
SIMULAR LANZAMIENTO DE UN DADO EN UNITY

JUAN FRANCISCO MENDOZA MORENO, PH.D.

Universidad Santo Tomás Seccional Tunja, 2022

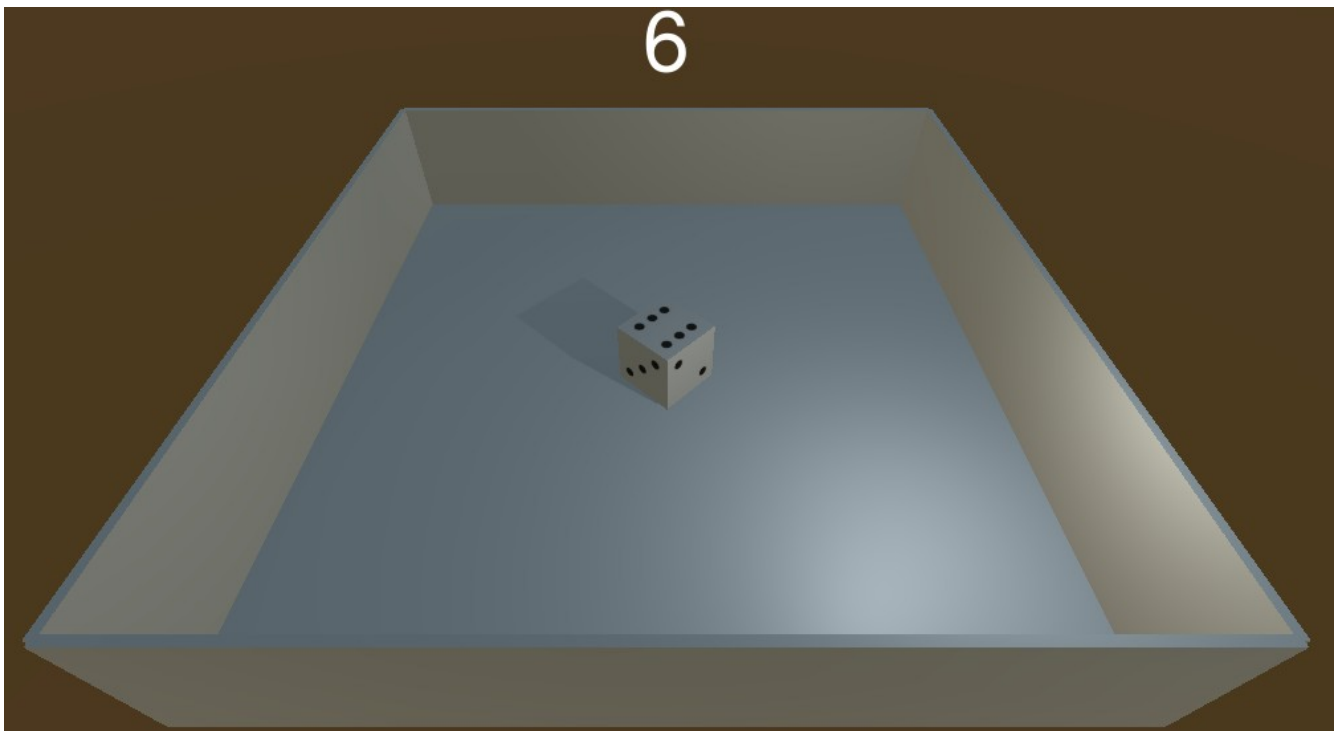


TABLA DE CONTENIDO

Introducción.....	3
1. Construcción de la Bandeja.....	4
2. Inclusión del objeto del dado.....	9
3. Configurar los lados del dado.....	13
4. Configuración de la Iluminación del escenario.....	20
5. Configurar la zona de colisión de la Bandeja.....	22
6. Configurar el audio de lanzamiento.....	27
Referencias.....	31

INTRODUCCIÓN

La generación de números aleatorios y la aleatoriedad son componentes importantes para un simulador. Un ejemplo clásico y sencillo de un escenario real de aleatoriedad es el lanzamiento de un dado, con seis posibles resultados, en cada lanzamiento de forma aleatoria se puede obtener uno de estos valores de forma discreta.

Este taller trata sobre la simulación, lo más real posible, del lanzamiento de un dado dentro de una bandeja, el resultado es la cara superior, opuesta a la cara de contacto con la bandeja. Utilizando el generador pseudoaleatorio de C# incluido para soportar los scripts de Unity, se generan movimientos de torque y de movimiento de un cuerpo rígido en las condiciones físicas similares a los entornos reales.

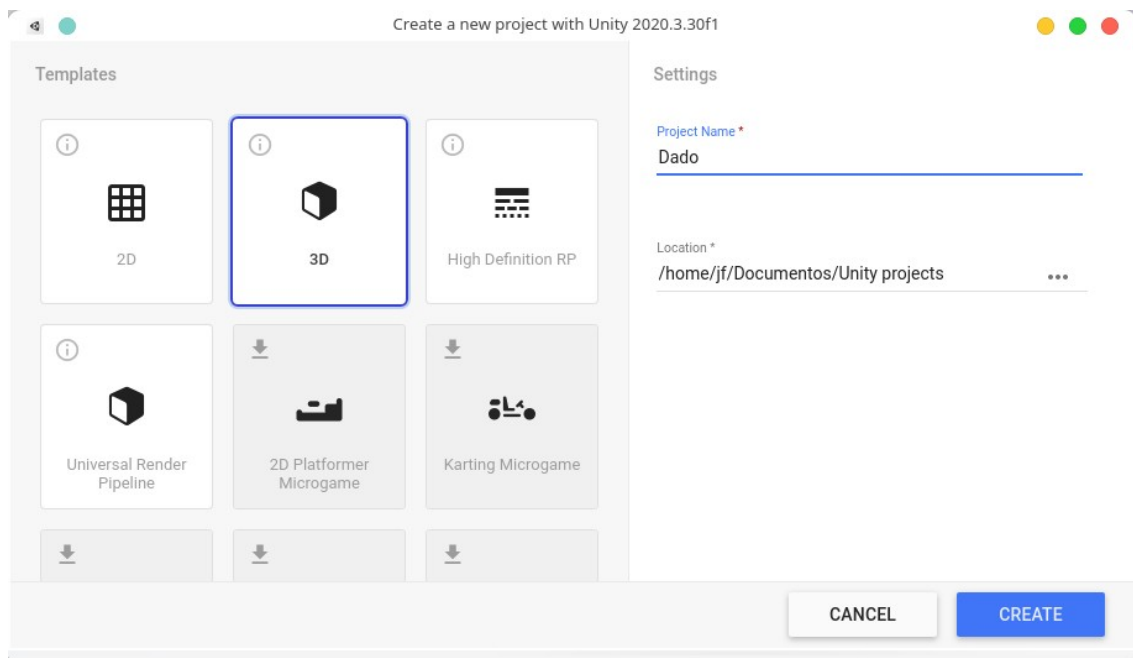
Cada lanzamiento del dado se genera con la pulsación de la tecla espaciadora y entonces se generan movimientos aleatorios (velocidad) en los ejes x, y, z, para que mediante el torque se brinde la apariencia del lanzamiento real.

1. CONSTRUCCIÓN DE LA BANDEJA

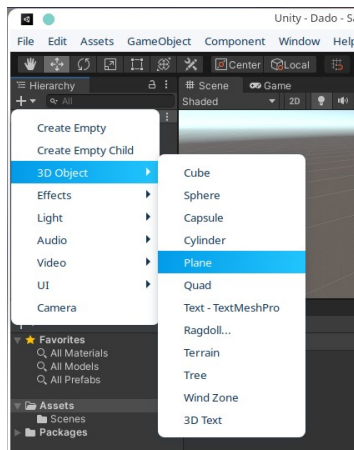
1. Abra **Unity Hub** y haga clic en el botón **New**:



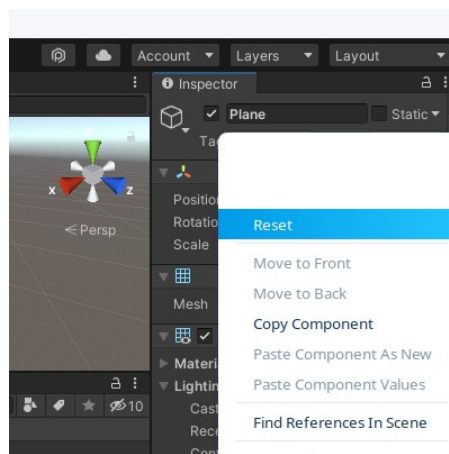
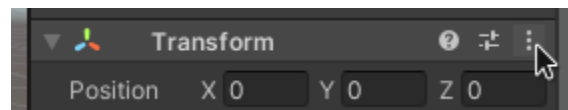
2. Teclee el Nombre del Proyecto y haga clic en el botón **Create**:



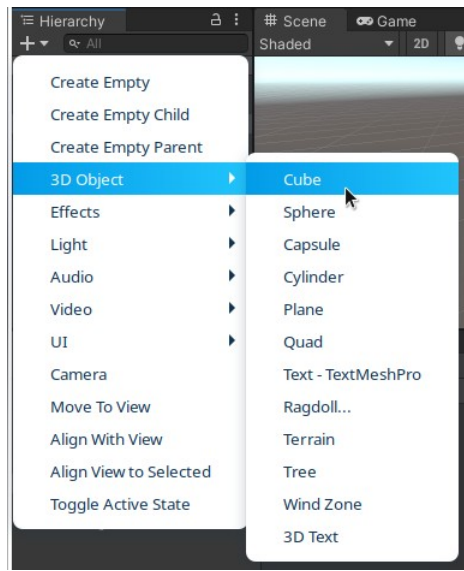
3. Haga clic en el signo + del panel **Hierarchy**, en el menú escoger **3D Object** y luego **Plane**. Este plano servirá de base de la bandeja de dados:



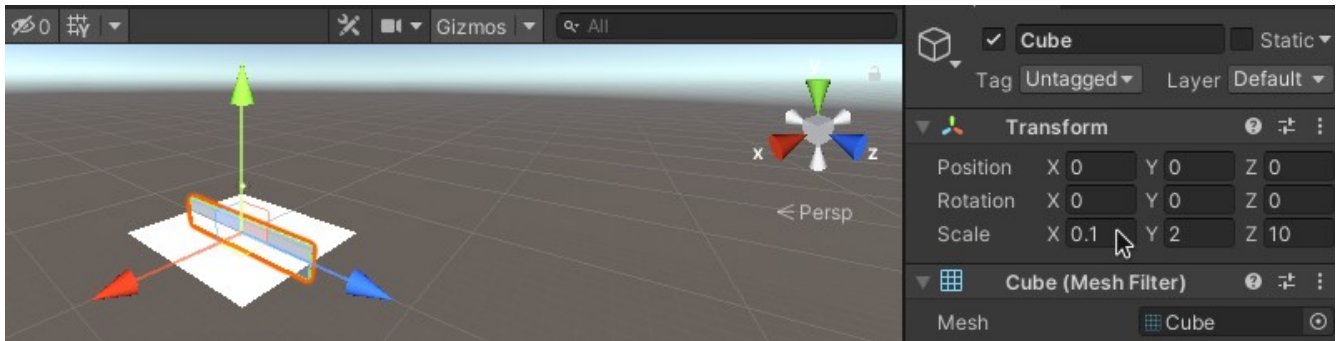
4. Reiniciamos la posición del plano, en el panel **Inspector** en los tres puntos verticales a la derecha de las propiedades de **Transform**:



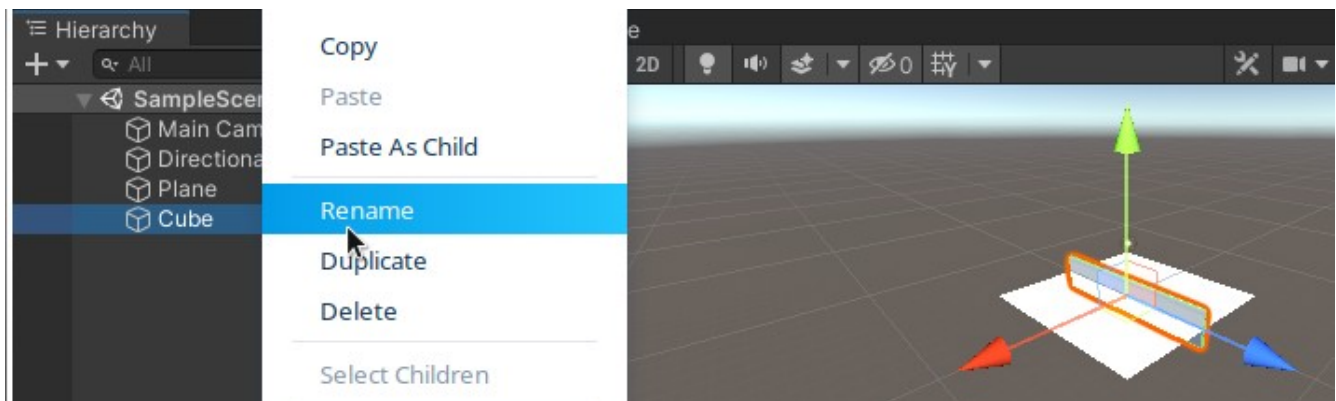
5. Agregar un cubo que servirá como el panel lateral (muro) de la bandeja. Haga clic en el signo + del panel **Hierarchy**, en el menú escoger **3D Object** y luego **Cube**.



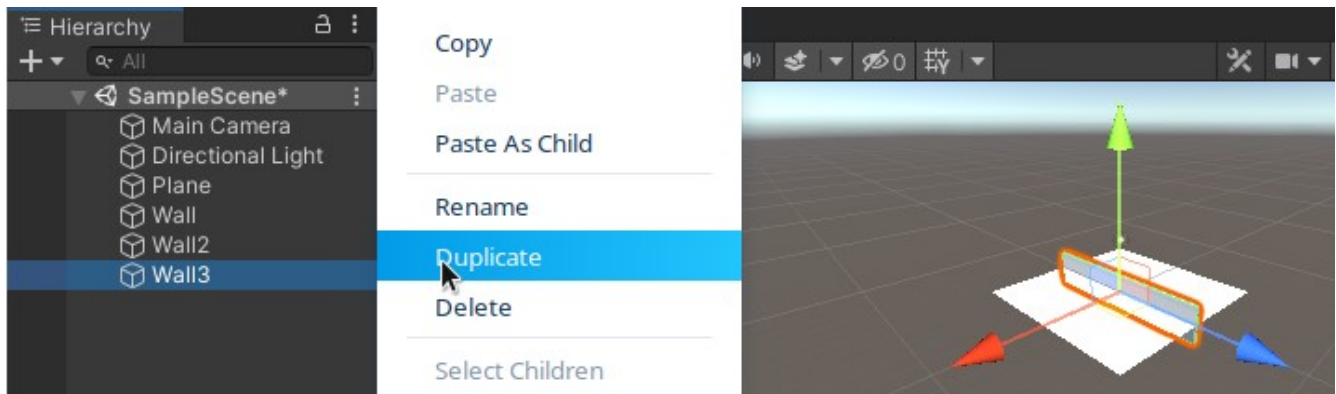
6. Modificar los parámetros de **Scale** del cubo adicionado (**x = 0.1, y = 2, z = 10**):



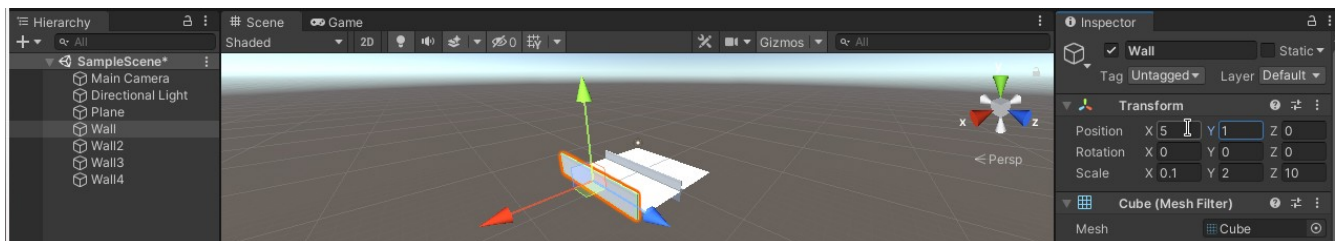
7. Renombrar el cubo a "**Wall**":



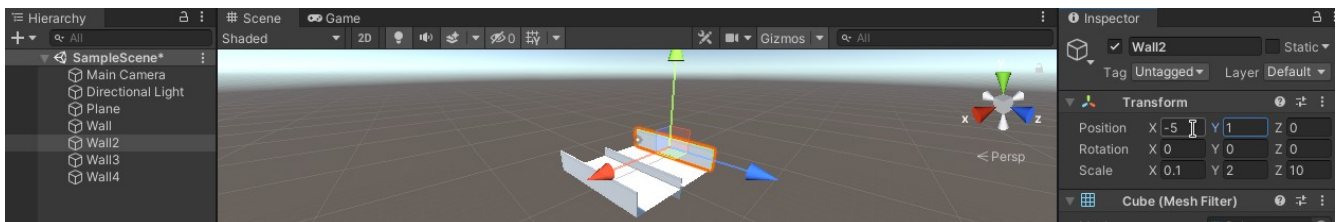
8. Hacer tres copias (cuatro en total) del cubo “**Wall**” y renombrar las copias como “**Wall2**”, “**Wall3**” y “**Wall4**”:



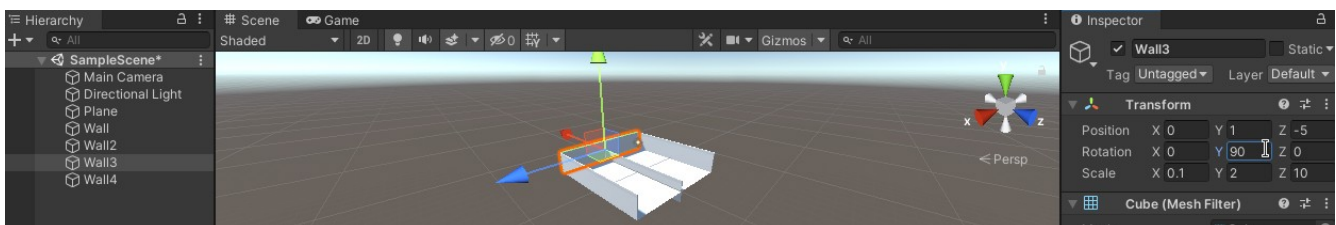
9. Cambiar los valores de posición de **Wall** ($x = 5$, $y = 1$, $z = 0$):



10. Cambiar los valores de posición de **Wall2** ($x = -5$, $y = 1$, $z = 0$):

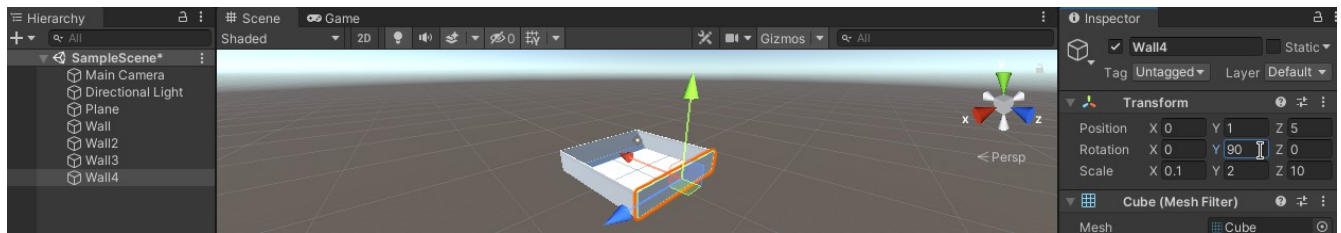


11. Cambiar los valores de posición de **Wall3** ($x = 0$, $y = 1$, $z = -5$) y de rotación en $y = 90$:



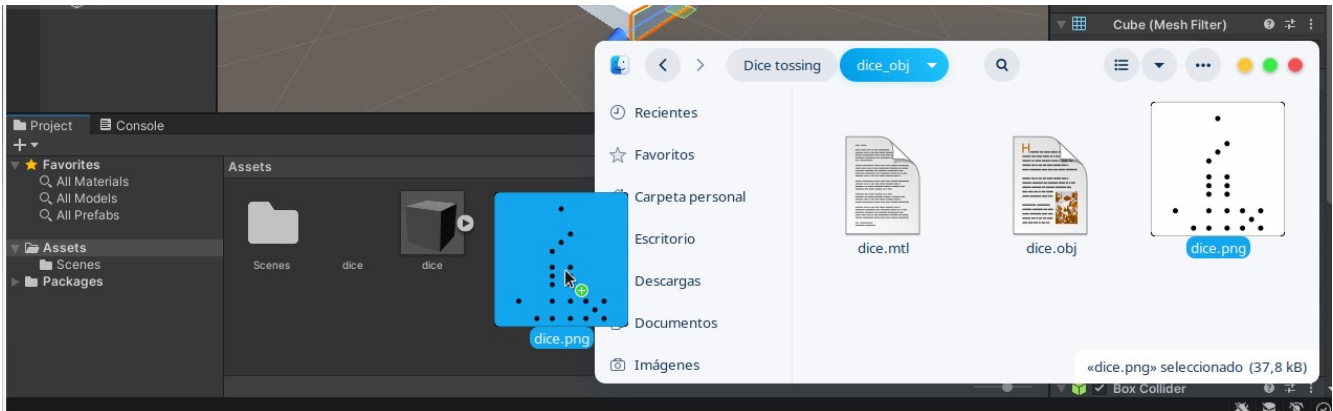
12. Cambiar los valores de posición de **Wall4** ($x = 0$, $y = 1$, $z = 5$) y de rotación en $y = 90$:

Simular lanzamiento de un dado en Unity – JF Mendoza M.

