



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

**Centro de
e-Learning**

Secretaría de Extensión y Cultura Universitaria

III.12. UNIDAD DIDÁCTICA

<< Desarrollo de Videojuegos >>



Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

www.sceu.frba.utn.edu.ar/e-learning

<< Modulo III Interacción >>

Interacción y sonido



Presentación:

Para que un objeto pueda interactuar con su entorno y con otros objetos, es necesario que siga una serie de leyes físicas, como puede ser responder a una fuerza de rozamiento, durante un movimiento. En esta unidad nos introduciremos en la forma en la cual Unity trabaja con interacciones entre cuerpos y luego pasaremos a ver como incorporar sonido a nuestros trabajos.



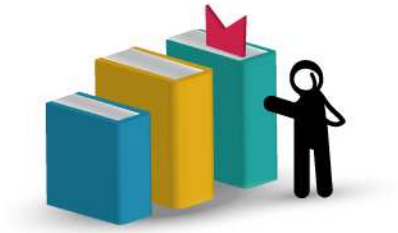
Objetivos:

Que los participantes logren:

Comprender el concepto de cuerpo rígido

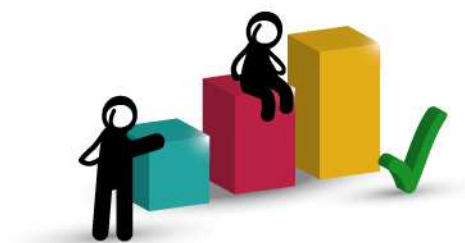
Entender cual es la metodología para agregar interacciones

Puedan agregarle sonido a sus trabajos



Bloques temáticos:

- 1.- Rigidbody
- 2.- Incorporar movimiento a puerta – Blender
- 3.- Sonido



Consignas para el aprendizaje colaborativo

En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.

** El MEC es el modelo de E-learning constructivista colaborativo de nuestro Centro.*



Tomen nota*

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.

**** Está página queda como está. El contenidista no le quita ni le agrega nada.***

Bloque temático 1: Rigidbody

En el mundo real, los objetos se encuentran sometidos a leyes físicas, cuando agregamos GameObjects a una escena de Unity, estos objetos no tienen en cuenta dichas leyes, sin embargo esto cambia cuando les asociamos un Rigidbody. Estos nuevos componentes son la forma en la cual Unity permite por ejemplo, simular los efectos de la gravedad y de fuerzas de rozamiento.

Para ver como funcionan, agreguemos primero un cubo a nuestra imagen:

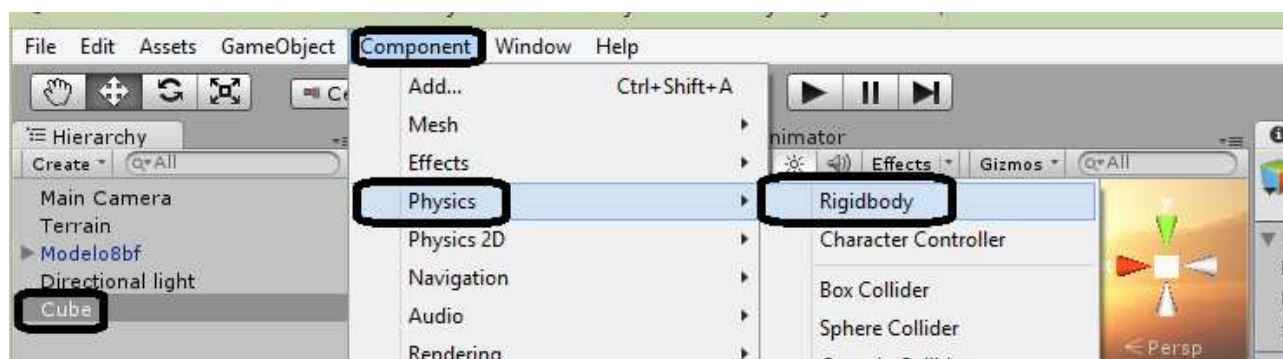


Como podemos ver al pasar a modo juego, el cubo permanece en el lugar en el cual lo ubicamos, y los componentes asociados al mismo en el panel Inspector son:

- 1.- Transform
- 2.- Cube
- 3.- Box Collider
- 4.- Mesh Renderer

Si ahora le agregamos un Rigidbody desde:

Component > Physics > Rigidbody



El cubo ahora responde a leyes físicas y cae sobre la superficie del terreno como si la fuerza de la gravedad estuviera actuando sobre él, como se muestra a continuación:



El menú correspondiente al Rigidbody tiene los siguientes campos:

Mass: la masa es una medida de la cantidad de materia que posee un cuerpo, no debe confundirse masa con peso.

Drag: es la resistencia del aire lineal.

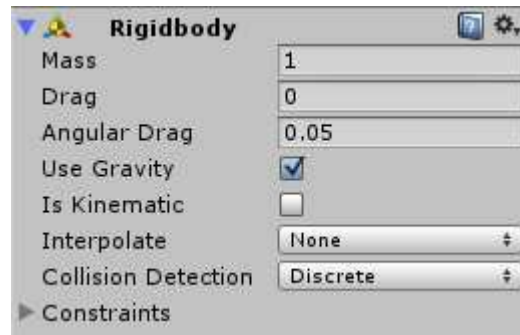
Angular Drag: es la resistencia al aire durante un movimiento de rotación.

