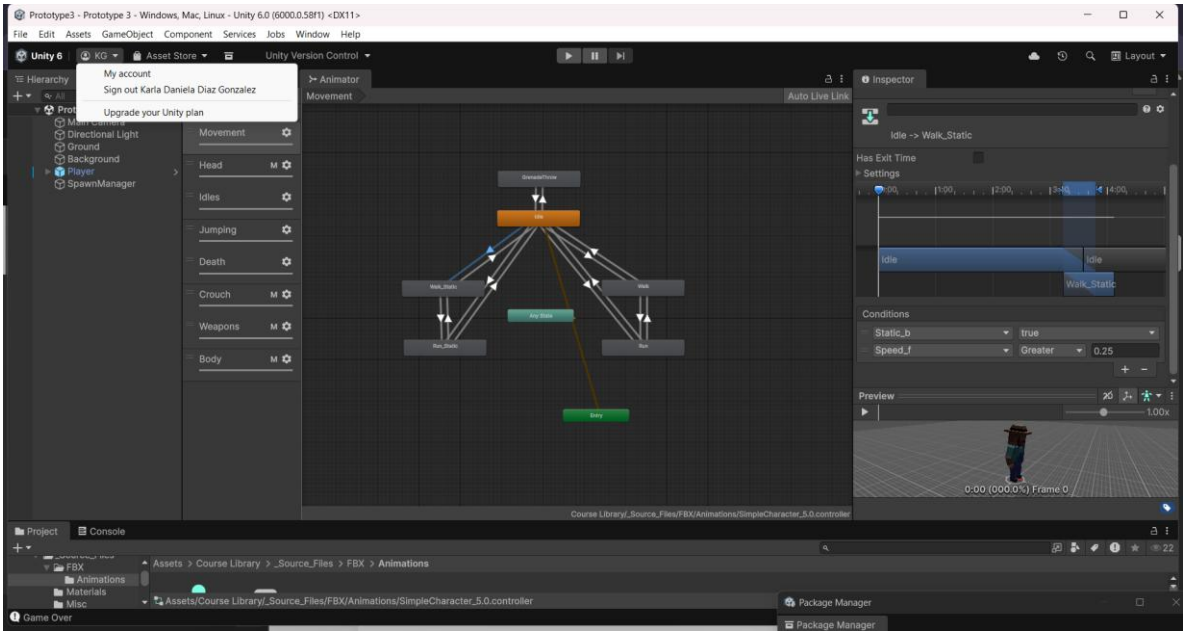


SpawnManager.cs | RepeatBackground.cs | **MoveLeft.cs** | PlayerController.cs

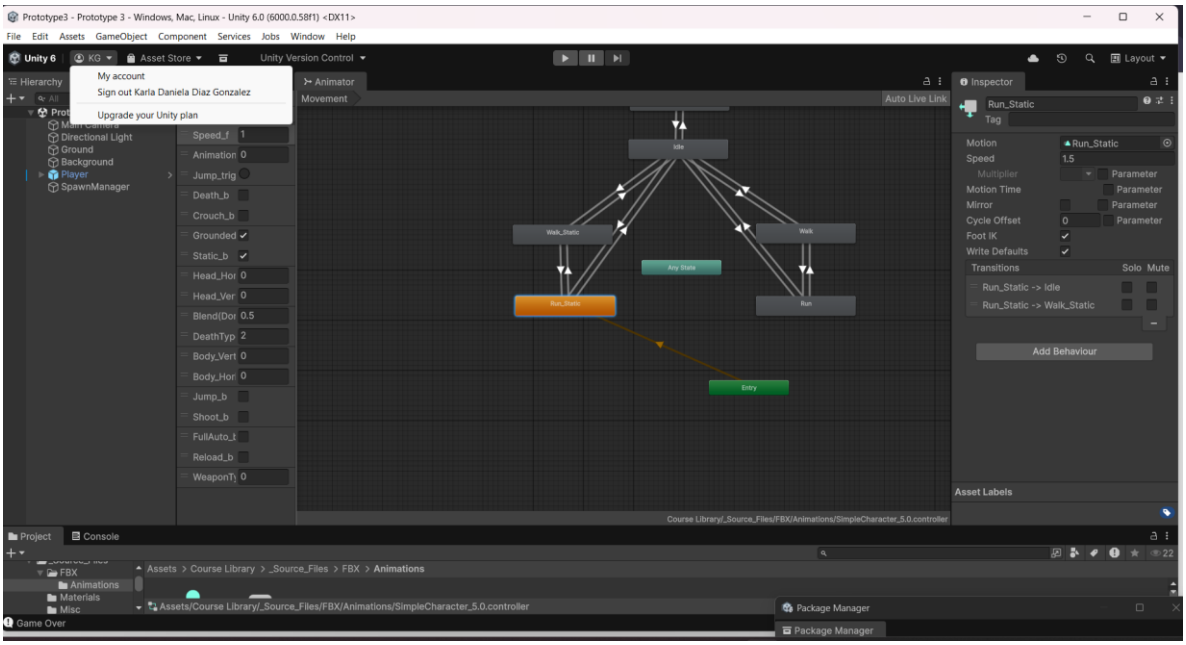
Archivos varios | MoveLeft | speed

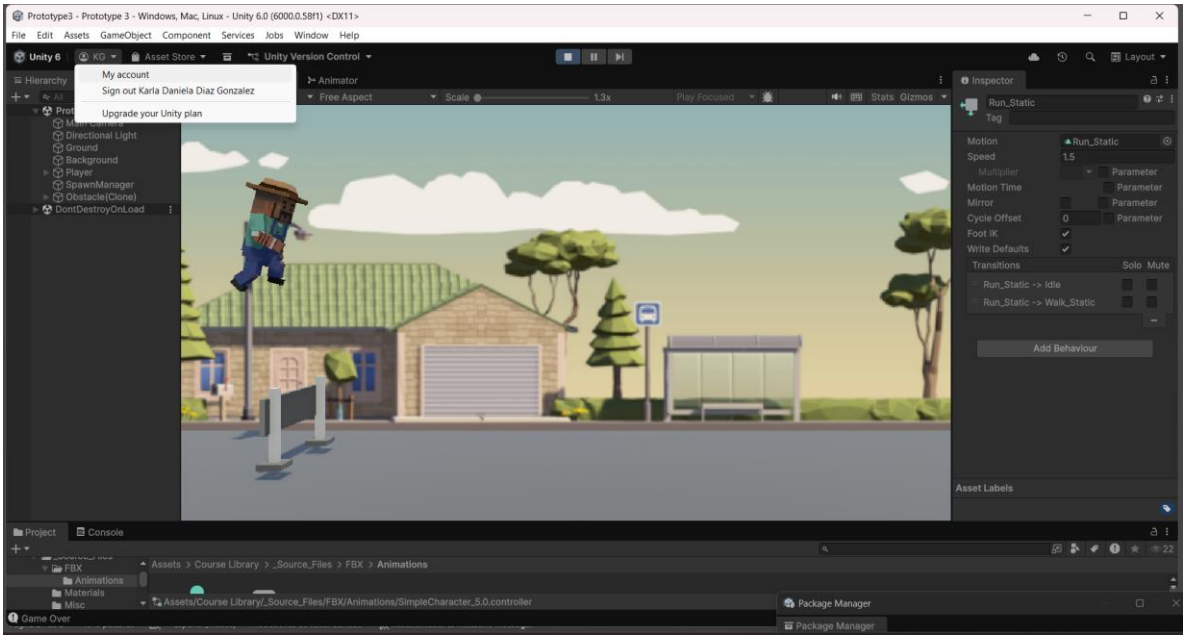
```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class MoveLeft : MonoBehaviour
6   {
7       private float speed = 30;
8       private PlayerController playerControllerScript;
9       private float leftBound = -15;
10
11      // Start is called before the first frame update
12      void Start()
13      {
14          playerControllerScript = GameObject.Find("Player").GetComponent<PlayerController>();
15      }
16
17      // Update is called once per frame
18      void Update()
19      {
20          if (playerControllerScript.gameOver == false)