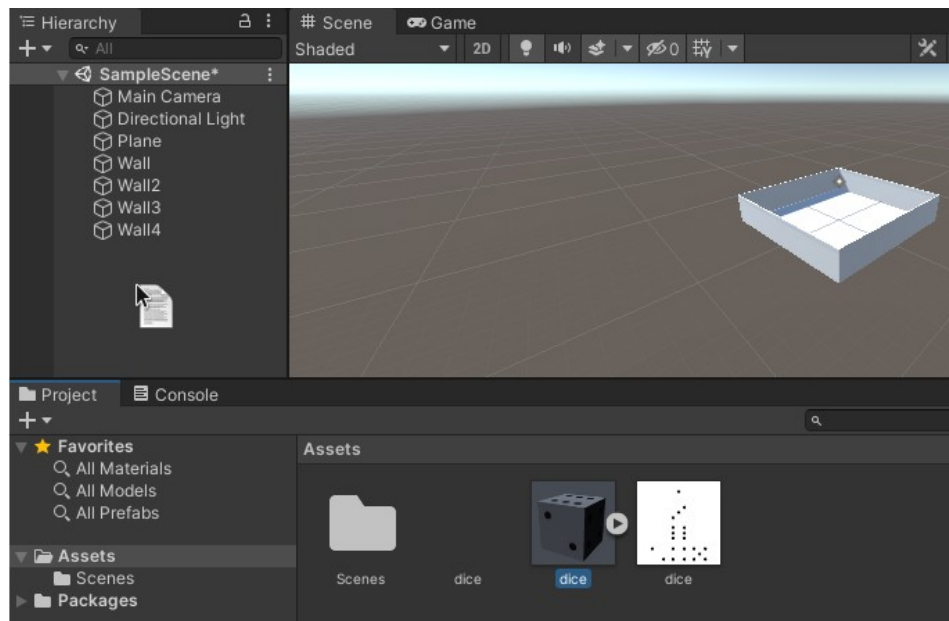


2. INCLUSIÓN DEL OBJETO DEL DADO

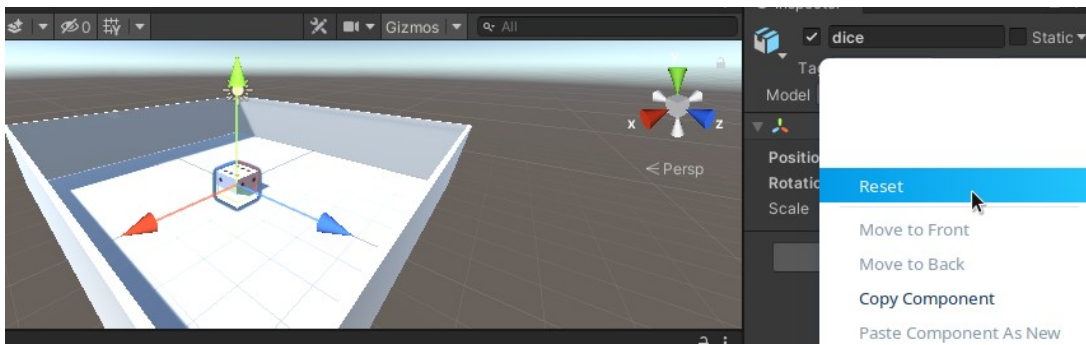
13. De esta forma hemos terminado de construir la bandeja. Ahora, Copiamos los tres archivos del objeto dado a la carpeta de **Assets**. Estos archivos son “**dice.mtl**”, “**dice.obj**” y “**dice.png**”. Se arrastra desde el explorador de archivos hasta el espacio libre de **Assets**:



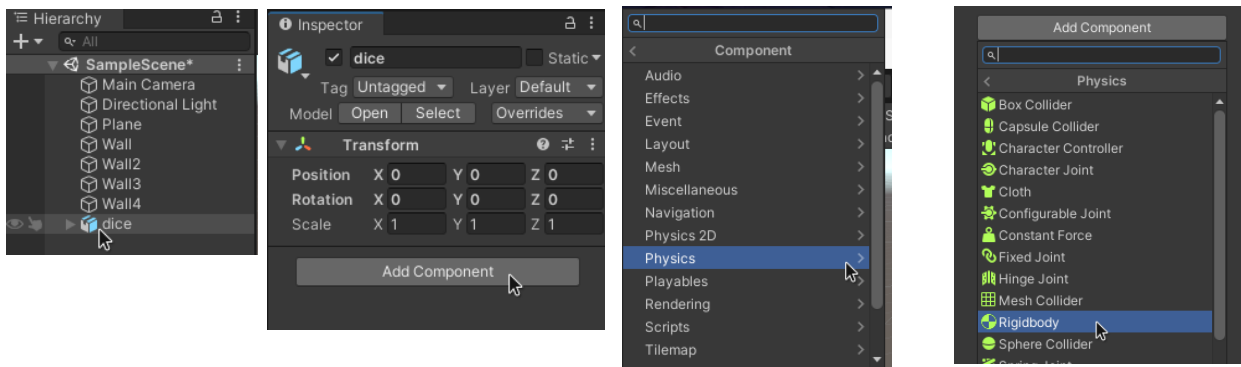
14. Arrastrar el asset “**dice.obj**” al área disponible de la ventana de **Hierarchy**:



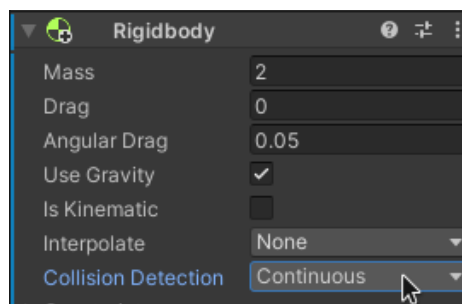
15. Reiniciar su posición, en el panel **Inspector** en los tres puntos verticales a la derecha de las propiedades de **Transform**:



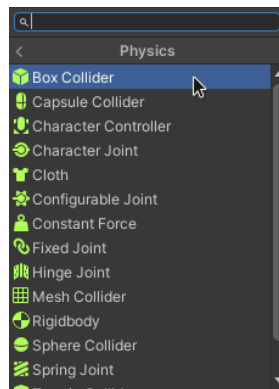
16. Adicionar un componente **Rigidbody** para el objeto **dice**. Seleccionado el objeto **dice**, clic en el botón **Add component**, del panel de **Inspector**, luego seleccionar **Physics** y entonces **Rigidbody**:



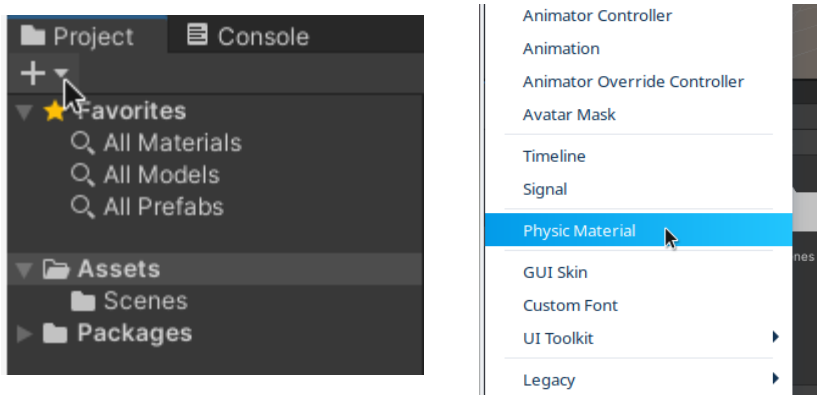
17. Configurar los parámetros de Rigidbody: **Mass = 2**, **Collision detection: Continuous**:



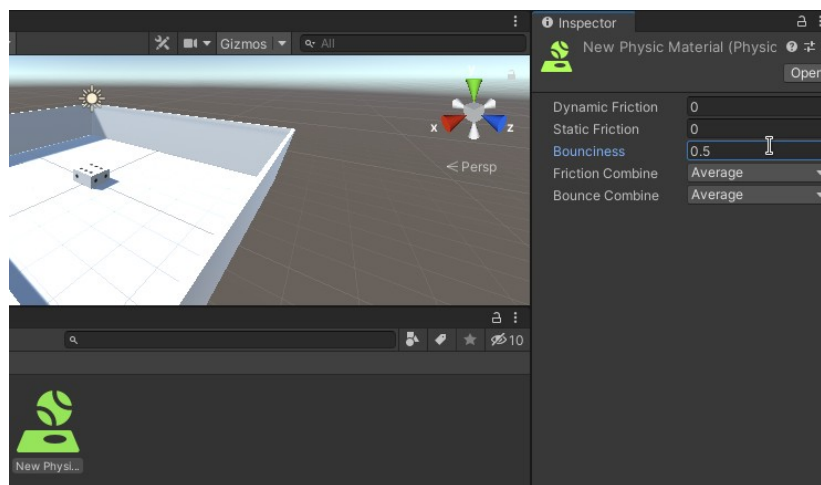
18. Adicionar un componente **Box Collider** para el objeto **dice**. Seleccionado el objeto **dice**, clic en el botón **Add component**, del panel de **Inspector**, luego seleccionar **Physics** y entonces **Box Collider**:



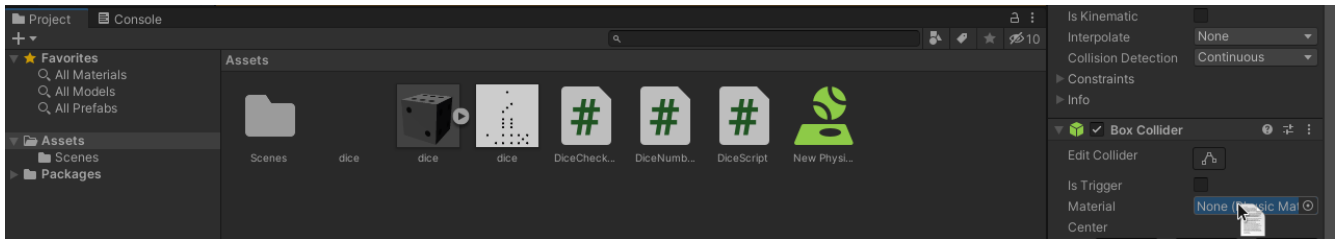
19. Agregar un **Physic Material** al proyecto. En el panel de **Project**, clic en el signo “+” y en el menú clic en **Physic Material**:



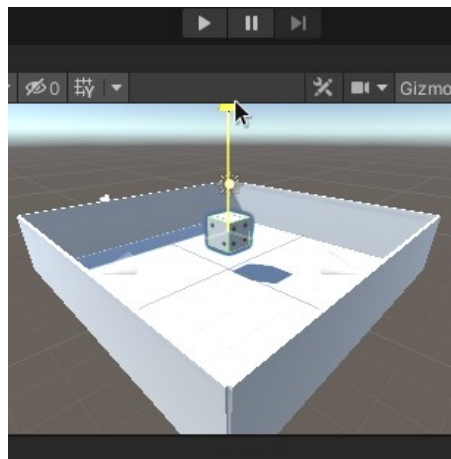
20. Configurar los parámetros de **Physic Material**: **Dynamic friction = 0**, **Static friction = 0**, **Bounciness = 0.5**:



21. Arrastrar **New Physic Material** a **Material** del **Box Collider** del objeto **dice**:

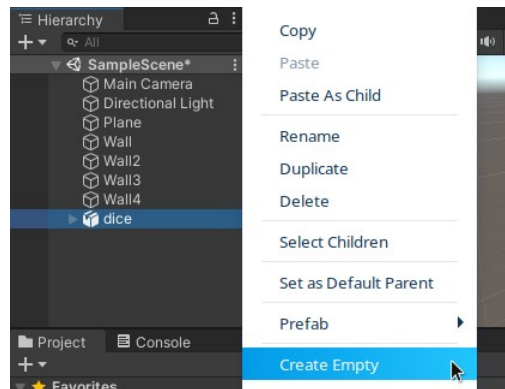


22. Alejar el dado de la bandeja en altura. Desplazar el **eje y** del dado (**eje verde**) hacia arriba:

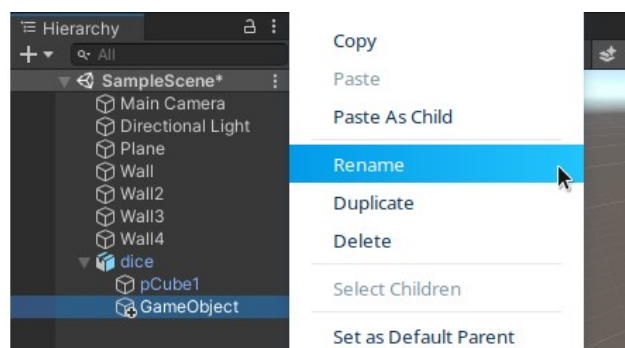


3. CONFIGURAR LOS LADOS DEL DADO

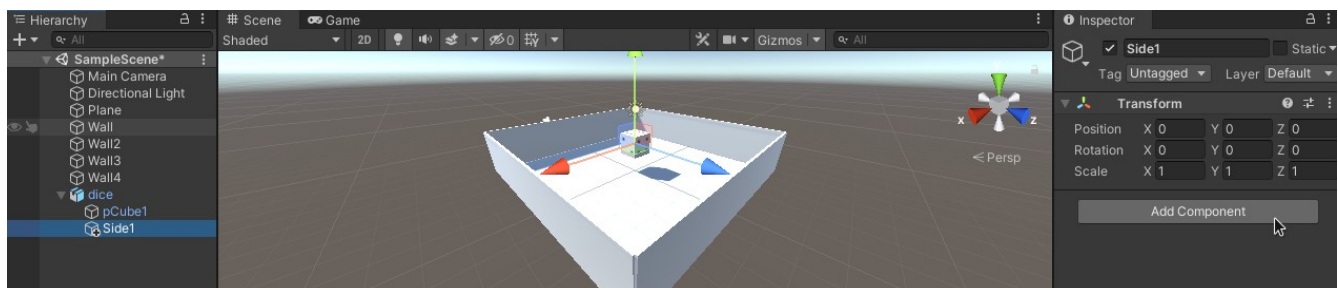
23. Crear subcomponente (**Game object**) del objeto **dice**. Clic derecho sobre el objeto **dice**, seleccionar del menú **Create Empty**:

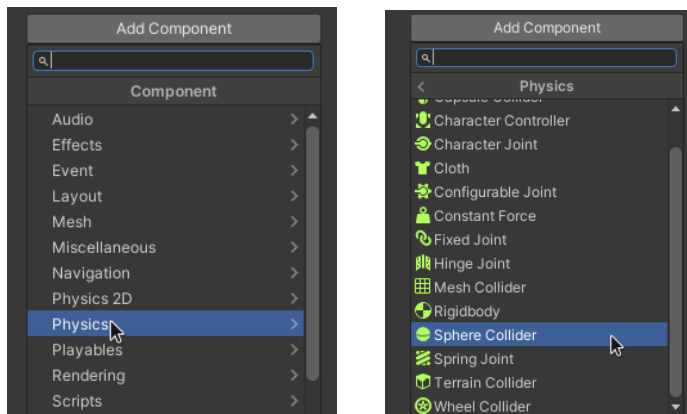


24. Renombrarlo a **Side1**:

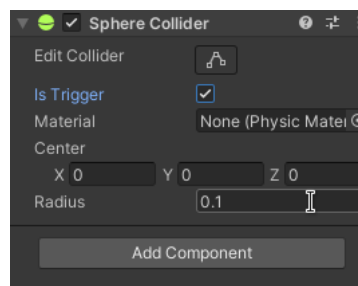


25. Adicionar un **Sphere Collider** a **Side1**. Clic en el botón **Add component**, luego en **Physics** y luego en **Sphere Collider**:

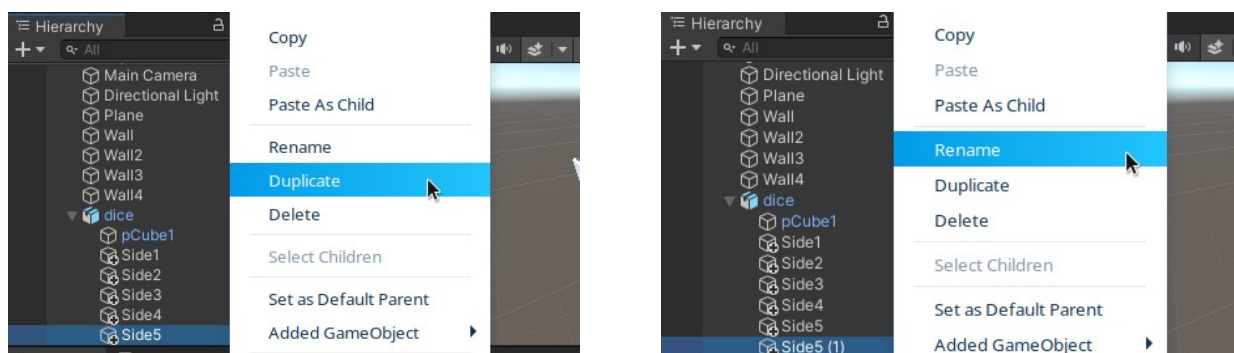




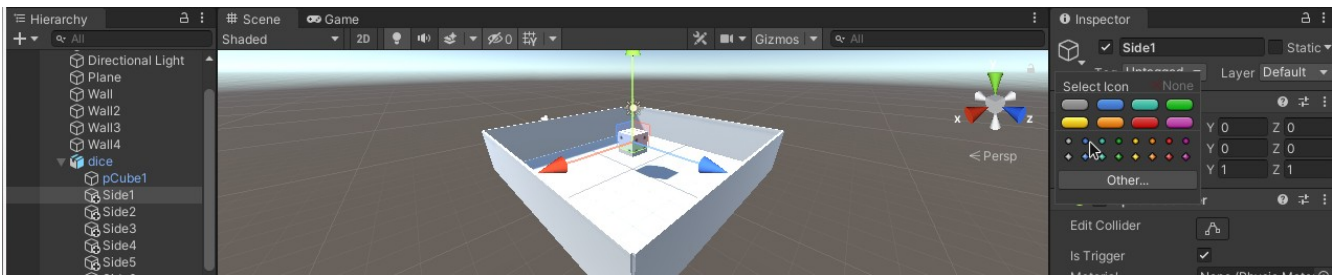
26. Configurar los parámetros: **Seleccionar Is Trigger, Radius = 0.1:**



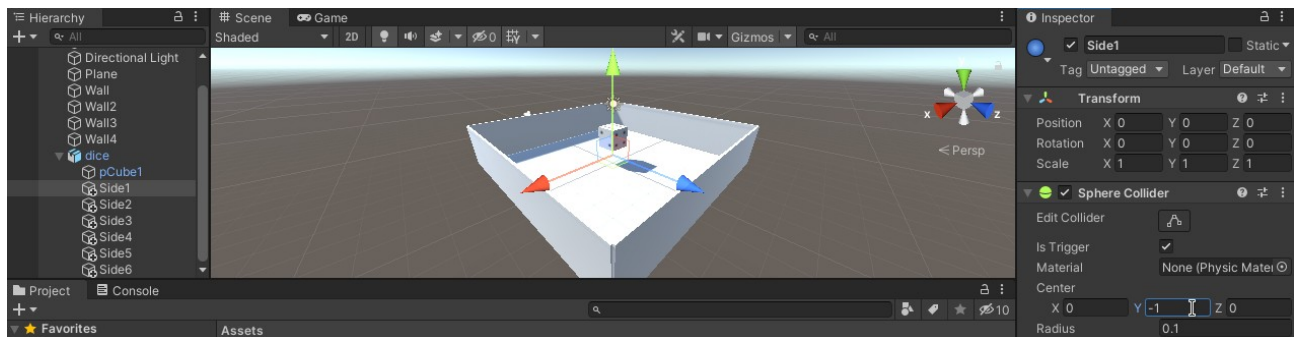
27. Copiar **Side1** cinco veces más (es decir en total seis objetos). Renombrarlos a **Side2, Side3, Side4, Side5** y **Side6**:



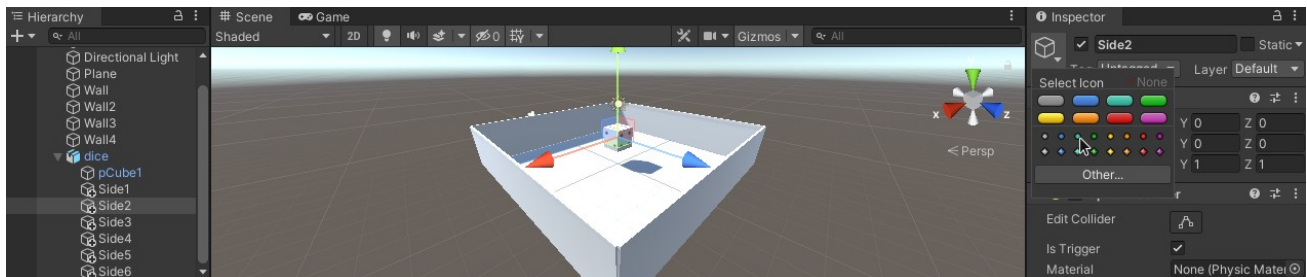
28. Al seleccionar **Side1** en el panel de **Hierarchy**, en el ícono en forma de cubo del panel **Inspector** se selecciona el pequeño ícono del punto **azul** de la primera fila de puntos:



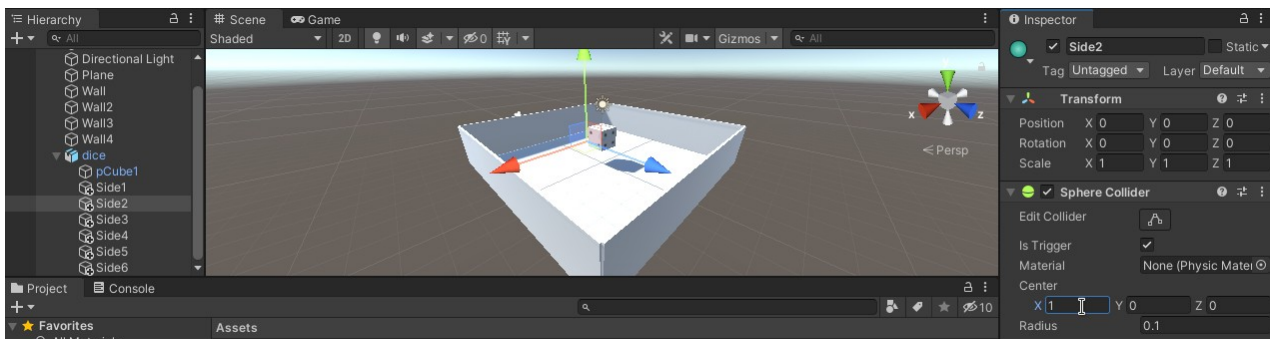
29. En la coordenada de Center $y = -1$:



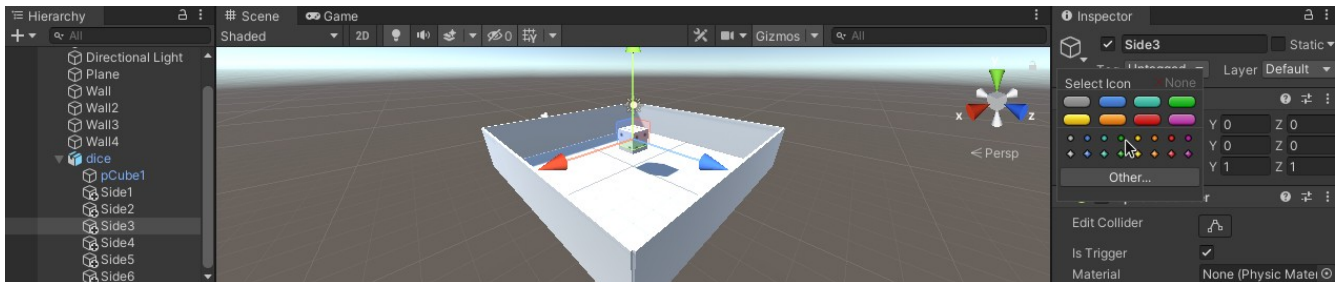
30. Al seleccionar **Side2** en el panel de **Hierarchy**, en el ícono en forma de cubo del panel **Inspector** se selecciona el pequeño ícono del punto **cyan** de la primera fila de puntos:



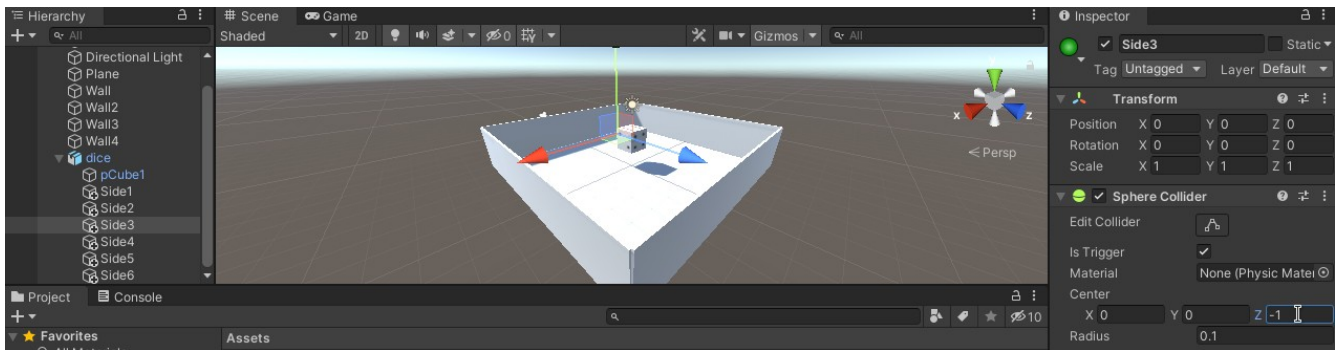
31. En la coordenada de Center $x = 1$:



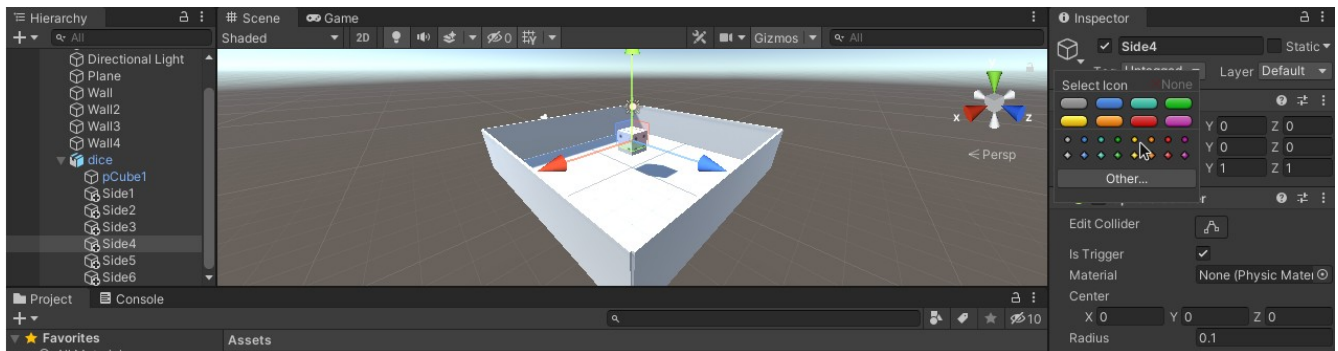
32. Al seleccionar **Side3** en el panel de **Hierarchy**, en el ícono en forma de cubo del panel **Inspector** se selecciona el pequeño ícono del punto **verde** de la primera fila de puntos:



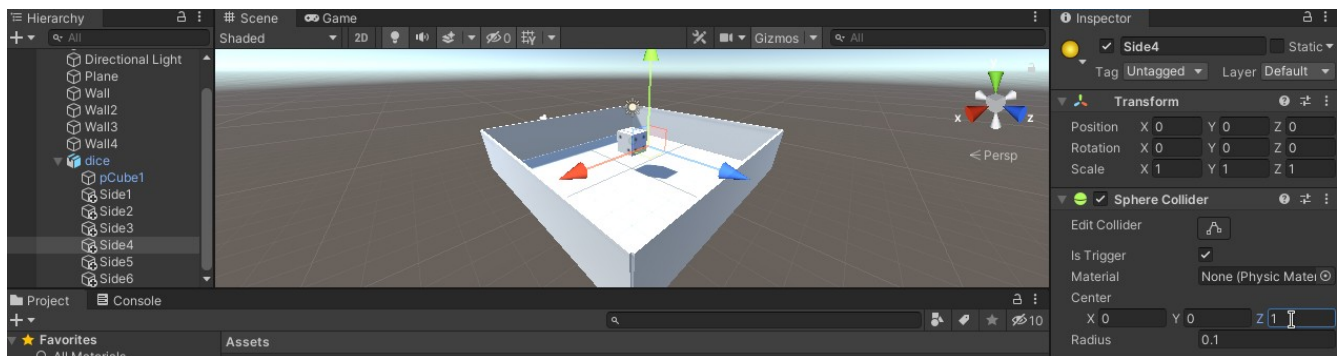
33. En la coordenada de Center **z = -1**:



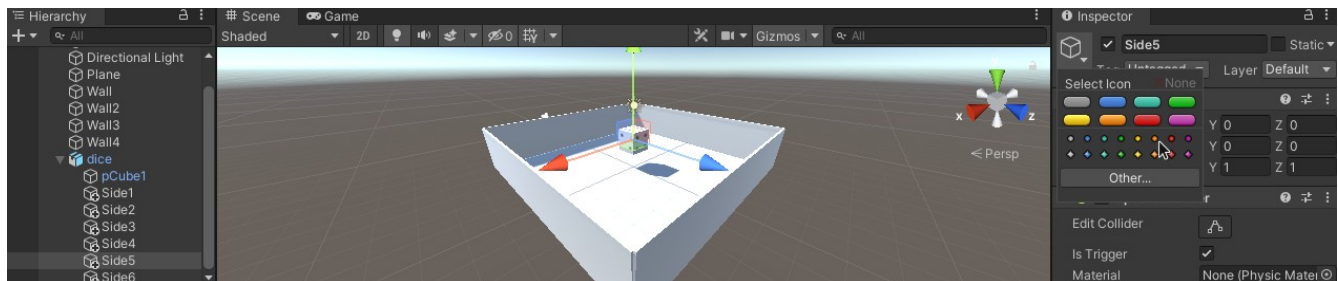
34. Al seleccionar **Side4** en el panel de **Hierarchy**, en el ícono en forma de cubo del panel **Inspector** se selecciona el pequeño ícono del punto **amarillo** de la primera fila de puntos:



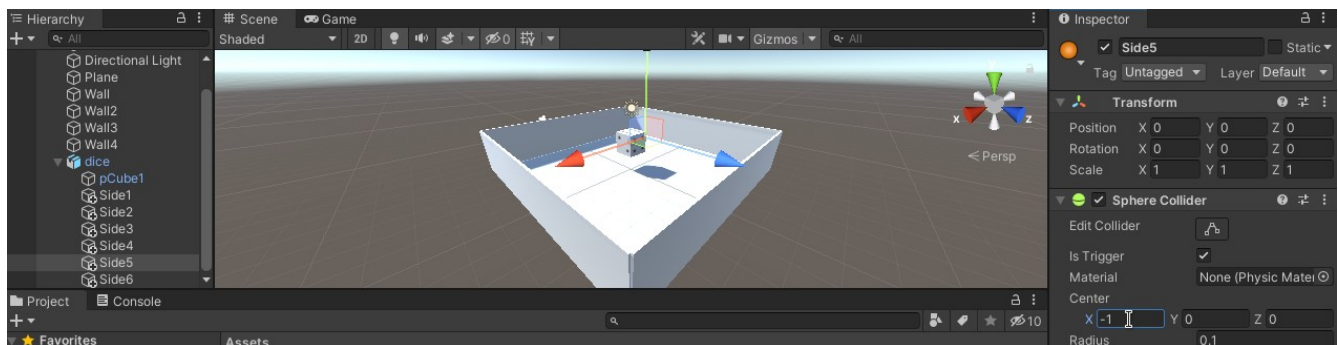
35. En la coordenada de Center **z = 1**:



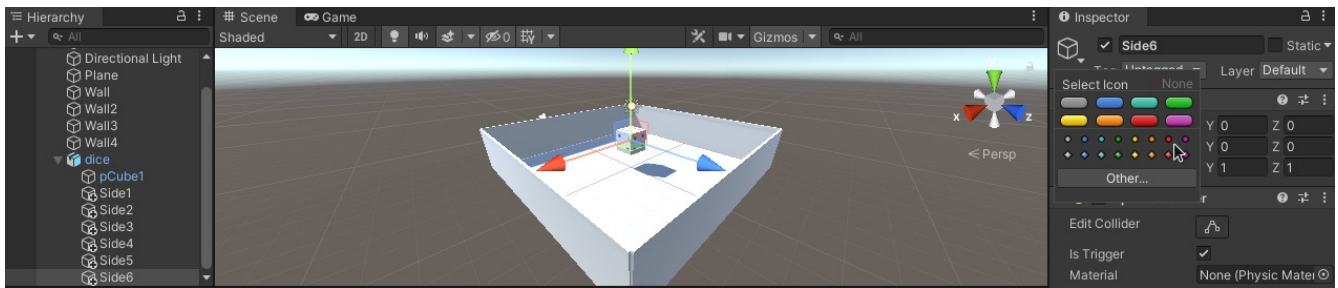
36. Al seleccionar **Side5** en el panel de **Hierarchy**, en el ícono en forma de cubo del panel **Inspector** se selecciona el pequeño ícono del punto **naranja** de la primera fila de puntos:



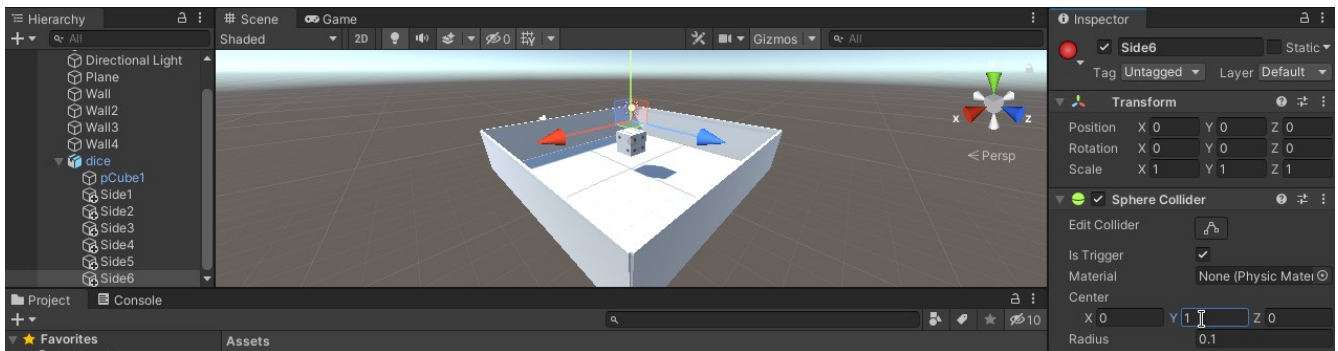
37. En la coordenada de Center **x = -1**:



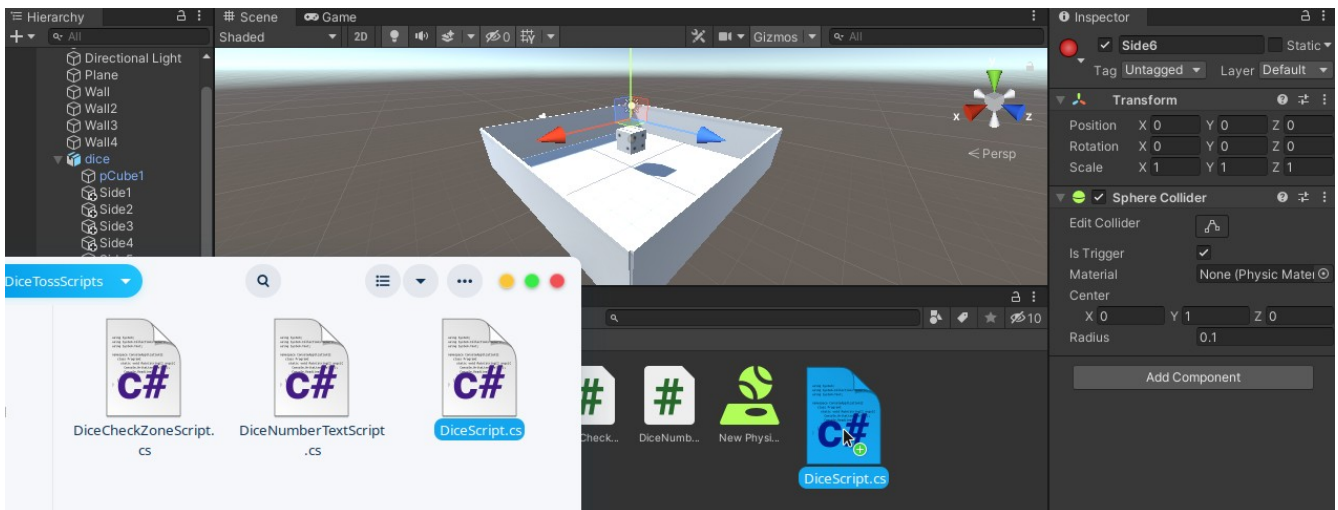
38. Al seleccionar **Side6** en el panel de **Hierarchy**, en el ícono en forma de cubo del panel **Inspector** se selecciona el pequeño ícono del punto **rojo** de la primera fila de puntos:



39. En la coordenada de Center $y = 1$:

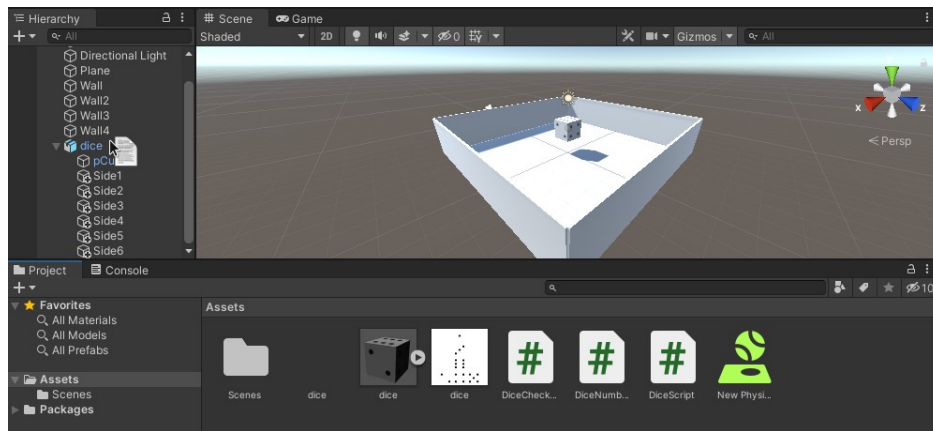


40. Copiar los tres archivos de script al área disponible de **Assets**. Los tres archivos son “DiceCheckZoneScript.cs”, “DiceNumberTextScript.cs” y “DiceScript.cs”:



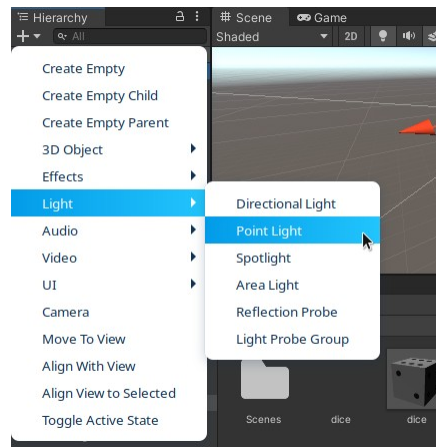
41. Arrastrar el script “**DiceScript**” al objeto **dice** (Nota: Asegúrese que el objeto sea asignado al objeto **dice**, no a ningún otro componente o sub-componente del panel **Hierarchy**):

Simular lanzamiento de un dado en Unity – JF Mendoza M.

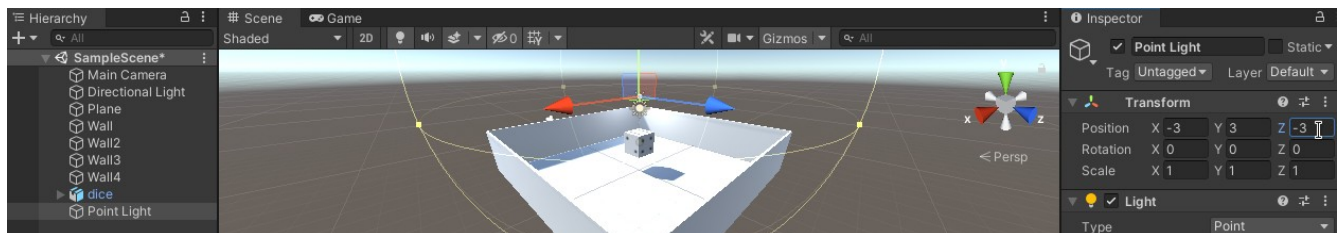


4. CONFIGURACIÓN DE LA ILUMINACIÓN DEL ESCENARIO

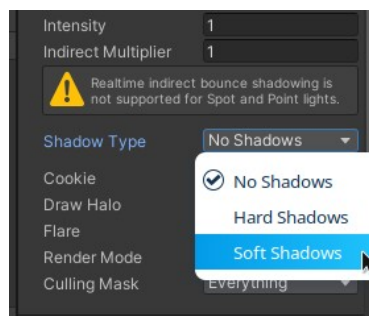
42. Crear un **Point Light**. En el panel **Hierarchy**, hacer clic en el signo “+”, escoger la opción del menú **Light**, y a continuación escoger **Point light**:



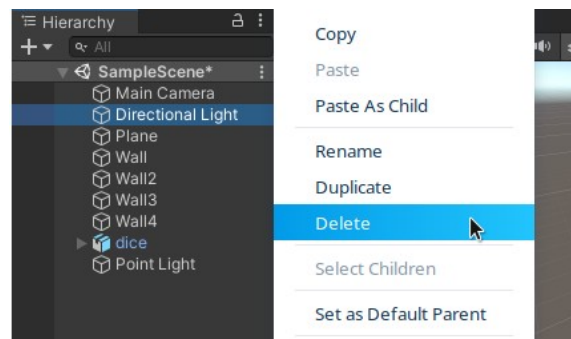
43. Configurar las coordenadas del **Point Light**: **X = -3, Y = 3, Z = -3**:



44. Escoger **Shadow Type = Soft Shadows**:



45. Borrar el objeto de **Directional Light**:

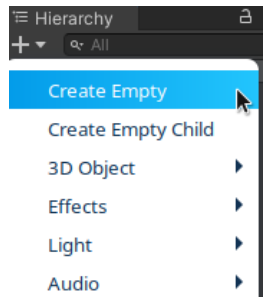


46. Señalar **Main Camera**, cambiar los parámetros de **Position** a **x = -7, y = 7, z = 0**. Parámetros de **Rotation**
x = 45, y = 90, z = 0:

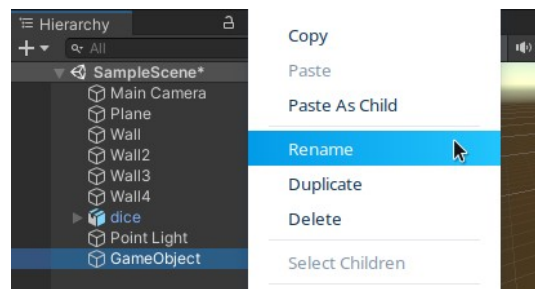


5. CONFIGURAR LA ZONA DE COLISIÓN DE LA BANDEJA

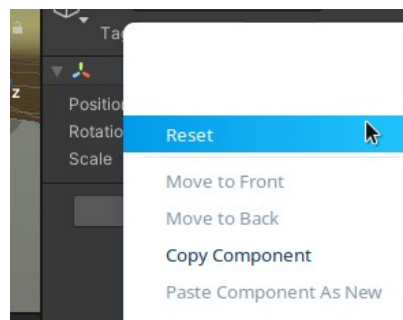
47. En el panel de **Hierarchy**, crear un GameObject vacío (**Empty**):



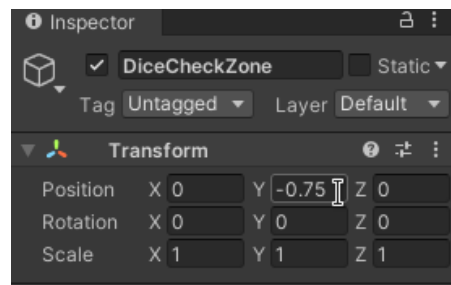
48. Renombrarlo a **DiceCheckZone**:



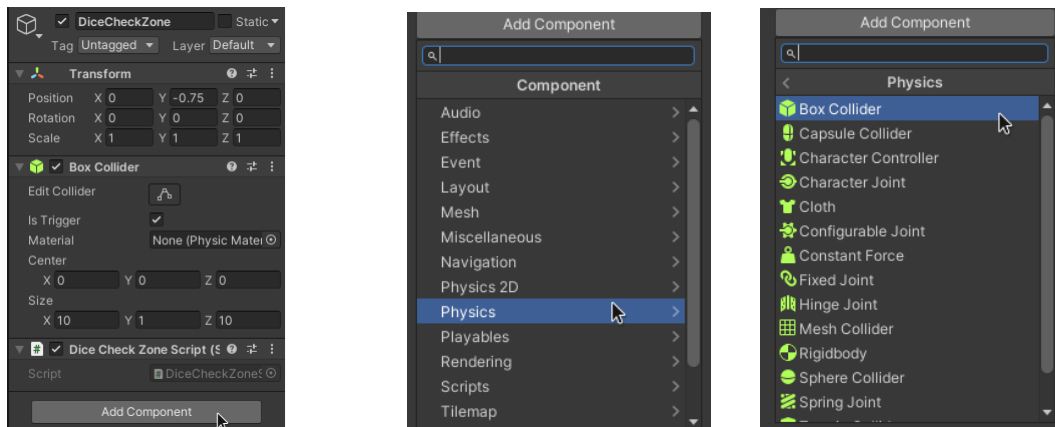
49. Reiniciar su posición:



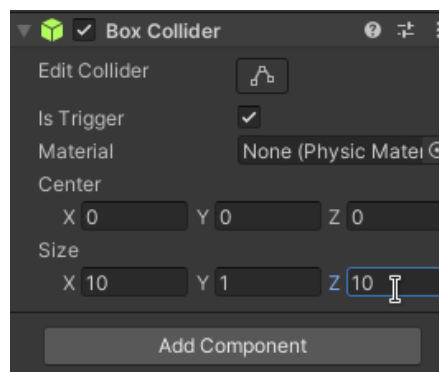
50. Cambiar su posición en **y = - 0.75**:



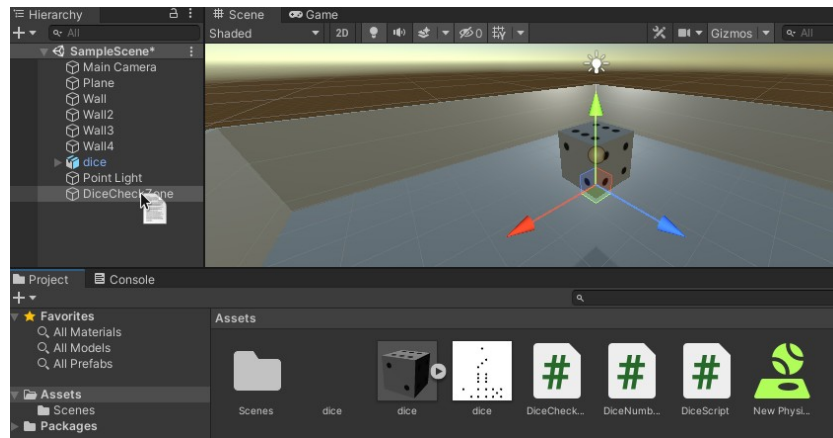
51. Adicionarle un componente **Box Collider**:



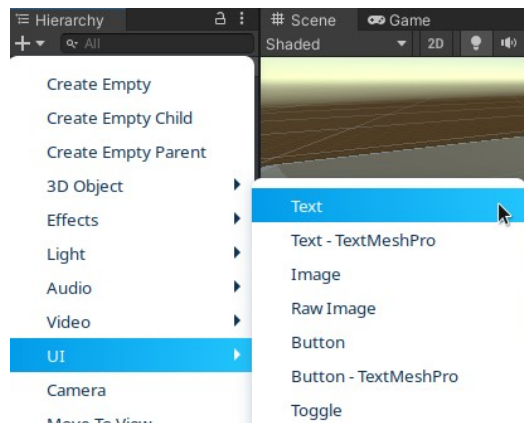
52. Configurar las propiedades del **Box Collider** (Seleccionar **Is Trigger**, Size x = 10, y = 1, z = 10):



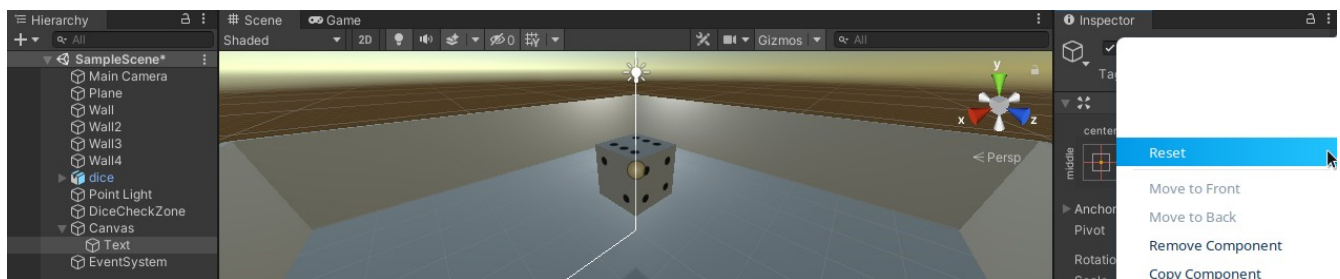
53. Asignar el script **DiceCheckZoneScript** al objeto **DiceCheckZone**:



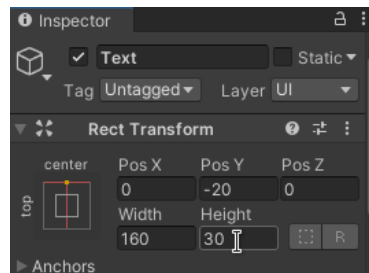
54. Crear el texto:



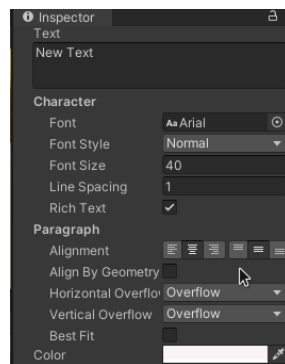
55. Reiniciar su posición:



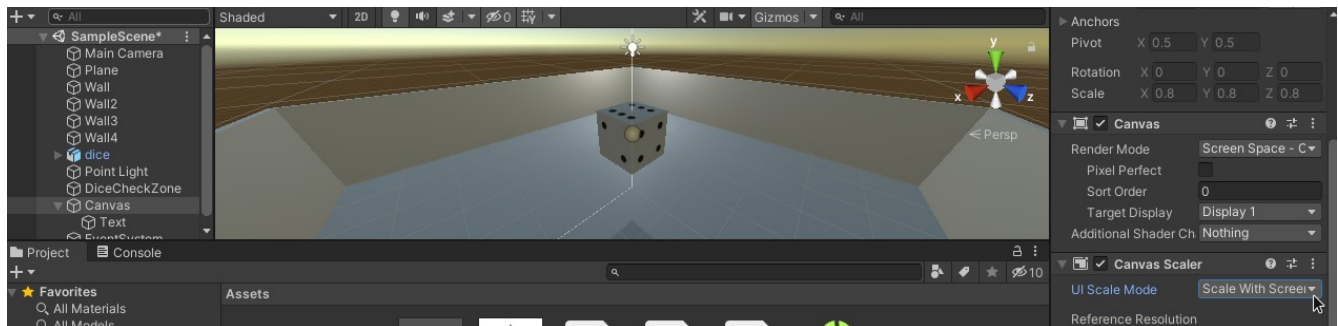
56. En sus propiedades, configurar: **Width: 160, Height: 30, Rect Transform: Top – Center:**



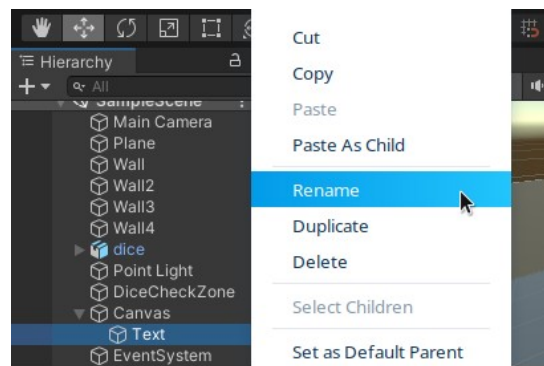
57. Configurar las propiedades del tipo de letra. **Font Size: 40, Alignment: Center, Horizontal Overflow: Overflow, Vertical Overflow: Overflow, Color: White:**



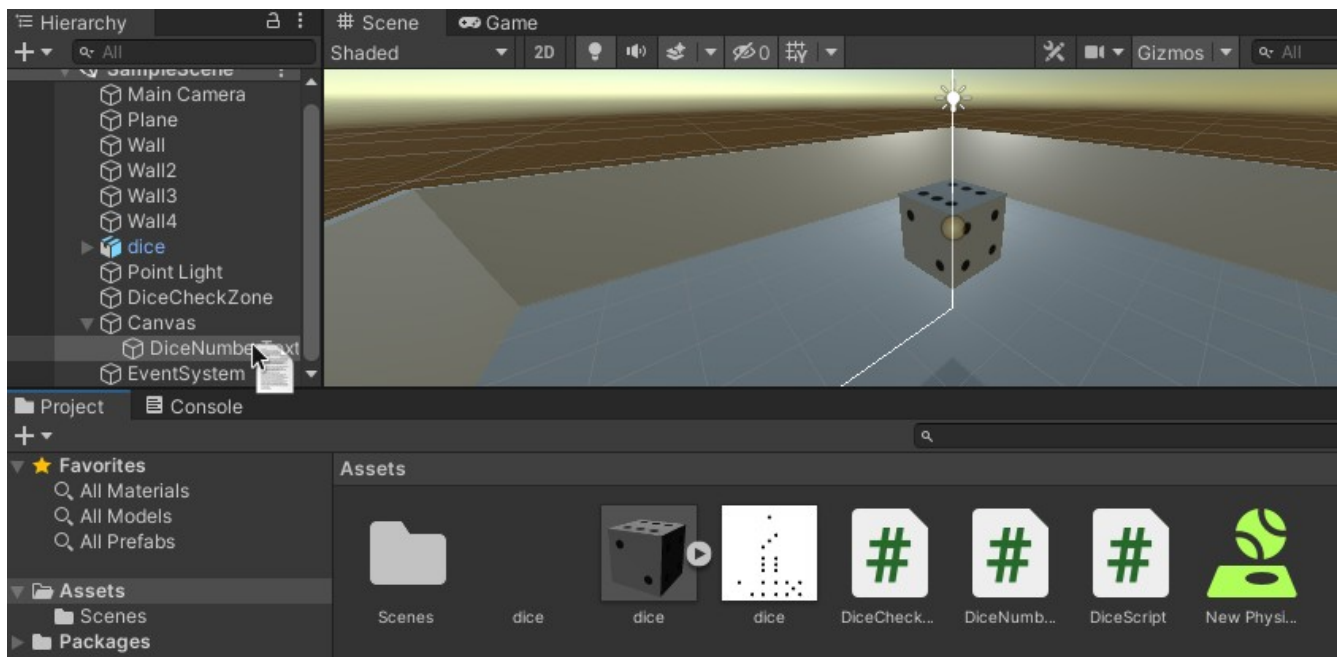
58. Clic en **Canvas**, en **Canvas Scaler** escoger la selección de **UI Scale Mode** a **Scale with Screen Size**.