Use Gravity: Indica si el cuerpo responde o no a la gravedad.

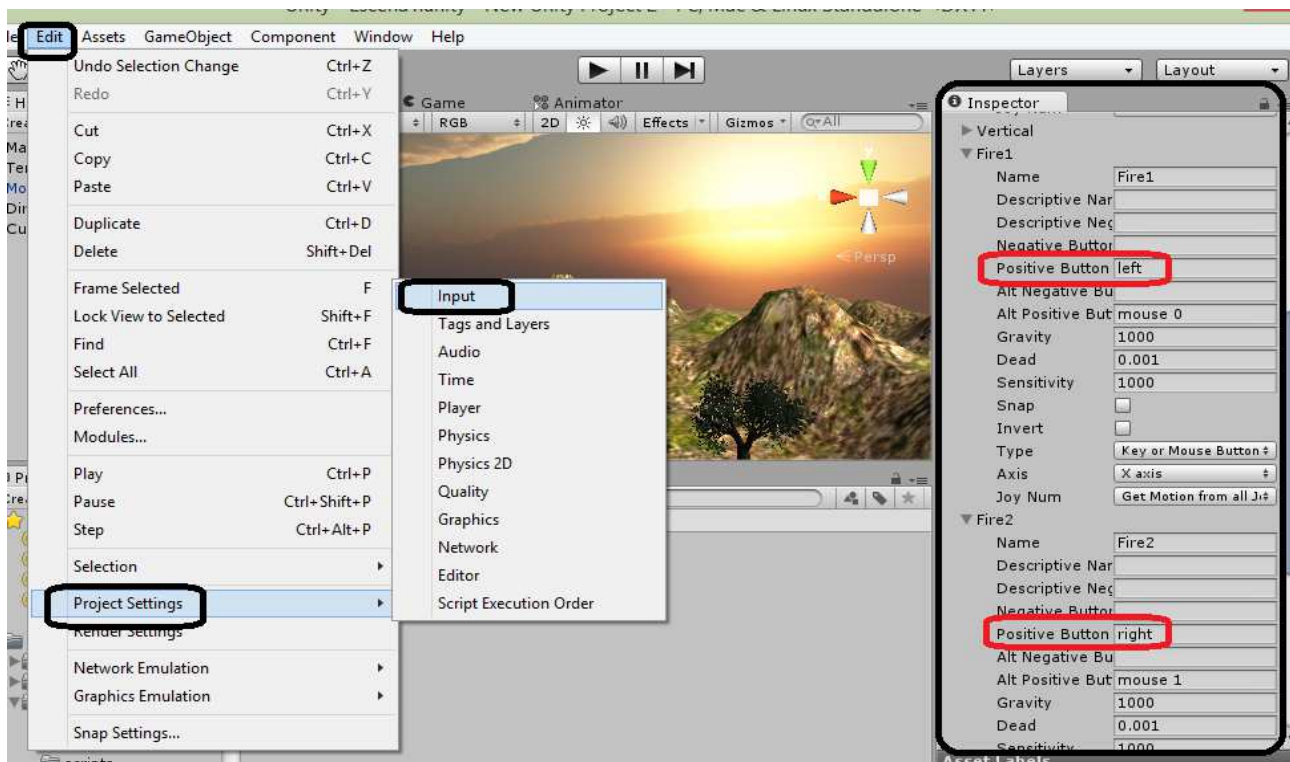
Is Kinematic: Indica si el cuerpo es manipulado o no por el Physic Engine.

Interpolate: la interpolación se utiliza para suavizar el movimiento de un objeto.

Collision Detection: Determina la forma en la cual se calcula la colisión.



Ahora ya podemos interactuar con el cubo y asociarle movimientos que respondan por ejemplo al teclado. Vamos a: Edit > Project Settings > Input y ajustamos en el panel de Inspector Position Button de Fire1 a left y Position Button de Fire2 a right.



Si creáramos y asociáramos el siguiente script al cubo, podríamos alterar su posición según una determinada velocidad de translación en x, y ,z:

```
#pragma strict

function FixedUpdate () {
    if (Input.GetButtonDown ("Fire1")) {
        rigidbody.velocity = Vector3(-3,0,0);
    }
    if (Input.GetButtonDown ("Fire2")) {
        rigidbody.velocity = Vector3(3,0,0);
    }
    if (Input.GetButtonDown ("Jump")) {
        rigidbody.velocity = Vector3(0,3,0);
    }
}
```

Podríamos asociar una tecla también a una velocidad de rotación según un determinado eje al emplear “angularVelocity” en lugar de “velocity” o utilizar una fuerza de frenado lineal o angular al emplear “drag” o “angularDrag” respectivamente.