21          {
22              transform.Translate(Vector3.left * Time.deltaTime * speed);
23          }
24
25          if (transform.position.x < leftBound && gameObject.CompareTag("Obstacle"))
26          {
27              Destroy(gameObject);
28          }
29      }
30  }
31
```

- Haga doble clic en el Controlador de animación del jugador, luego explore las diferentes capas, haciendo doble clic en Estados para ver sus animaciones y Transiciones para ver sus condiciones



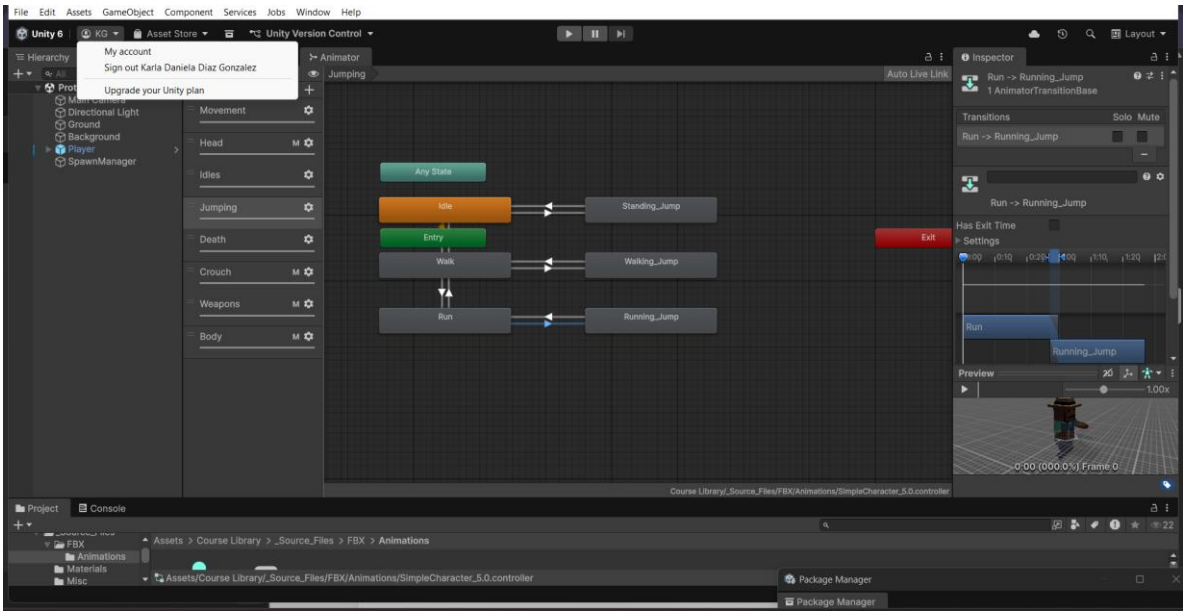
- En la pestaña Parámetros, cambie la variable Speed_f a 1.0
- Haga clic derecho en Run_Static > Establecer como estado predeterminado de capa
- Haga clic una vez en el estado Run_Static y ajuste el valor de Velocidad en el inspector para que coincida con la velocidad del fondo.