59. Renombrar **Text** por **DiceNumberText**:

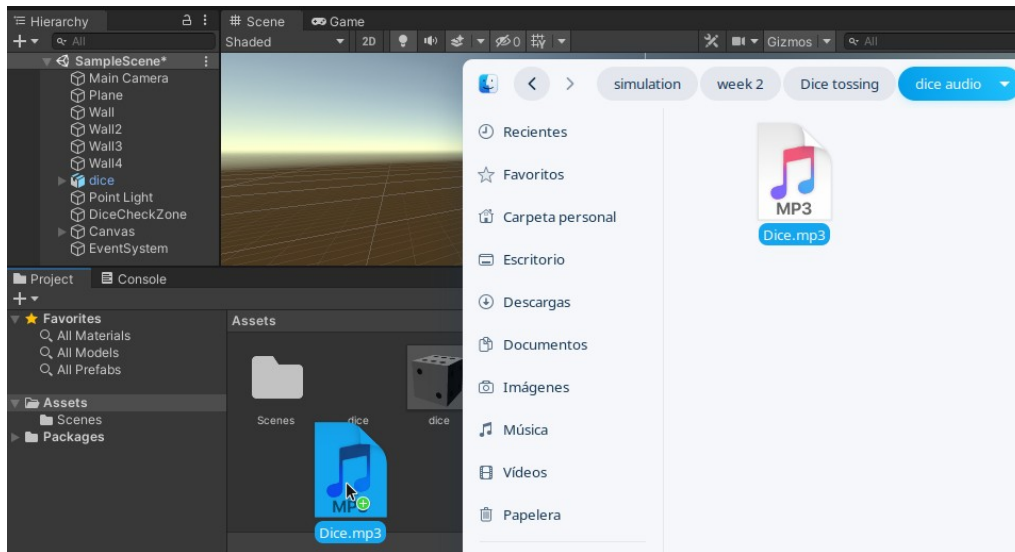


60. Arrastrar el script “DiceNumberTextScript” al objeto DiceNumberText:

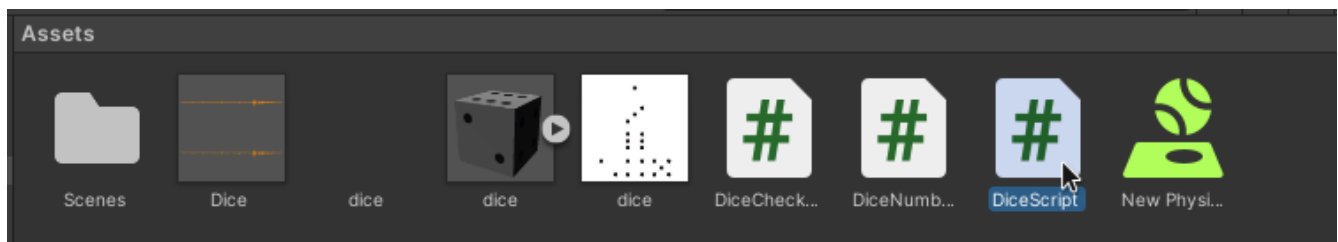


6. CONFIGURAR EL AUDIO DE LANZAMIENTO

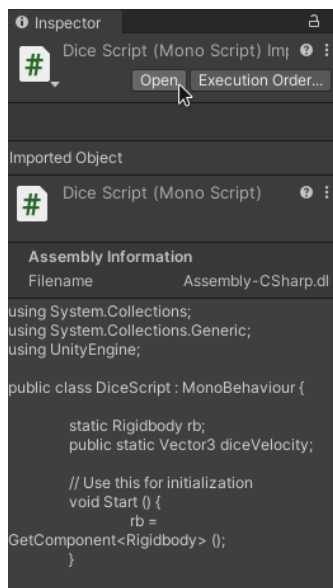
61. Arrastrar el archivo “Dice.mp3” al área disponible de **Assets**:



62. Seleccionar el script “DiceScript.cs”:



63. Hacemos clic en el botón de **Open** para abrir el script con el IDE por defecto:



64. Agregar las siguientes líneas de código después de **public static Vector3 diceVelocity;** :

```
public AudioSource diceEmitter;
```

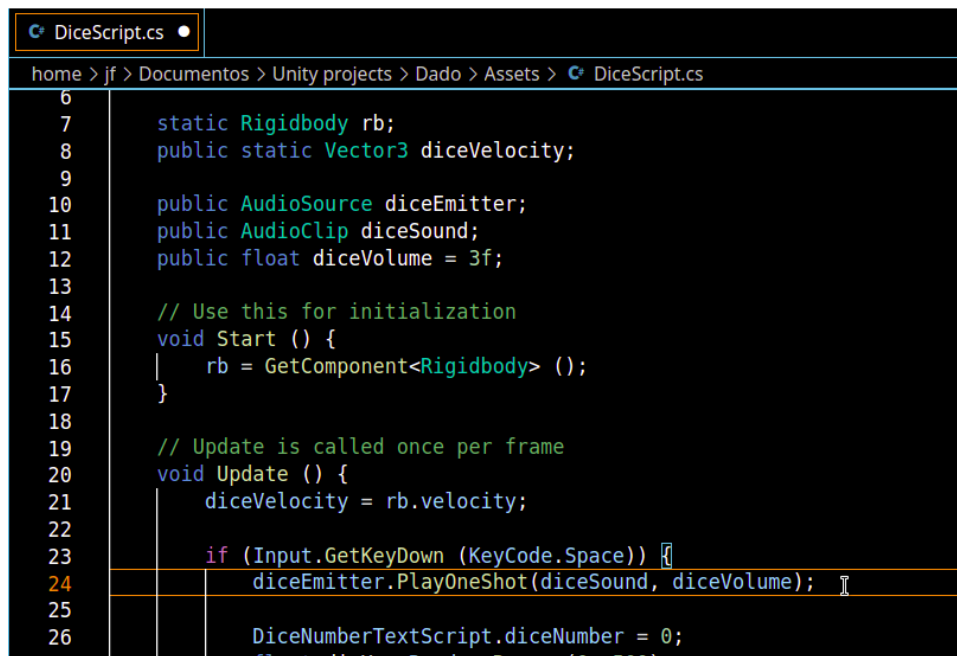
```
public AudioClip diceSound;
```

```
public float diceVolume = 3f;
```

```
DiceScript.cs
home > jf > Documentos > Unity projects > Dado > Assets > DiceScript.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class DiceScript : MonoBehaviour {}
6
7      static Rigidbody rb;
8      public static Vector3 diceVelocity;
9
10     public AudioSource diceEmitter;
11     public AudioClip diceSound;
12     public float diceVolume = 3f;
13
14     // Use this for initialization
15     void Start () {
16         rb = GetComponent<Rigidbody> ();
17     }
18
```

65. Agregar esta línea de código después de **if (Input.GetKeyDown (KeyCode.Space)) { :**

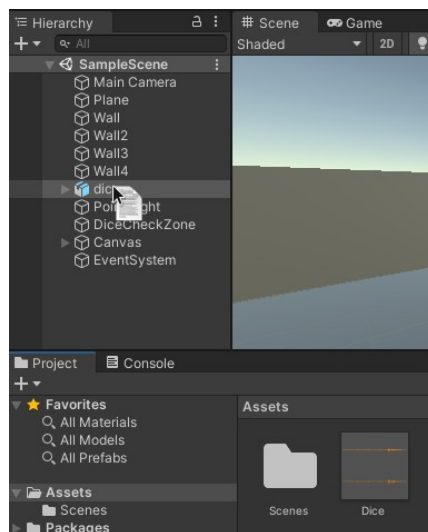
```
diceEmitter.PlayOneShot(diceSound, diceVolume);
```



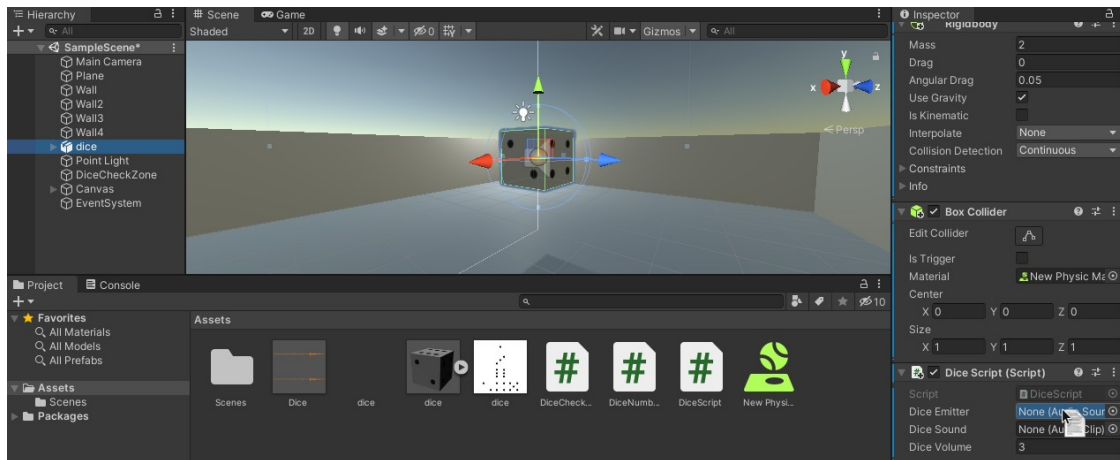
```
DiceScript.cs
home > jf > Documentos > Unity projects > Dado > Assets > DiceScript.cs
6
7 static Rigidbody rb;
8 public static Vector3 diceVelocity;
9
10 public AudioSource diceEmitter;
11 public AudioClip diceSound;
12 public float diceVolume = 3f;
13
14 // Use this for initialization
15 void Start () {
16 | rb = GetComponent<Rigidbody> ();
17 | }
18
19 // Update is called once per frame
20 void Update () {
21 | diceVelocity = rb.velocity;
22 |
23 | if (Input.GetKeyDown (KeyCode.Space)) {
24 | | diceEmitter.PlayOneShot(diceSound, diceVolume);
25 | |
26 | DiceNumberTextScript.diceNumber = 0;
27 | }
```

66. Guardar el archivo y cerrar el IDE.

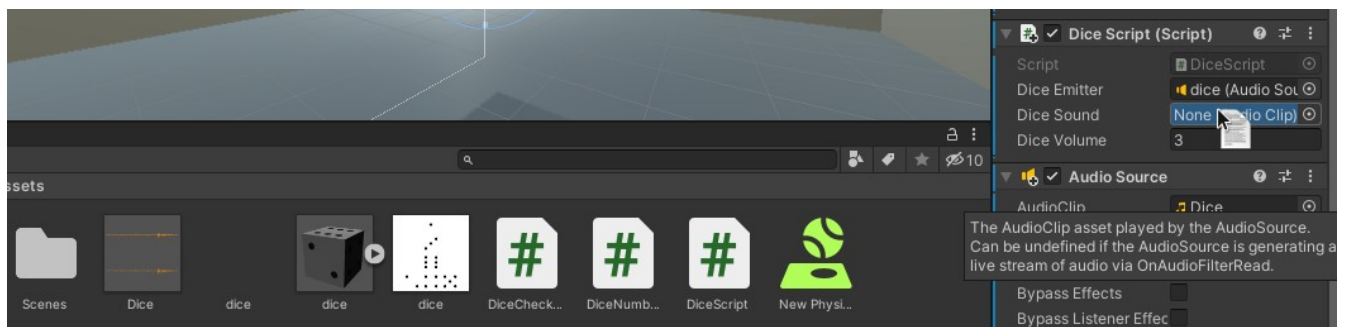
67. Arrastrar el audio al objeto **dice**:



68. Seleccionar el objeto **dice** del panel de **Hierarchy** a la propiedad **Dice Emitter** de las variables públicas de **Dice Script**:



69. Arrastrar el objeto de audio **Dice** de la ventana de **Assets** a la variable **Dice Sound**:



REFERENCIAS

- ZOTOV, Alexander. *How to create simple Dice Tossing game with Unity? Dice Roll Easy Unity 2D tutorial (Video)*. Recuperado de <https://www.youtube.com/watch?v=ouGJJzNPsSk> el 21 de marzo de 2022.
- CONTENIDO AP. *Unity Básico - Como emitir un sonido al destruir un objeto (Video)*. Recuperado de <https://www.youtube.com/watch?v=j0qTU49rnQY> el 21 de marzo de 2022.
- SoundEffectsFactory. *Dice Roll Sound Effect ~ Rolling dice sound (Audio)*. Recuperado de <https://www.youtube.com/watch?v=dEHggEjNsms> el 21 de marzo de 2022.
- Unity Technologies. *Manual de Unity versión 5.3*. Recuperado de <https://docs.unity3d.com/es/530/Manual/UnityManual.html> el 21 de marzo de 2022.