Limitar Movimientos

En determinadas ocasiones podríamos querer limitar el movimiento de un objeto según un determinado eje, tanto en traslación o rotación para lo cual podemos utilizar atributos como:

FreezeAll
Freeze Positon
FreezePositionX
FreezeRotationZ

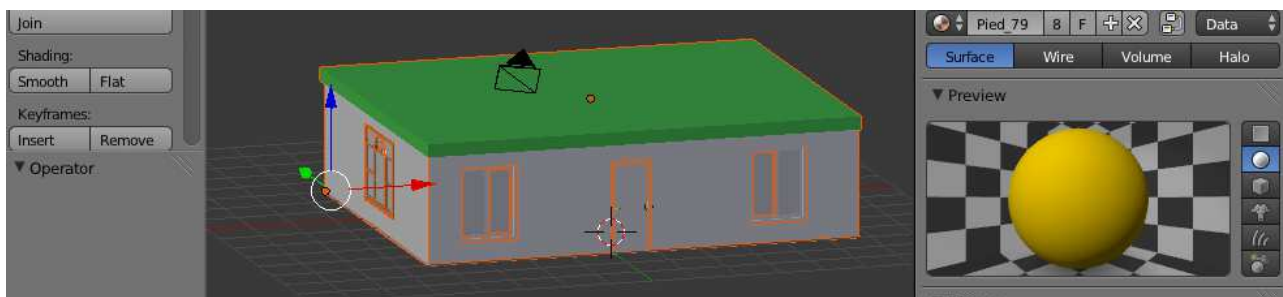
Un ejemplo en el cual limitamos el movimiento según el eje y podría ser como sigue:

```
function FixedUpdate () {
    if(Input.GetButtonDown("Fire1")){
        rigidbody.velocity=Vector3(3,0,0);
        rigidbody.constraints = RigidbodyConstraints.FreezePositionY;
    }
}
```

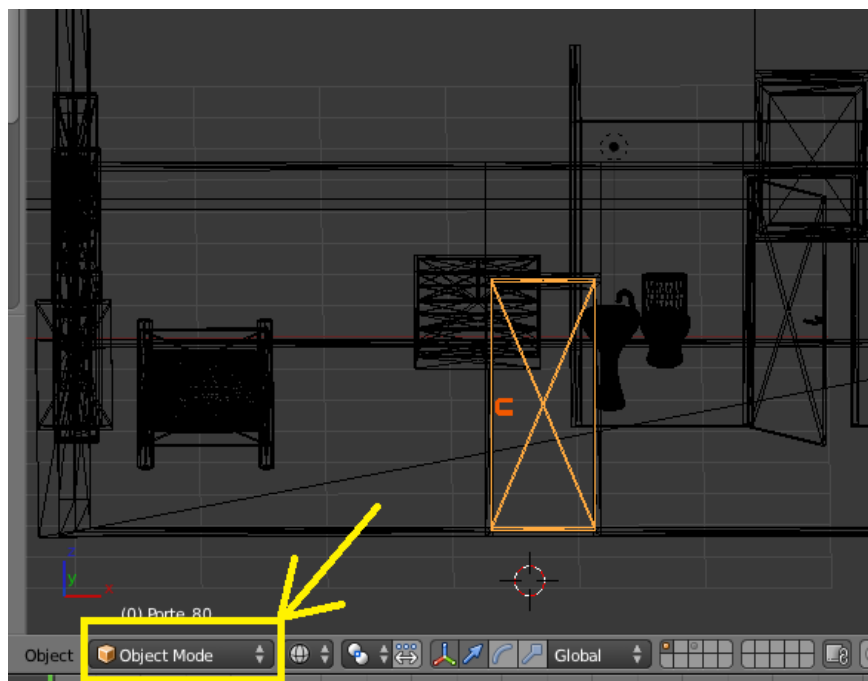
Bloque temático 2: Incorporar movimiento a puerta - Blender

Vamos a trabajar para agregar interacción a la puerta de la casa creada con Sweet home, para eso vamos a seguir un serie de pasos según es describe a continuación.

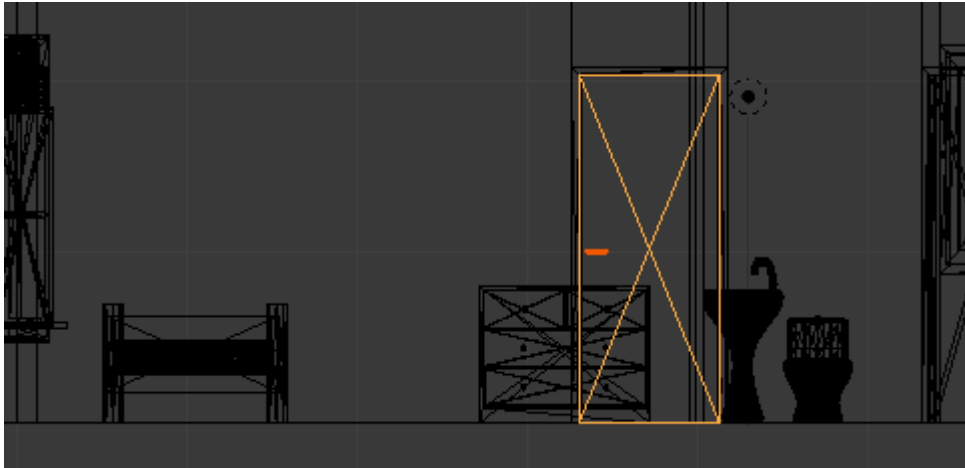
Paso 1: Abrimos el Proyecto de la casa que tenemos en Blender.