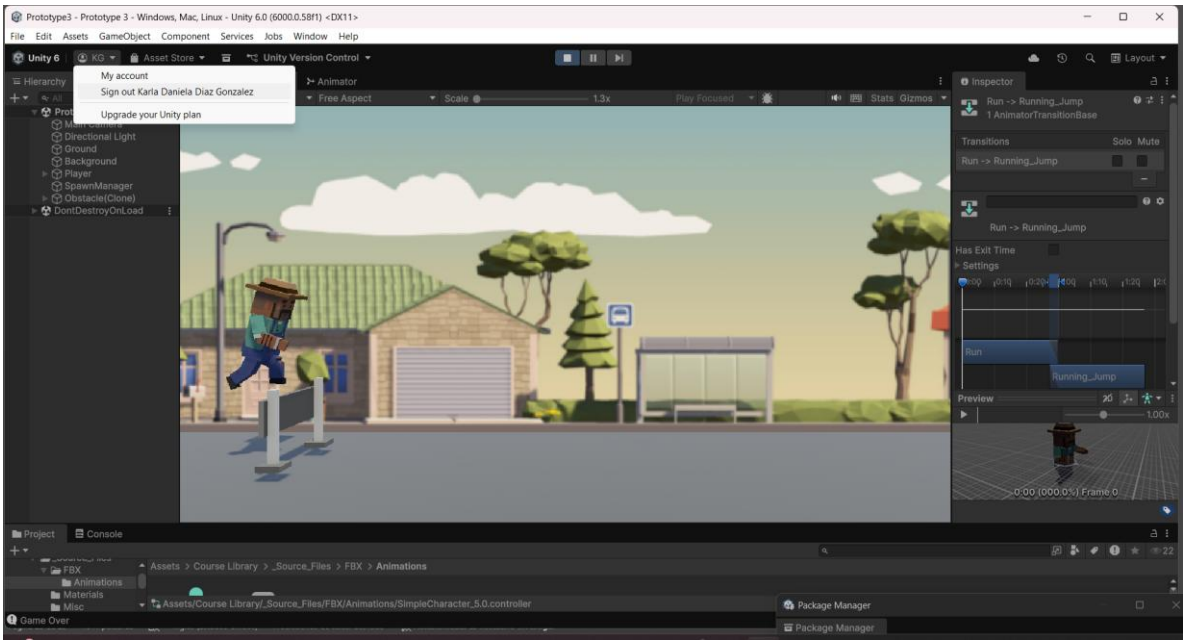


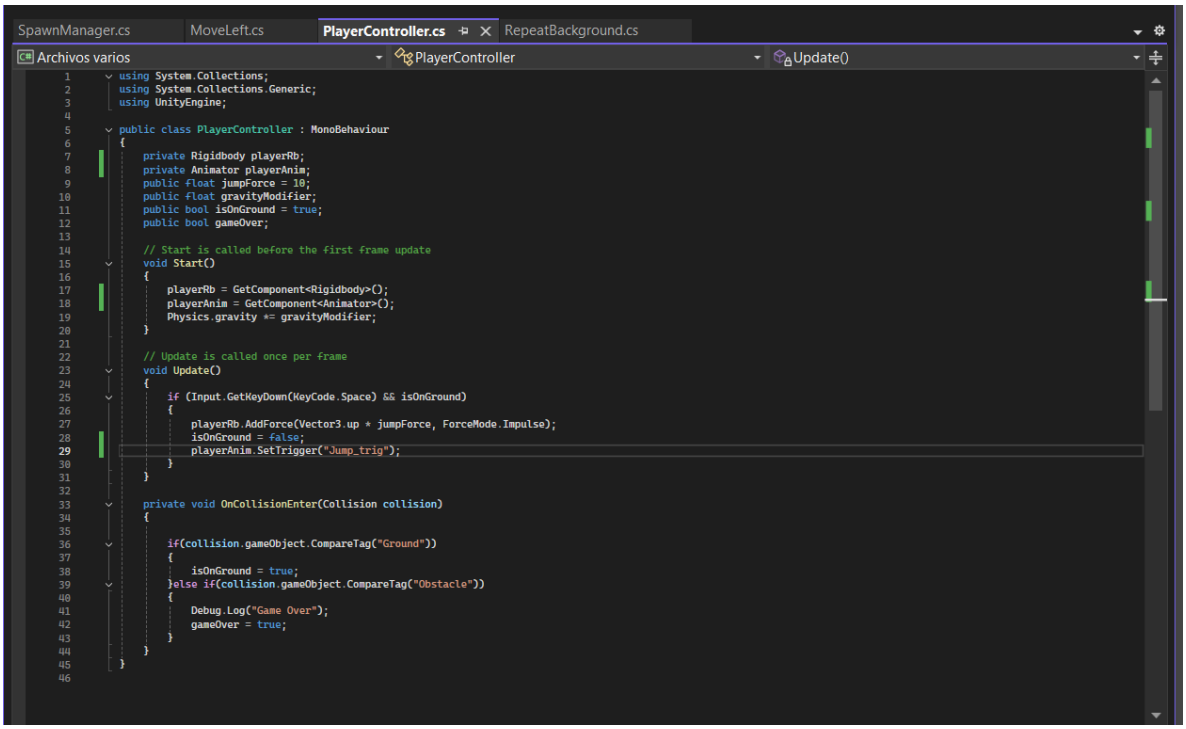


- En PlayerController.cs , declare un nuevo *Animator privado playerAnim;*

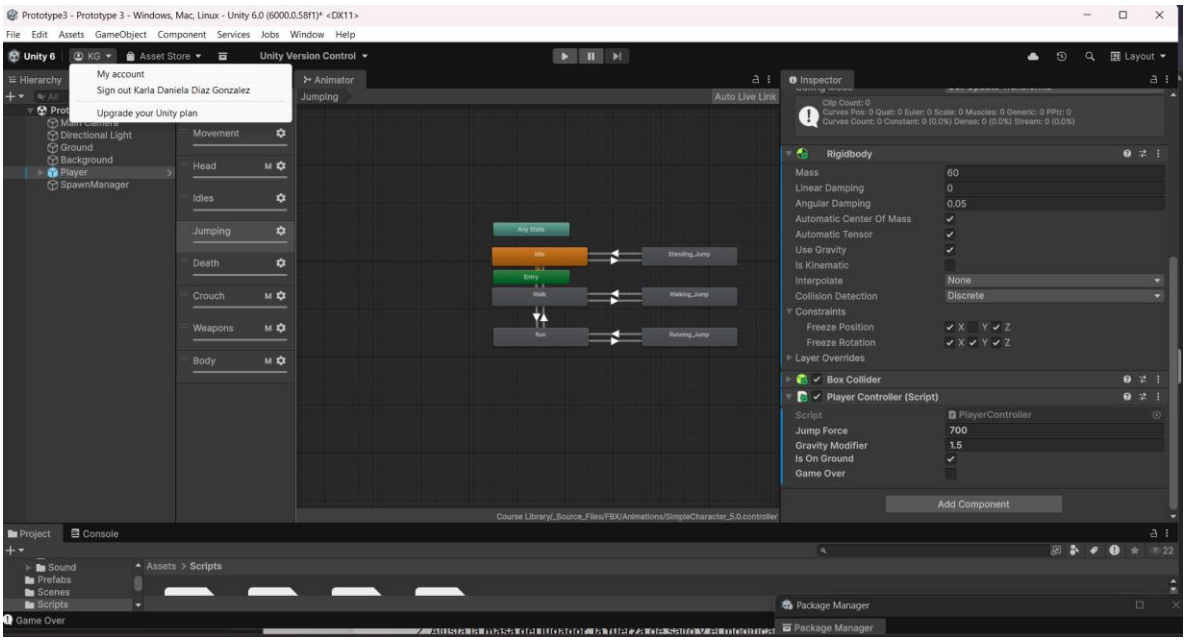
- En *Start()* , establezca *playerAnim = GetComponent<Animator>();*
- En la declaración if para cuando el jugador salta, activa el salto: playerAnim .*SetTrigger(“Jump_trig”);*

