«Talento Tech»

Videojuegos

Clase 05

Buenos Aires Ciudad

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Clase N° 5 | Conceptos básicos

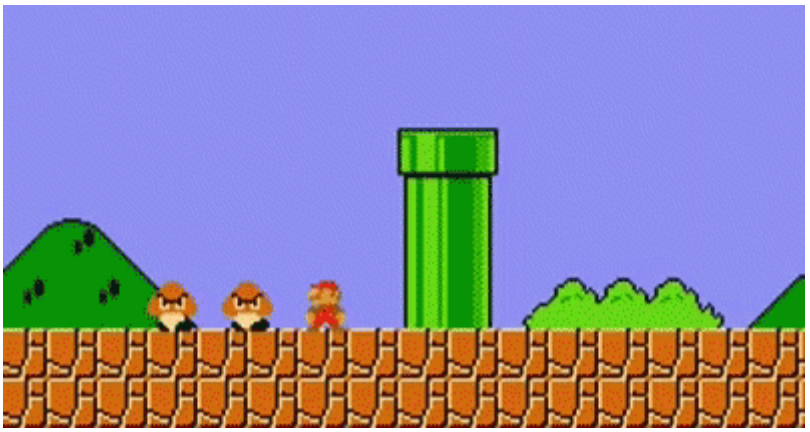
Temario:

- GameObjects
- Scripts
- Funciones
- Condicionales
- Inputs
- Etiquetas

Buenos Aires aprende
Agencia de Habilidades para el Futuro

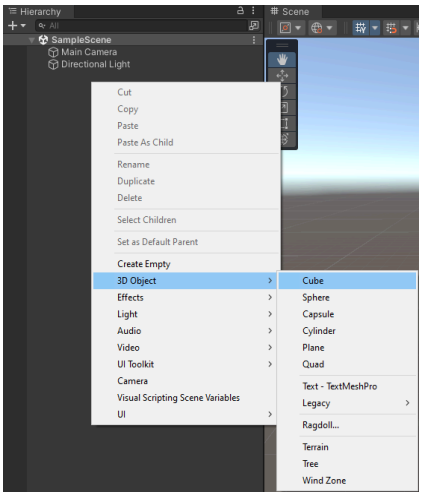
Repaso

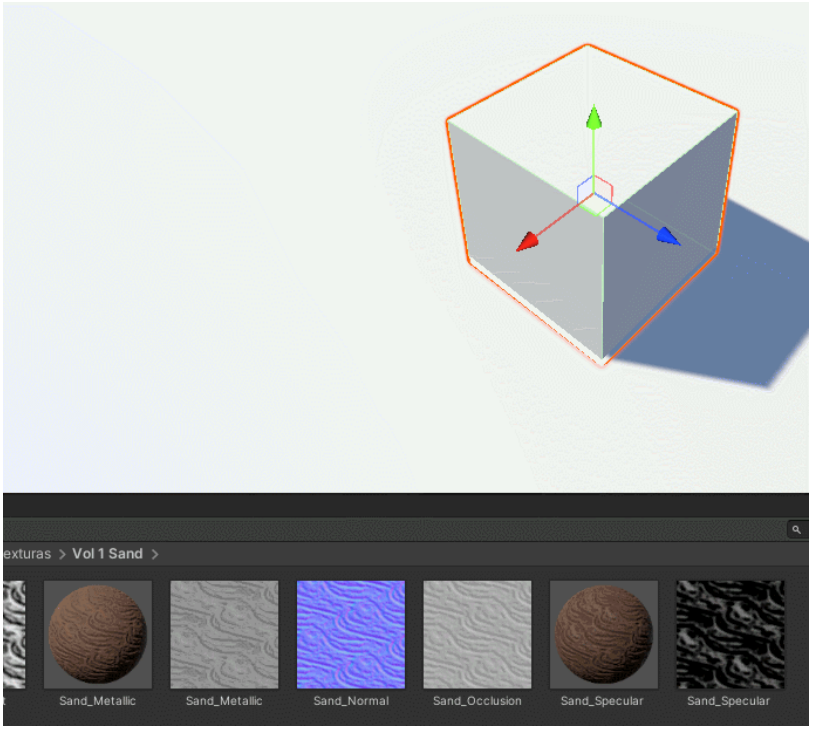
Hoy **integraremos** todos los **conceptos** aprendidos hasta ahora para crear un obstáculo que se desplace de un lado a otro y que, al entrar en contacto con él, haga que nuestro personaje pierda vida.