Paso 2: Seleccionamos la puerta en modo Object Mode

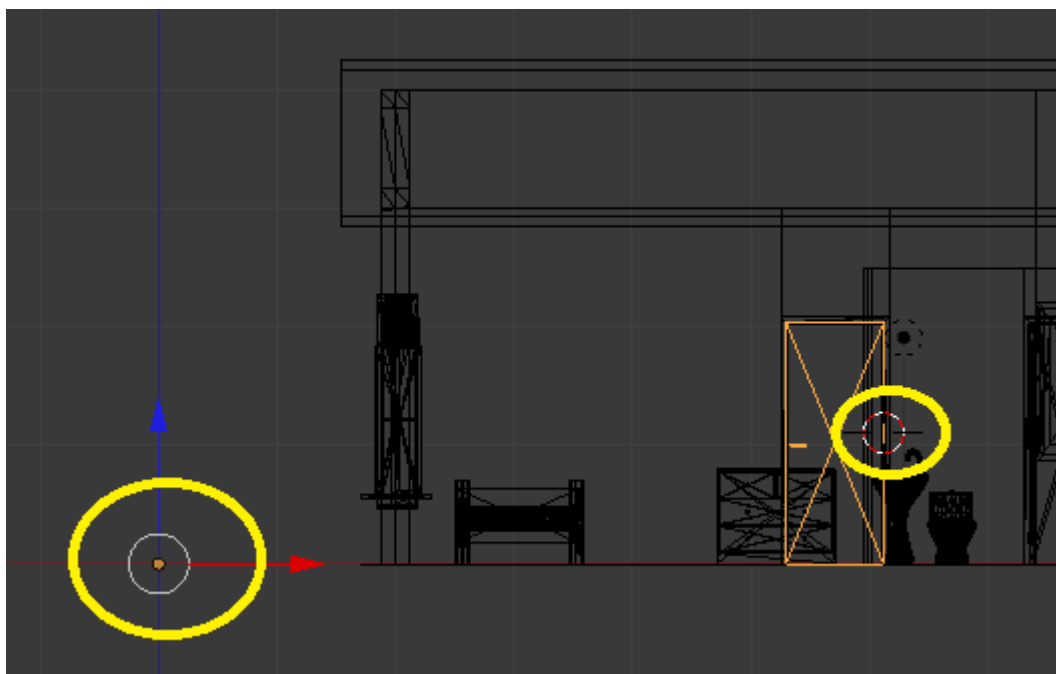


Paso 3: Seleccionamos la puerta y el pestillo y pasamos a vista frontal.



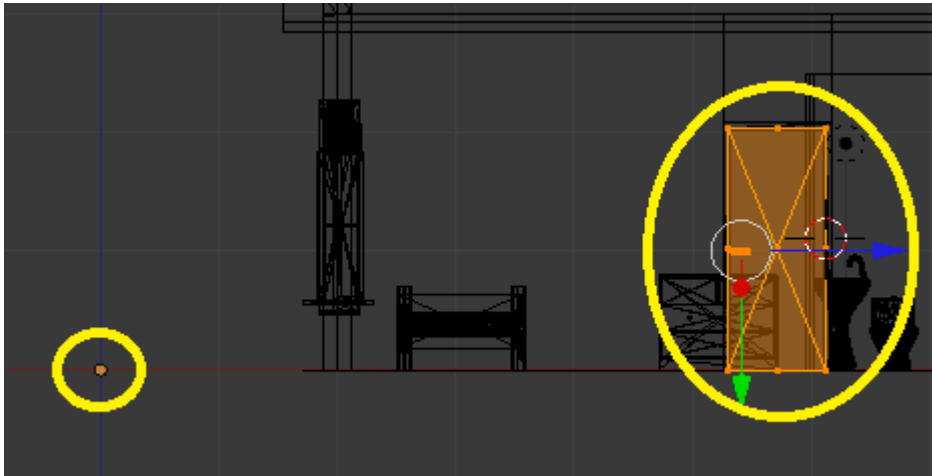
Paso 4: Presionamos CTRL + J para unirlos en un solo mayado.

Paso 5: Ahora que son un solo objeto, tenemos que hacer coincidir el punto sobre el cual se aplican las rototranslaciones, y cambios de escala con el punto en donde queremos que pivotee la puerta al abrirse, en todas las vistas