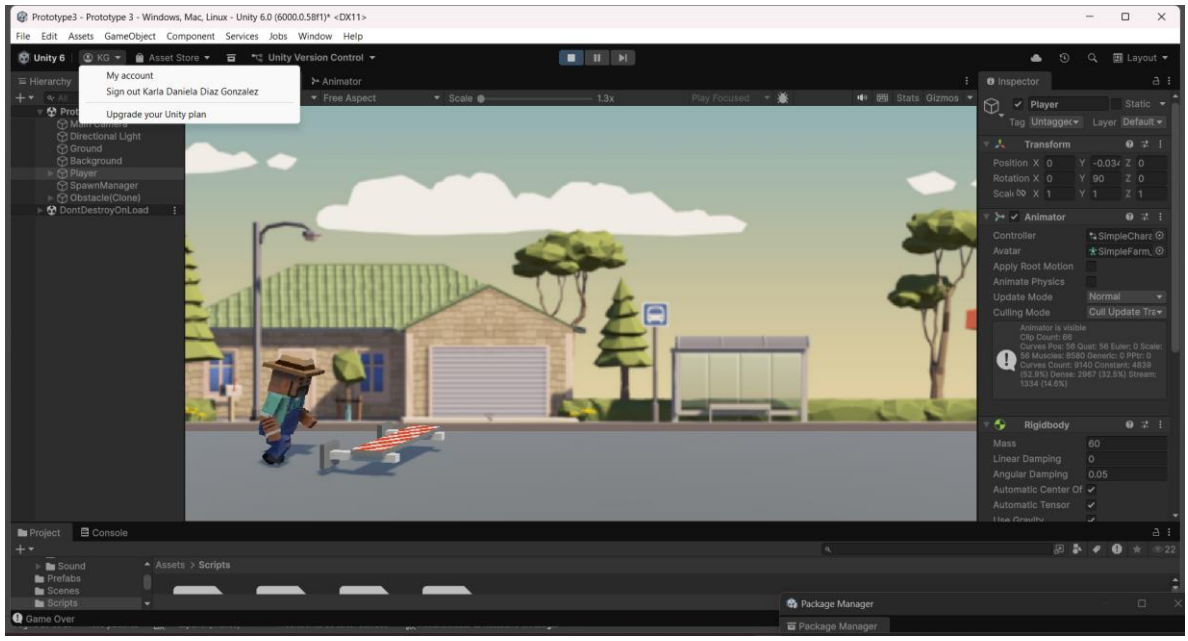




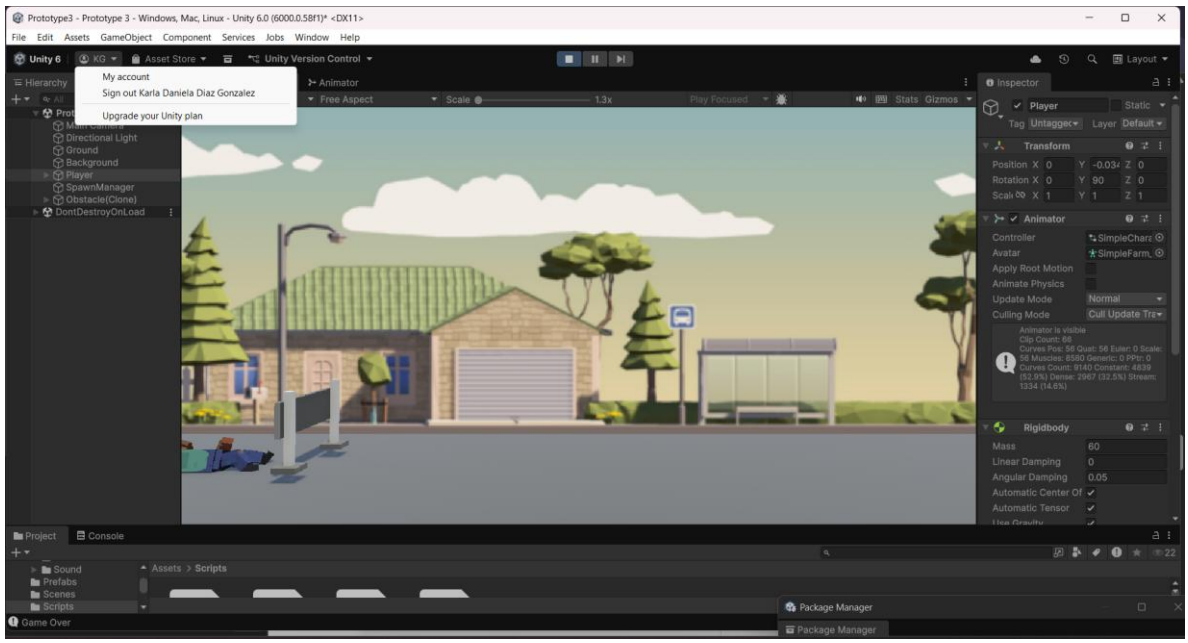


- En la ventana Animator, haga clic en el estado Running_Jump , luego en el inspector y reduzca su valor de Velocidad para ralentizar la animación.
- Ajusta la masa del jugador, la fuerza de salto y el modificador de gravedad para que tu salto sea perfecto.





- En la condición de que el jugador colisione con un obstáculo, establezca el booleano Muerte en verdadero
- En la misma declaración if , establezca el entero DeathType en 1



SpawnManager.cs | MoveLeft.cs | **PlayerController.cs** | RepeatBackground.cs

Archivos varios — PlayerController

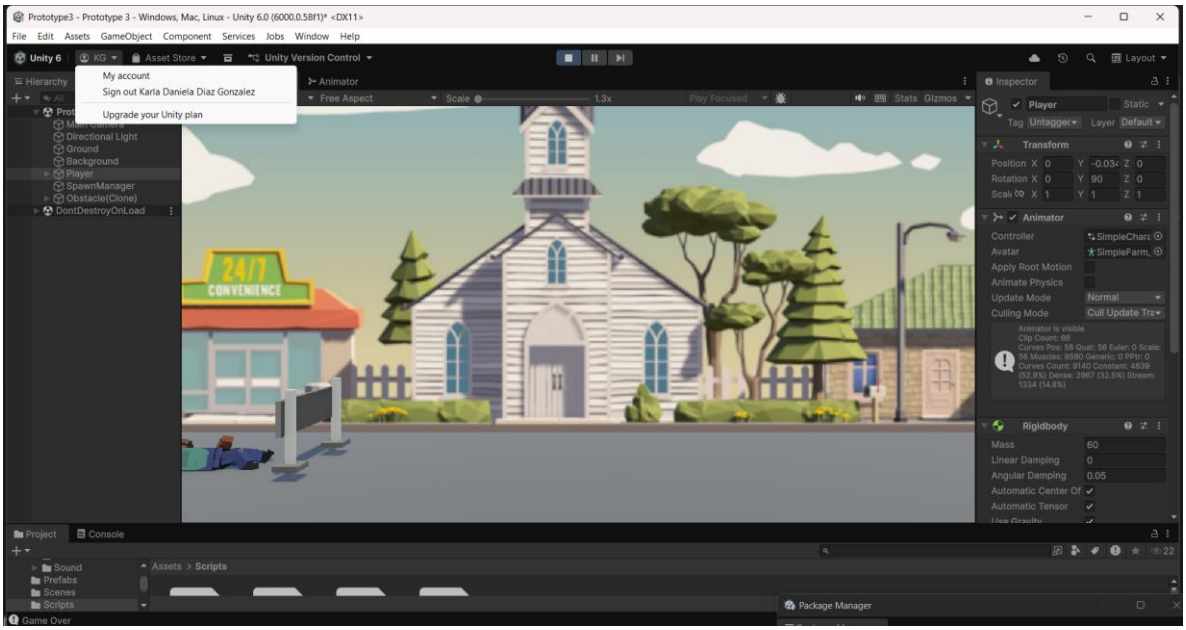
```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class PlayerController : MonoBehaviour
6   {
7       private Rigidbody playerRb;
8       private Animator playerAnim;
9       public float jumpForce = 10;
10      public float gravityModifier;
11      public bool isOnGround = true;
12      public bool gameOver;
13
14      // Start is called before the first frame update
15      void Start()
16      {
17          playerRb = GetComponent<Rigidbody>();
18          playerAnim = GetComponent<Animator>();
19          Physics.gravity *= gravityModifier;
20      }
21
22      // Update is called once per frame
23      void Update()
24      {
25          if (Input.GetKeyDown(KeyCode.Space) && isOnGround)
26          {
27              playerRb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
28              isOnGround = false;
29              playerAnim.SetTrigger("Jump_trig");
30          }
31      }
32
33      private void OnCollisionEnter(Collision collision)
34      {
35
36          if(collision.gameObject.CompareTag("Ground"))
37          {
38              isOnGround = true;
39          }else if(collision.gameObject.CompareTag("Obstacle"))
40          {
41              Debug.Log("Game Over");
42              gameOver = true;
43              playerAnim.SetBool("Death_b", true);
44              playerAnim.SetInteger("DeathType_int", 1);
45          }
46      }
47  }
48
```