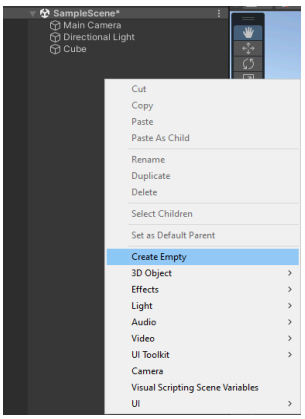
Objetos

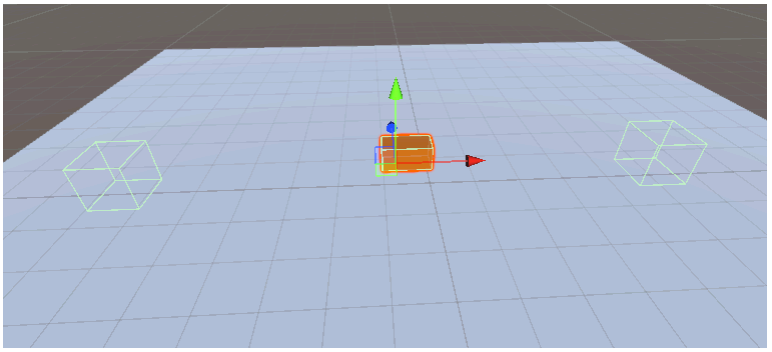
Crearemos un **3D Object** del tipo **cubo** y le agregaremos un **material** a preferencia.





Luego dos **Empty Objects** agregándoles un **componente Collider** a cada uno y los colocaremos en dos extremos opuestos, para que nuestro objeto “rebote” entre ellos.

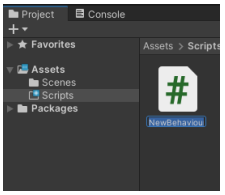




Con nuestros objetos preparados, pasemos a nuestro código:

Scripts

Pero primero creamos un **script,** que vamos a utilizarlo para el movimiento de obstáculos:



Funciones y condicionales

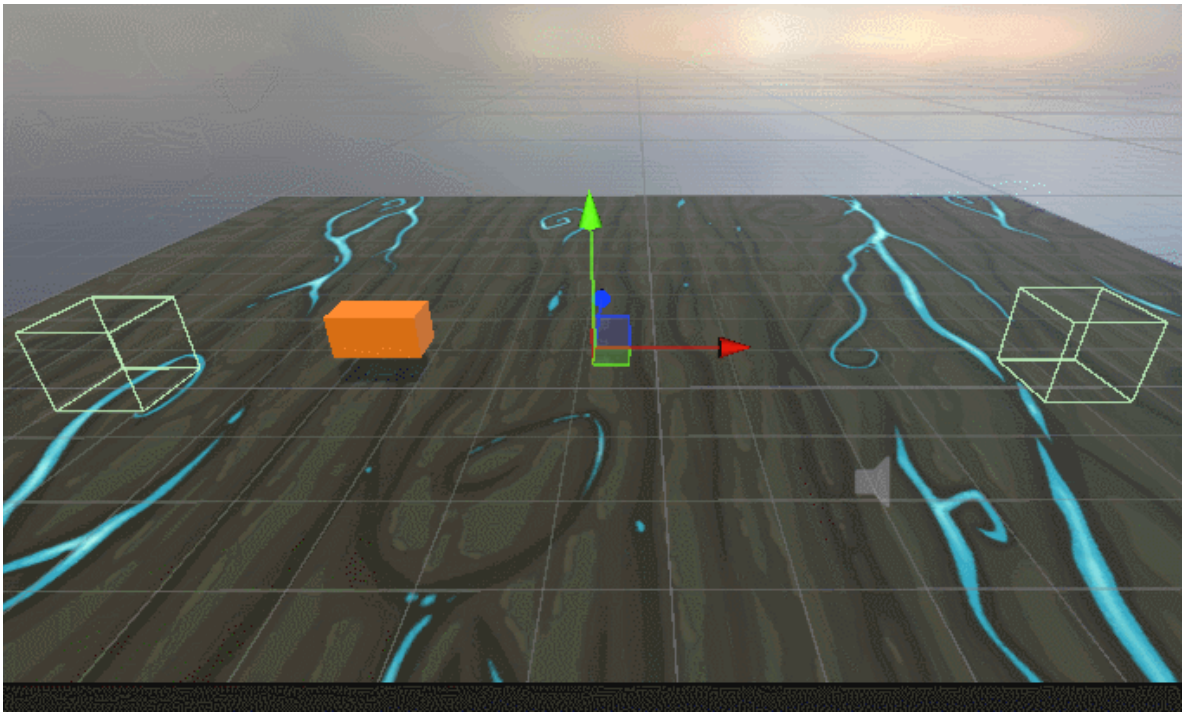
```
public float dir = 1f;

public float speed = 3f;

void Update(){

    transform.Translate(new Vector3(dir, 0, 0) * Time.deltaTime * speed);

}

private void OnTriggerEnter(collider other){
```

```
    if (other.CompareTag("PosAb")){

        dir *= (-1);

    }

}
```

Utilizaremos un **Translate()** dentro de nuestro **Update()** para el movimiento constante de nuestro obstáculo. La variable **"dir"**, puesta dentro de nuestro **Vector3** en el eje **X** será la responsable de dirigir nuestro objeto. Recordemos que el Translate() funciona de una manera en que desplaza o "traslada" al **GameObject** en la dirección asignada, y en nuestro caso, lo multiplicamos por la variable **speed** para modificar su velocidad.

Utilizando la función **OnTriggerEnter()**, haremos que cada vez que el **objeto colisione** con un **Empty** cuya etiqueta sea **"PosAB"**, cambiará la variable **"dir"** del **Vector3**, haciendo que se dirija al lado opuesto.

Tengan en cuenta que en este caso estamos **modificando** solamente al eje **X** para lograr un movimiento **horizontal**, si ustedes desean que se mueva en alguna **otra dirección** deberán de cambiar también los ejes **Y** o **Z**.



Ahora continuaremos con **nuestro personaje** con el código de movimiento seleccionado, empezando por definir las variables:

Crearemos dos funciones llamadas **“DirX()”** y **“DirZ()”** que serán las encargadas de asignar la dirección del movimiento de nuestro personaje.

```
void DirX(){
    if(Input.GetKey(KeyCode.A)) //Si apriero la tecla A
    {
        //Le asigno el valor de mi direccion en X
        //Recordemos que X negativo es para la izquierda
        movimientoX = (-1f);
    } else if (Input.GetKey(KeyCode.D)){
        movimientoX = 1f;
    }
}
void DirZ(){
```

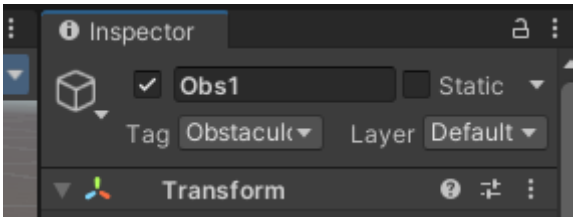
```
    if (Input.GetKey(KeyCode.W)) //Si aprieto la tecla W
    {
        movimientoZ = 1f;
    } else if (Input.GetKey(KeyCode.S)){
        movimientoZ = (-1f);
    }
}
```