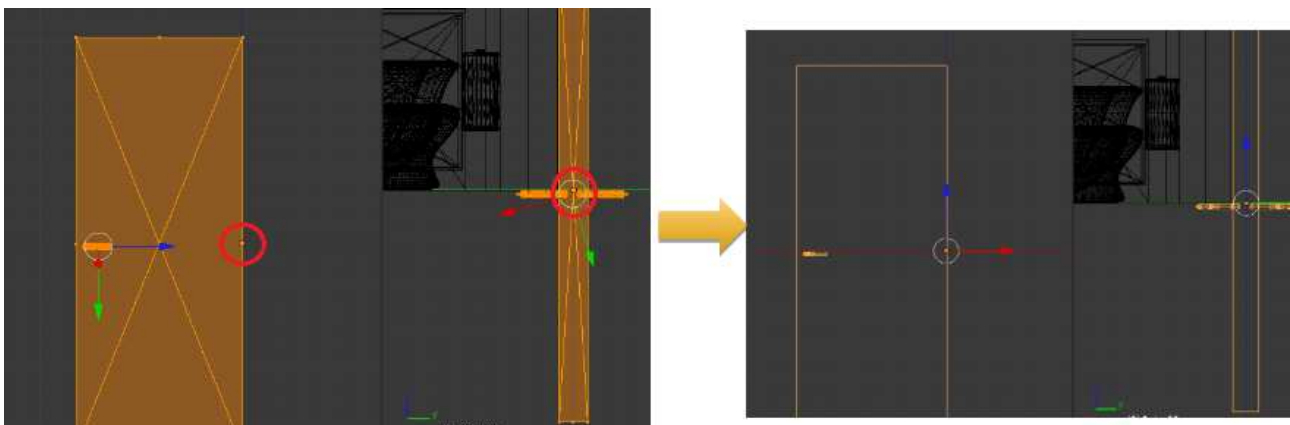


Paso 6: Entramos en Edit Mode, y vemos que ahora podemos mover la puerta, mientras que el punto sobre el cual se movía en Object Mode permanece fijo.

Hacemos que los movimientos necesarios, para estar en la condición especificada en el punto 5

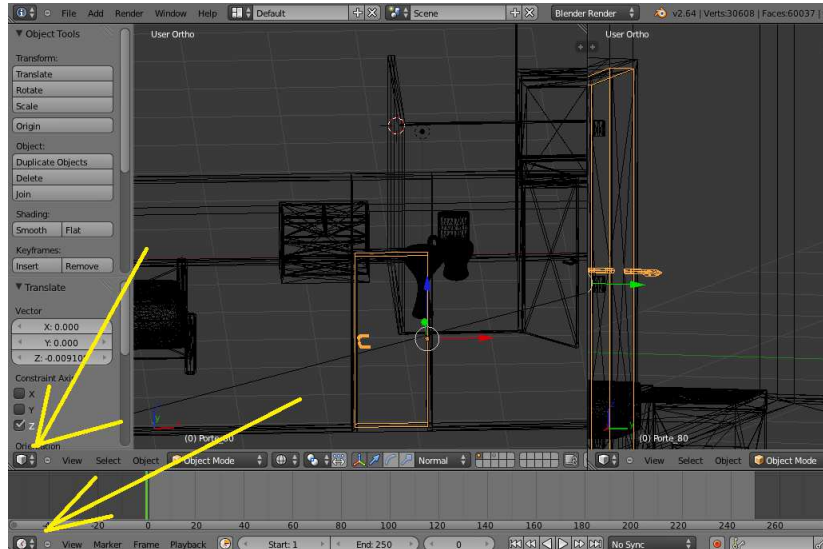


Paso 7: Nos ayudamos de las vistas Frontal y Derecha para realizar la ubicación, y luego pasamos a modo Object Mode

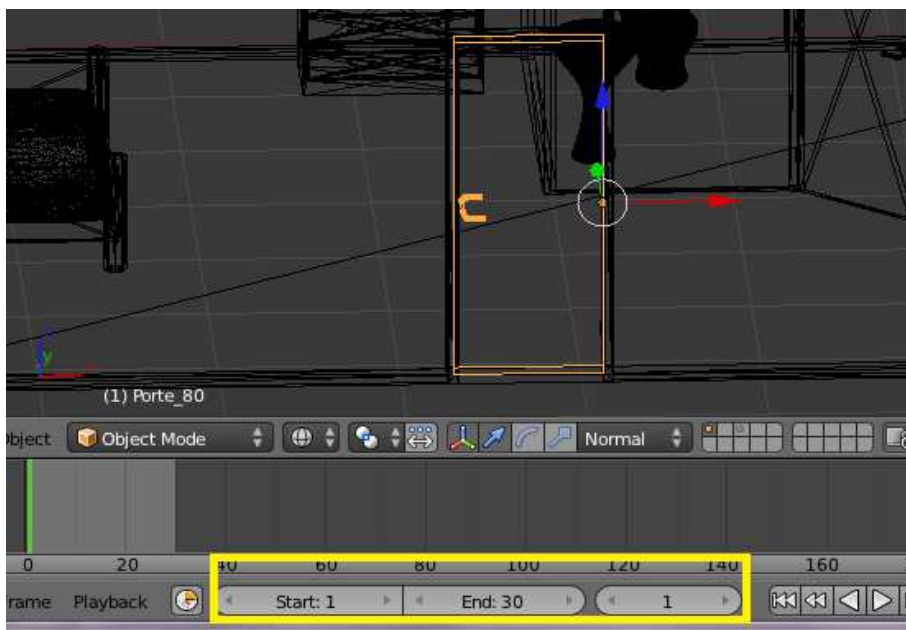


Paso 8: Retornamos la puerta a su lugar.