- Para evitar que el jugador salte mientras está inconsciente, agregue && !gameOver a la condición de salto

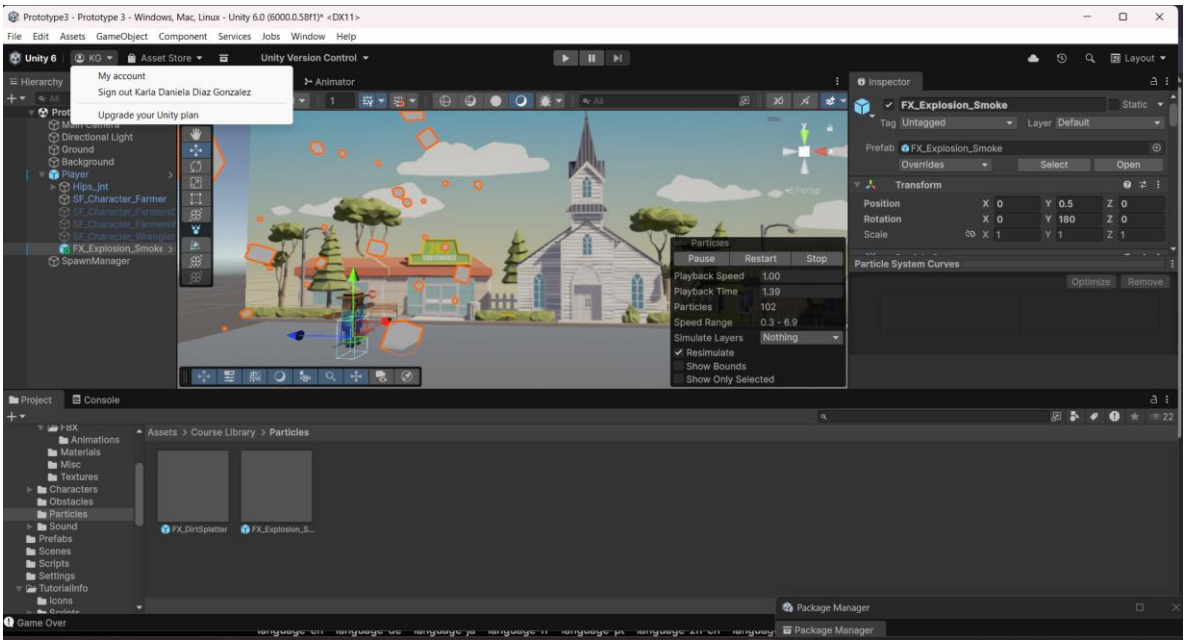
SpawnManager.cs | MoveLeft.cs | **PlayerController.cs** | RepeatBackground.cs

Archivos varios — PlayerController

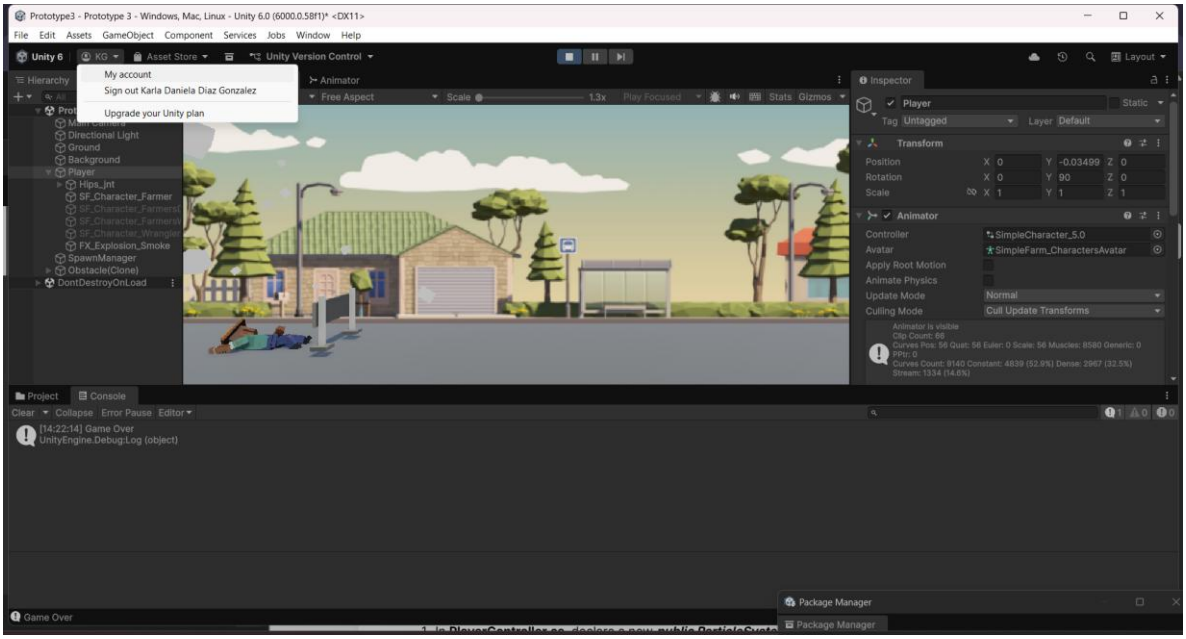
```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerController : MonoBehaviour
6  {
7      private Rigidbody playerRb;
8      private Animator playerAnim;
9      public float jumpForce = 10;
10     public float gravityModifier;
11     public bool isOnGround = true;
12     public bool gameOver;
13
14     // Start is called before the first frame update
15     void Start()
16     {
17         playerRb = GetComponent<Rigidbody>();
18         playerAnim = GetComponent<Animator>();
19         Physics.gravity *= gravityModifier;
20     }
21
22     // Update is called once per frame
23     void Update()
24     {
25         if (Input.GetKeyDown(KeyCode.Space) && isOnGround && !gameOver)
26         {
27             playerRb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
28             isOnGround = false;
29             playerAnim.SetTrigger("Jump_trig");
30         }
31     }
32
33     private void OnCollisionEnter(Collision collision)
34     {
35
36         if(collision.gameObject.CompareTag("Ground"))
37         {
38             isOnGround = true;
39         }else if(collision.gameObject.CompareTag("Obstacle"))
40         {
41             Debug.Log("Game Over");
42             gameOver = true;
43             playerAnim.SetBool("Death_b", true);
44             playerAnim.SetInteger("DeathType_int", 1);
45         }
46     }
47 }
```