Una vez armadas, las llamaremos desde nuestra **función *Update()*** y asignaremos nuestro **Vector3, "dir".** Esto lo haremos para poder "normalizarlo" **(Entendemos normalizar como tomar un vector de cualquier longitud y, mientras sigue apuntando en la misma dirección, cambiar su longitud a 1. Es decir, reduciremos su valor a 1, manteniendo su dirección.)** y que no vaya más rápido en el caso de ir en diagonal. Por último verán que estamos utilizando **"transform.Translate"** en vez de **AddForce**. Esto es para dar otra perspectiva diferente de sus usos.

Ya tenemos nuestro obstáculo y personaje. Ahora necesitamos programar la interacción entre ellos.

Tags o Etiquetas

Empezaremos por crear y asignar la etiqueta **"Obstáculo"** a nuestro objeto.



Buenos Aires aprende
Agencia de Habilidades para el Futuro

Segundo colocaremos la función **OnTriggerEnter()** para **detectar contacto** con nuestro **personaje**, en este caso con los objetos que cumplan con la **condición**: **GameObject** cuya **etiqueta** sea igual a **"Obstaculo"**.

```
private void OnTriggerEnter(Collidr other){
   if (other.gameObject.CompareTag("Obstaculo")){

   }
}
```

Seguiremos con la **definición** de una **variable Global**, siendo **"vida"**, que será **reducida** al **contacto** con nuestro obstáculo, es decir, dentro de la **condición** armada.

🤖 **RECORDÁ QUE:**

Debe ser una variable Global y hay que definirla arriba justo después del comienzo de la definición de una clase o class, como la mayoría de nuestras variables.

```
Script de Unity (1 referencia de recurso) | 0 referencias
public class MovCol : MonoBehaviour
{
    public int vida = 10;
```

⭐**¡Ya casi estamos!**⭐Ahora nos queda restar nuestra vida y debemos asegurarnos de que nuestro personaje **se destruya** al llegar a **"0 de vida"**.

```
private void OnTriggerEnter(Collidr other){
   if (other.gameObject.CompareTag("Obstaculo")){
      vida--;
      Debug.Log("Daño recibido. Vida Restante: " + vida);
      if(vida <= 0){
         Destroy(gameObject);
      }
   }
}
```

Con esto ya tendremos un sistema sencillo donde nuestro personaje debe evitar un obstáculo para pasar al otro lado sin perder vida.

🎮 **DATAZO SOBRE DISEÑO DE VIDEOJUEGOS:** En ciertos juegos, la vida no solo es una métrica de juego o información que se va a ver en pantalla, sino también un elemento narrativo. La historia puede estar vinculada a la salud del personaje, y eventos importantes pueden ocurrir cuando la vida alcanza ciertos niveles críticos.

En uno de los juegos donde mejor lo vemos reflejado es en el videojuego *The Walking Dead (2012)* perteneciente a la ex desarrolladora de videojuegos *Telltale Games* donde la toma de decisiones tomará un papel crucial para la progresión de la narrativa junto a la interacción con el entorno.

🎥 **Podés ver un trailer acá:** The Walking Dead: The Game 🎥

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Desafío N° 5:

Con lo hecho en clase: Duplicá los obstáculos y distribuirlos por el escenario para hacer un mini nivel. Poné a prueba tus habilidades de diseño.

Aprovechá para practicar lo visto hasta ahora.

No te olvides de **guardar** todos los cambios.

Tomá una **captura de pantalla** del nivel.

Subilas al espacio correspondiente del Desafío 5.


Buenos Aires
aprende
Buenos
Aires
Ciudad