Paso 9: Mantenemos al menos una pantalla en modo 3D y una en modo TimeLine

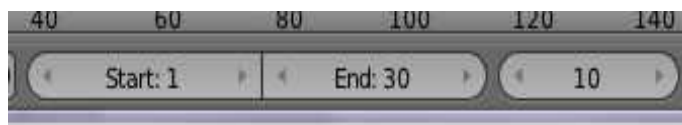


Paso 10: Establecemos la línea de tiempo entre 1 y 30 y nos paramos en el frame 1.

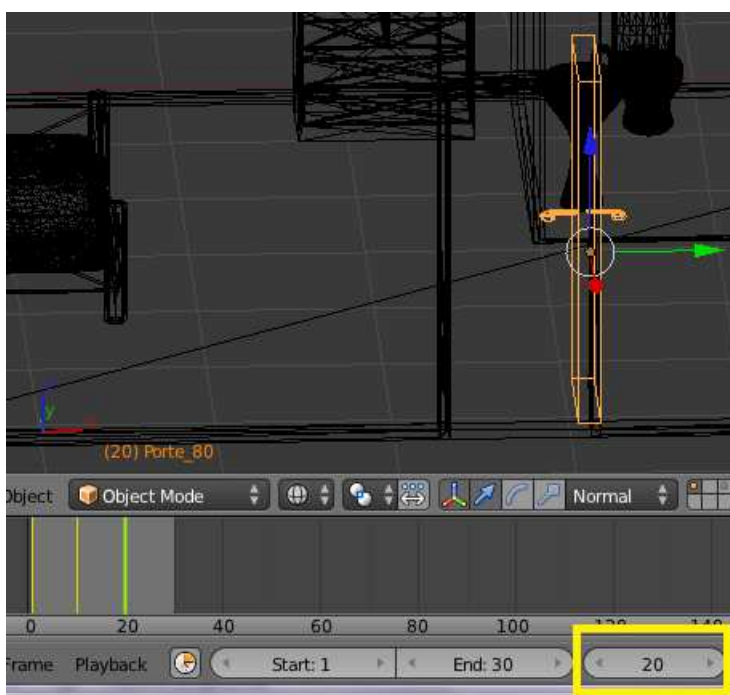


Paso 11: En la pantalla 3D presionamos la tecla “i” y seleccionamos LocRot.

Paso 12: Nos movemos al frame 10 y nuevamente sobre la pantalla 3D presionamos la tecla “i” y seleccionamos LocRot

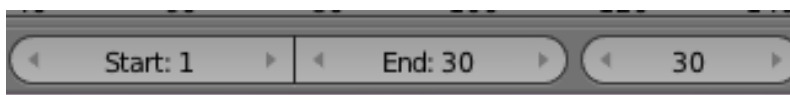


Paso 13: Nos movemos al frame 20, rotamos sobre el eje vertical (en mi caso el z) y aplicamos una rotación de 90° hacia adentro de la casa (en mi caso -90°) y nuevamente sobre la pantalla 3D presionamos la tecla “i” y seleccionamos LocRot

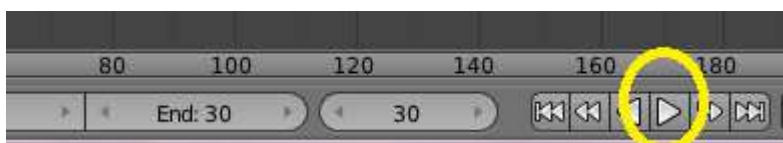


Nota: Recordar que para rotar apretamos la tecla *r*, seguida de la letra que identifica al eje sobre el cual se va a aplicar la rotación y luego se introduce el numero de grados positivo o negativo según la dirección de giro. Ej: *r > z > -90*

Paso 14: Nos movemos al frame 30, y rotamos sobre el eje vertical para retornar a la posición de puerta cerrada. Nuevamente sobre la pantalla 3D presionamos la tecla “i” y seleccionamos LocRot.



Paso 15: Si ahora presionamos el botón de play, podemos ver el movimiento completo



Pasamos a trabajar desde UNITY

Paso 16: Vamos a la carpeta “Assets” y dentro de ella creamos una carpeta en la cual vamos a colocar todos los archivos importados desde blender y las texturas. Llamémosla “casa”.