- Desde la Biblioteca *de cursos* > *Partículas* , arrastre FX_Explosion_Smoke a la jerarquía y luego use los botones Reproducir / Reiniciar / Detener para obtener una vista previa.
- Juega con la configuración para conseguir que tu sistema de partículas sea como lo deseas.
- Asegúrese de desmarcar la configuración Reproducir al despertar
- Arrastre la partícula hacia su reproductor para convertirla en un objeto secundario y luego colóquela en relación con el reproductor.



- En PlayerController.cs , declare un nuevo ParticleSystem explosionParticle público;
- En el Inspector, asigne la explosión a la variable de partícula de explosión
- En la declaración if donde el jugador choca con un obstáculo, llame a explosionParticle.Play(); , luego pruebe y ajuste las propiedades de la partícula

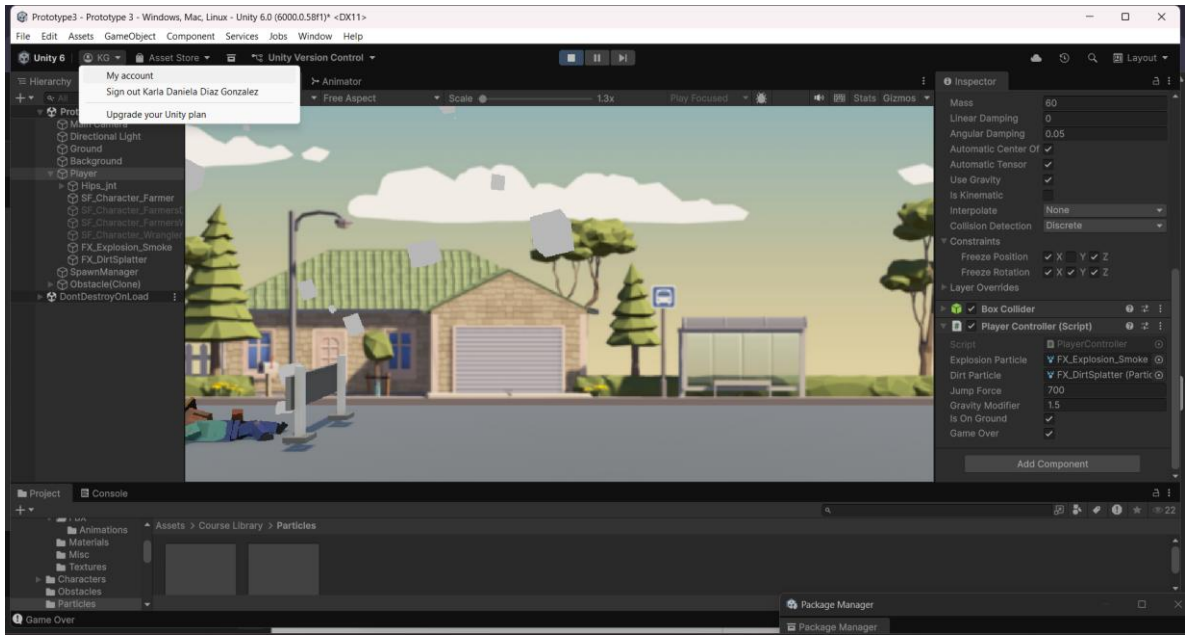


SpawnManager.cs | MoveLeft.cs | **PlayerController.cs** | RepeatBackground.cs

Archivos varios — PlayerController

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class PlayerController : MonoBehaviour
6   {
7       private Rigidbody playerRb;
8       private Animator playerAnim;
9       public ParticleSystem explosionParticle;
10      public float jumpForce = 10;
11      public float gravityModifier;
12      public bool isOnGround = true;
13      public bool gameOver;
14
15      // Start is called before the first frame update
16      void Start()
17      {
18          playerRb = GetComponent<Rigidbody>();
19          playerAnim = GetComponent<Animator>();
20          Physics.gravity *= gravityModifier;
21      }
22
23      // Update is called once per frame
24      void Update()
25      {
26          if (Input.GetKeyDown(KeyCode.Space) && isOnGround && !gameOver)
27          {
28              playerRb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
29              isOnGround = false;
30              playerAnim.SetTrigger("Jump_trig");
31          }
32      }
33
34      private void OnCollisionEnter(Collision collision)
35      {
36          if (collision.gameObject.CompareTag("Ground"))
37          {
38              isOnGround = true;
39          }
40          else if (collision.gameObject.CompareTag("Obstacle"))
41          {
42              Debug.Log("Game Over");
43              gameOver = true;
44              playerAnim.SetBool("Death_b", true);
45              playerAnim.SetInteger("DeathType_int", 1);
46              explosionParticle.Play();
47          }
48      }
49  }
50
```

- Arrastre FX_DirtSplatter como objeto secundario del jugador , reposicionelo, rótelo y edite su configuración
- Declare un nuevo *ParticleSystem público dirtParticle;* y luego asígnelo en el Inspector