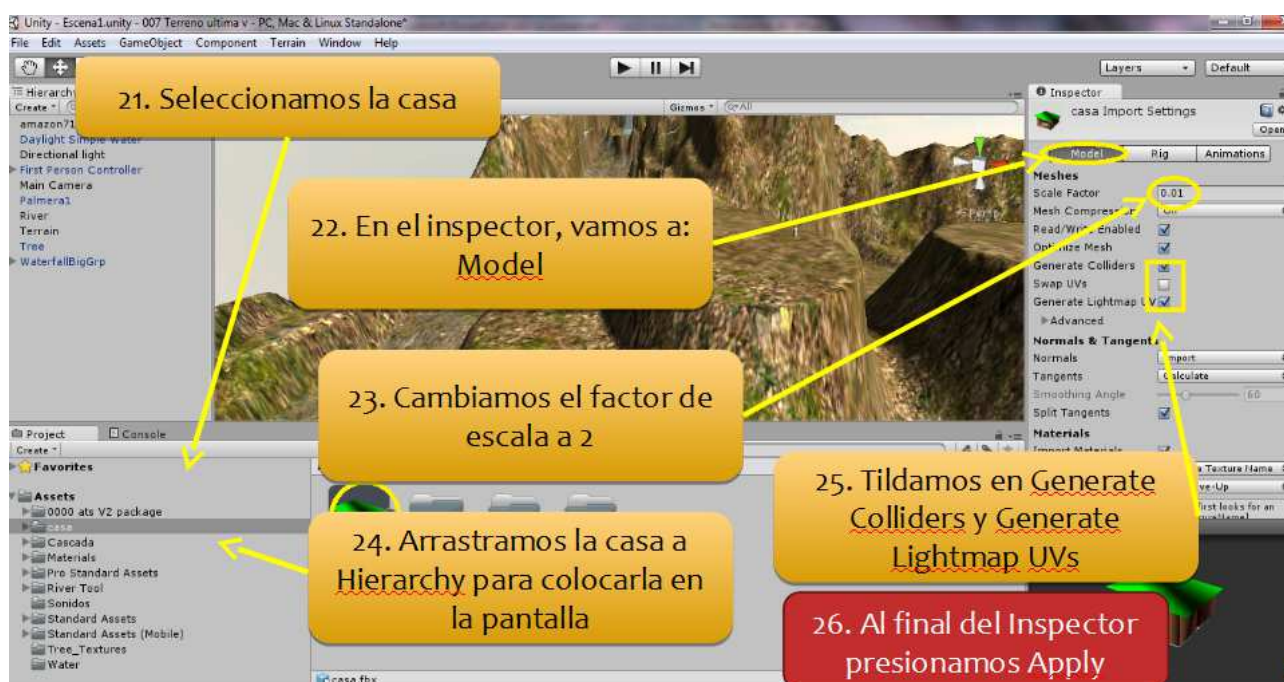
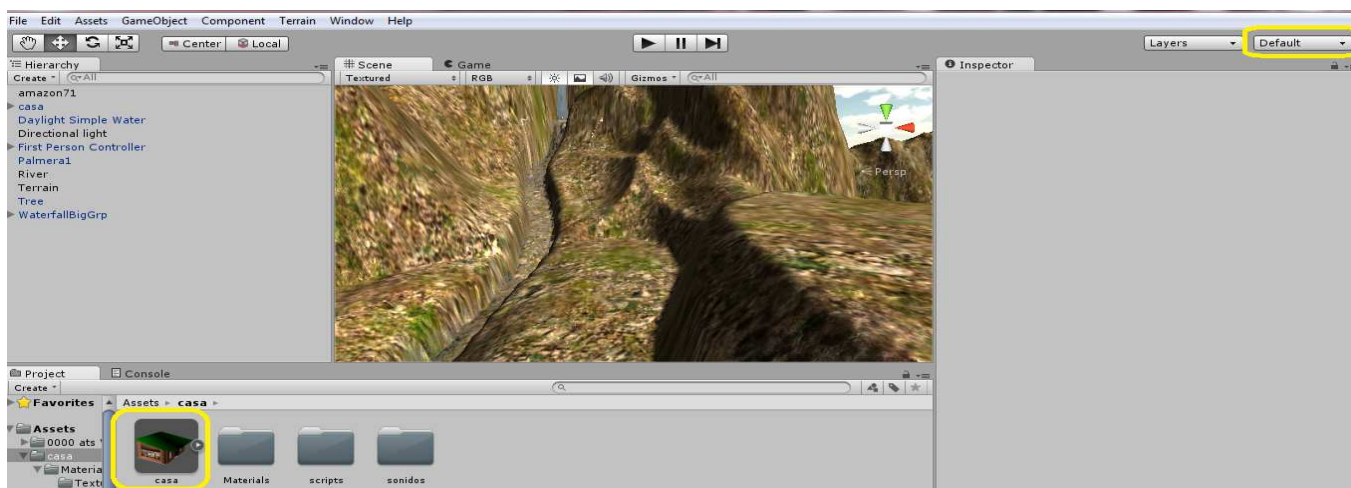
Paso 17: Eliminemos la casa que habíamos exportado anteriormente en Unity.

Paso 18: En la carpeta casa, creamos dos carpetas una llamada “scripts” en donde vamos a ubicar los scripts para el movimiento de la puerta, y otra llamada “sonidos” en donde vamos a poner los sonidos de apertura y cerrado de puerta

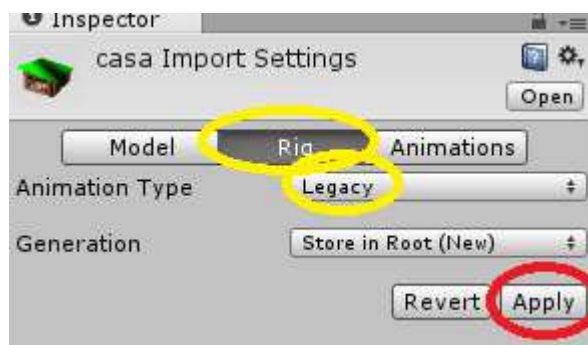
Nota: Los archivos van adjuntos al Power Point