- Añade *dirtParticle.Stop();* cuando el jugador salta o choca con un obstáculo
- Añade *dirtParticle.Play();* cuando el jugador aterriza en el suelo

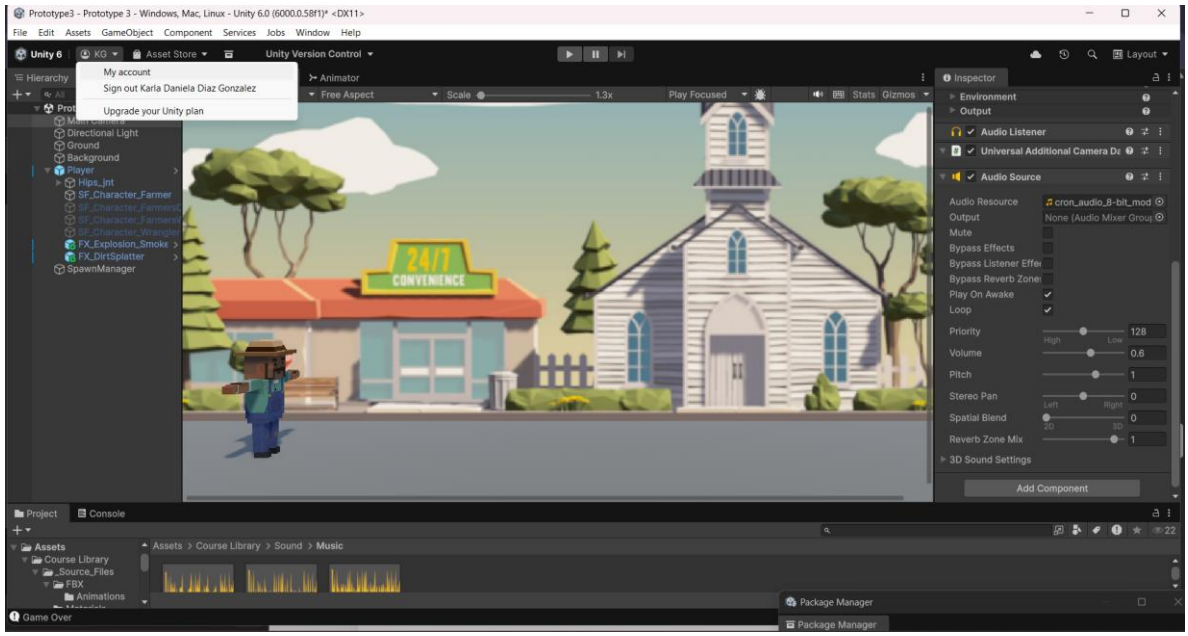

Prototype3 - Prototype 3 - Windows, Mac, Linux - Unity 6.0 (6000.0.58f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
KG
Asset Store
Unity Version Control
Layout
My account
Sign out Karla Daniela Diaz Gonzalez
Upgrade your Unity plan
Hierarchy
Animator
Free Aspect
Scale
1.3x
Play Focused
Stats
Gizmos
Directional Light
Ground
Background
Player
Hips_Jnt
SF_Character_Farmer
FX_Explosion_Smoke
FX_DirtSplatter
SpawnManager
Obstacle(Clone)
DontDestroyOnLoad
Inspector
Mass 60
Linear Damping 0
Angular Damping 0.05
Automatic Center Of
Automatic Tensor
Use Gravity
Is Kinematic
Interpolate None
Collision Detection Discrete
Constraints
Freeze Position
Freeze Rotation
Layer Overrides
Box Collider
Player Controller (Script)
Script PlayerController
Explosion Particle FX_Explosion_Smoke
Dirt Particle FX_DirtSplatter
Jump Force 700
Gravity Modifier 1.5
Is On Ground
Game Over
Add Component
Project
Console
Assets > Course Library > Particles
Animations
Materials
Misc
Textures
Characters
Obstacles
Particles
Game Over
Package Manager
Package Manager

SpawnManager.cs | MoveLeft.cs | **PlayerController.cs** | RepeatBackground.cs

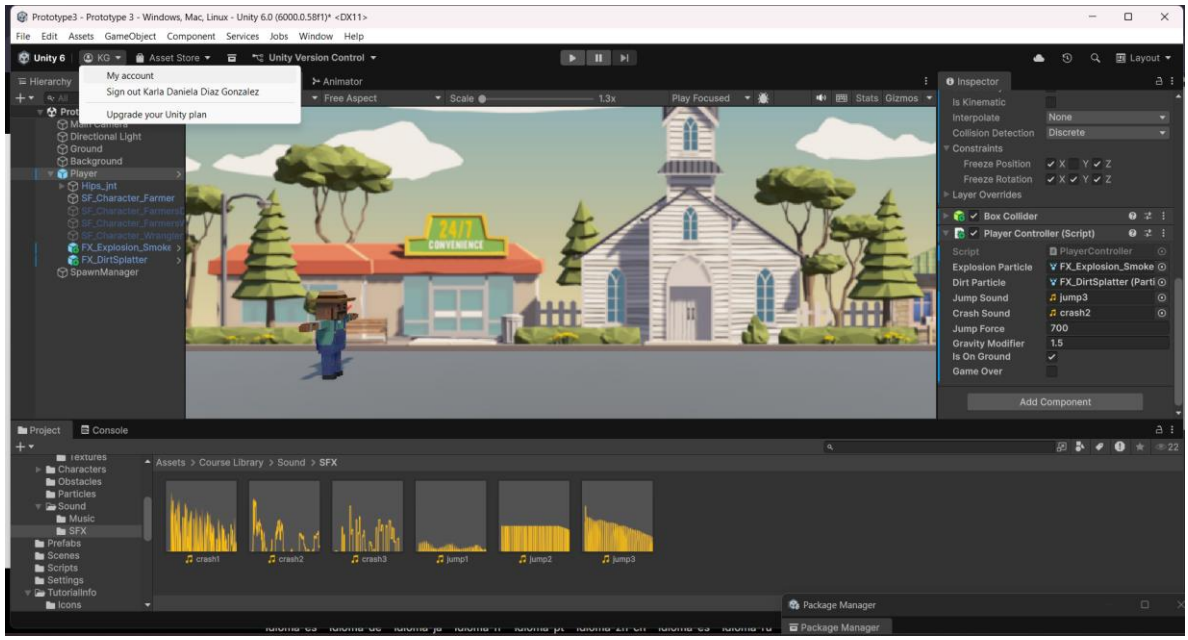
Archivos varios — PlayerController

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    private Rigidbody playerRb;
    private Animator playerAnim;
    public ParticleSystem explosionParticle;
    public ParticleSystem dirtParticle;
    public float jumpForce = 10;
    public float gravityModifier;
    public bool isOnGround = true;
    public bool gameOver;

    // Start is called before the first frame update
    void Start()
    {
        playerRb = GetComponent<Rigidbody>();
        playerAnim = GetComponent<Animator>();
        Physics.gravity *= gravityModifier;
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space) && isOnGround && !gameOver)
        {
            playerRb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
            isOnGround = false;
            playerAnim.SetTrigger("Jump_trig");
            dirtParticle.Stop();
        }
    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("Ground"))
        {
            isOnGround = true;
            dirtParticle.Play();
        }
        else if (collision.gameObject.CompareTag("Obstacle"))
        {
            Debug.Log("Game Over");
            gameOver = true;
            playerAnim.SetBool("Death_b", true);
            playerAnim.SetInteger("DeathType_int", 1);
            explosionParticle.Play();
            dirtParticle.Stop();
        }
    }
}
```

- Seleccione el objeto Cámara principal, luego Agregar componente > Fuente de audio
- Desde Biblioteca de cursos > Sonido, arrastre un clip de música a la variable Recurso de audio en el inspector
- Reduce el volumen para que sea más fácil escuchar los efectos de sonido.
- Marque la casilla de verificación Bucle



- En PlayerController.cs, declare un nuevo AudioClip público jumpSound; y un nuevo AudioClip público crashSound;
- Desde Biblioteca de cursos > Sonido, arrastre un clip sobre cada nueva variable de sonido en el inspector



SpawnManager.cs | MoveLeft.cs | **PlayerController.cs** | RepeatBackground.cs

Archivos varios — PlayerController

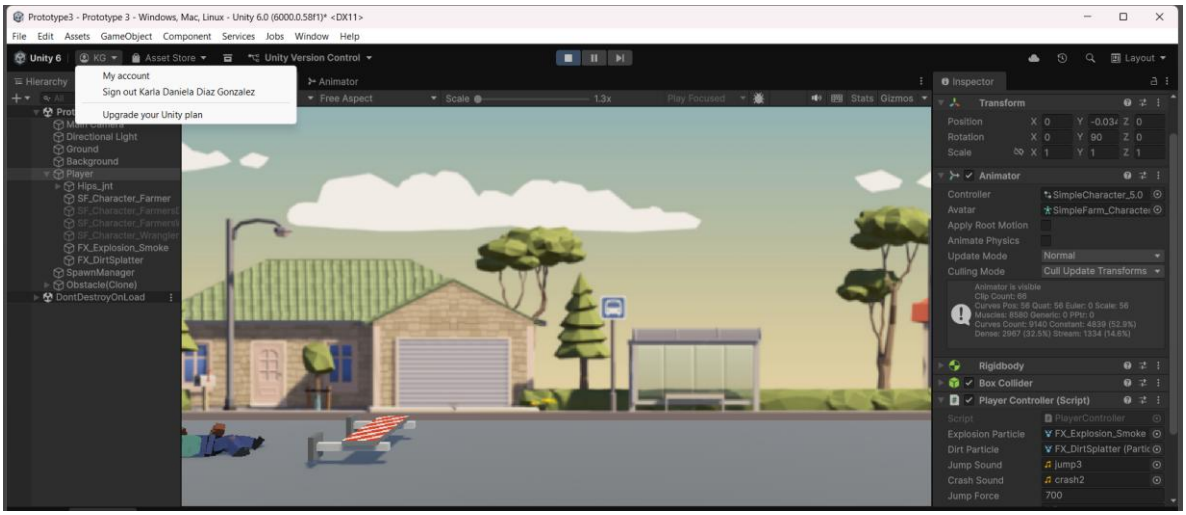
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    private Rigidbody playerRb;
    private Animator playerAnim;
    public ParticleSystem explosionParticle;
    public ParticleSystem dirtParticle;
    public AudioClip jumpSound;
    public AudioClip crashSound;
    public float jumpForce = 10;
    public float gravityModifier;
    public bool isOnGround = true;
    public bool gameOver;

    // Start is called before the first frame update
    void Start()
    {
        playerRb = GetComponent<Rigidbody>();
        playerAnim = GetComponent<Animator>();
        Physics.gravity *= gravityModifier;
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space) && isOnGround && !gameOver)
        {
            playerRb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
            isOnGround = false;
            playerAnim.SetTrigger("Jump_trig");
            dirtParticle.Stop();
        }
    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("Ground"))
        {
            isOnGround = true;
            dirtParticle.Play();
        }
        else if (collision.gameObject.CompareTag("Obstacle"))
        {
            Debug.Log("Game Over");
            gameOver = true;
            playerAnim.SetBool("Death_b", true);
            playerAnim.SetInteger("DeathType_int", 1);
            explosionParticle.Play();
            dirtParticle.Stop();
        }
    }
}
```

- Agregar un componente de fuente de audio al reproductor
- Declare un nuevo reproductor AudioSource privado playerAudio; e inicialícelo como playerAudio = GetComponent<AudioSource>();
- Llama a playerAudio.PlayOneShot(jumpSound, 1.0f); cuando el personaje salta
- Llama a playerAudio.PlayOneShot(crashSound, 1.0f); cuando el personaje se bloquea

SpawnManager.cs | MoveLeft.cs | **PlayerController.cs** | RepeatBackground.cs

Archivos varios — PlayerController

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    private Rigidbody playerRb;
    private Animator playerAnim;
    private AudioSource playerAudio;
    public ParticleSystem explosionParticle;
    public ParticleSystem dirtParticle;
    public AudioClip jumpSound;
    public AudioClip crashSound;
    public float jumpForce = 10;
    public float gravityModifier;
    public bool isOnGround = true;
    public bool gameOver;

    // Start is called before the first frame update
    void Start()
    {
        playerRb = GetComponent<Rigidbody>();
        playerAnim = GetComponent<Animator>();
        playerAudio = GetComponent<AudioSource>();
        Physics.gravity *= gravityModifier;
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space) && isOnGround && !gameOver)
        {
            playerRb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
            isOnGround = false;
            playerAnim.SetTrigger("Jump_trig");
            dirtParticle.Stop();
            playerAudio.PlayOneShot(jumpSound, 1.0f);
        }
    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("Ground"))
        {
            isOnGround = true;
            dirtParticle.Play();
        }
        else if (collision.gameObject.CompareTag("Obstacle"))
        {
            Debug.Log("Game Over");
            gameOver = true;
            playerAnim.SetBool("Death_b", true);
            playerAnim.SetInteger("DeathType_int", 1);
            explosionParticle.Play();
            dirtParticle.Stop();
            playerAudio.PlayOneShot(crashSound, 1.0f);
        }
    }
}

```


Prototype3 - Prototype 3 - Windows, Mac, Linux - Unity 6.0 (6000.0.58f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
KG
Asset Store
Unity Version Control
Layout
My account
Sign out Karla Daniela Diaz Gonzalez
Upgrade your Unity plan
Hierarchy
Animator
Free Aspect
Scale
1.3x
Play Focused
Stats
Gizmos
Directional Light
Ground
Background
Player
Hips_jnt
SF_Character_Farmer
FX_Explosion_Smoke
FX_DirtSplatter
SpawnManager
Obstacle(Clone)
DontDestroyOnLoad
Inspector
Transform
Position
Rotation
Scale
Animator
Controller
SimpleCharacter_5.0
Avatar
Apply Root Motion
Animate Physics
Update Mode
Normal
Culling Mode
Cull Update Transforms
Rigidbody
Box Collider
Player Controller (Script)
Script
PlayerController
Explosion Particle
FX_Explosion_Smoke
Dirt Particle
Jump Sound
jump3
Crash Sound
crash2
Jump Force
700