Paso 19: Guardamos la casa creada en blender dentro de la carpeta casa, en formato Blender como: casa.blend

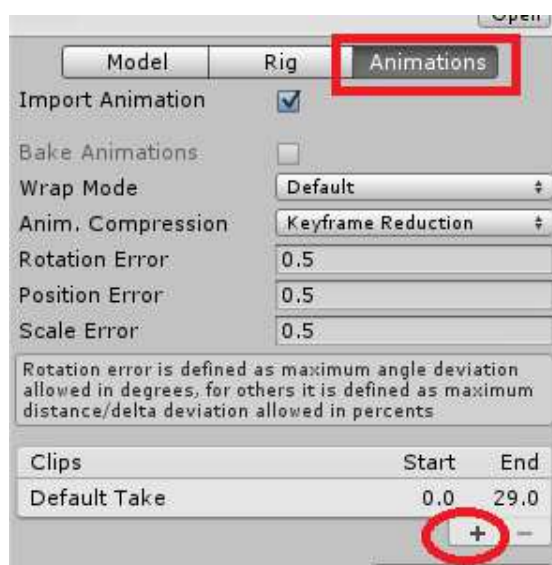
Paso 20: Ahora en Unity elijamos la vista Default, que nos va a permitir trabajar mejor con animaciones



Paso 27: Pasamos a Rig y ponemos el Tipo de Animación en “Legacy” Luego damos Apply

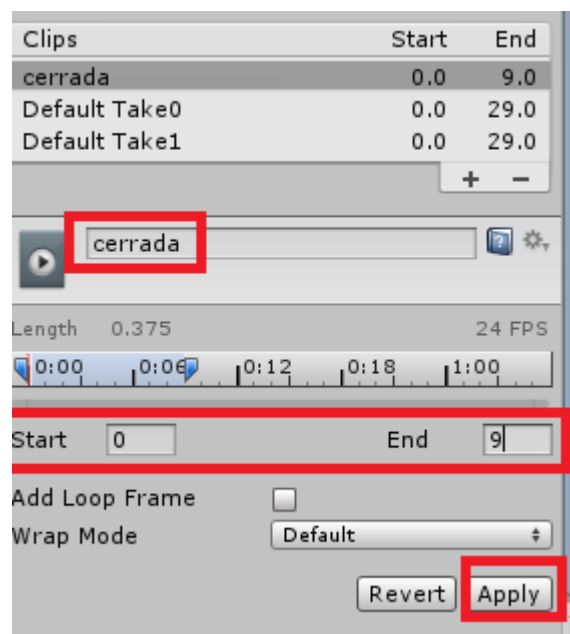


Paso 28: Vamos a Animations, y vemos que nuestra animación que iba de 1 a 30 frames, a pasado a de 0 a 29 frames. Damos dos veces en el signo de + para agregar dos animaciones.



Clips	Start	End
Default Take	0.0	29.0
Default Take0	0.0	29.0
Default Take1	0.0	29.0

Paso 29: Seleccionamos la primer animación, la renombramos como “cerrada” y ponemos los frames entre 0 y 9. Luego damos Apply



Paso 30: Repetimos el paso 29 con la siguiente animación, llamándola “abrir” y ponemos los frames entre 10 y 19. Luego damos Apply.

Paso 31: Repetimos el paso 29 con la siguiente animación, llamándola “cerrar” y ponemos los frames entre 20 y 29. Luego damos Apply.

Nos debe quedar algo como esto

Clips	Start	End
cerrada	0.0	9.0
abrir	10.0	19.0
cerrar	20.0	29.0

Paso 32: Al final del Inspector presionamos Apply

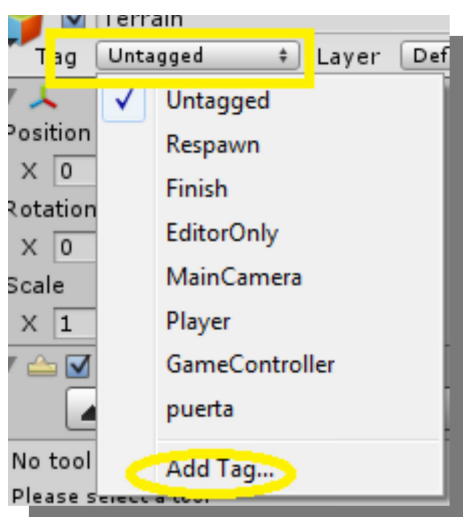
Paso 33: Copiamos el script adjunto, y los sonidos en las carpetas creadas en el punto 18

Paso 34: Arrastramos el script a nuestro personaje que es el que vamos a hacer colisionar con la puerta para que se abra. (Nuestro personaje debe poseer un Character Controller)

Paso 35: Seleccionamos el elemento puerta de la casa, y vamos a Tag en el inspecto.



Paso 36: Presionamos en el desplegable, y luego en Add Tag... Luego debajo del número 1 agregamos el valor “puerta”



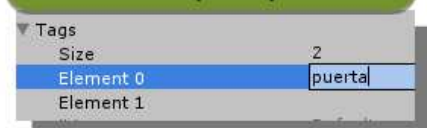
36.1. Presionamos sobre el triangulito junto a la palabra Tags



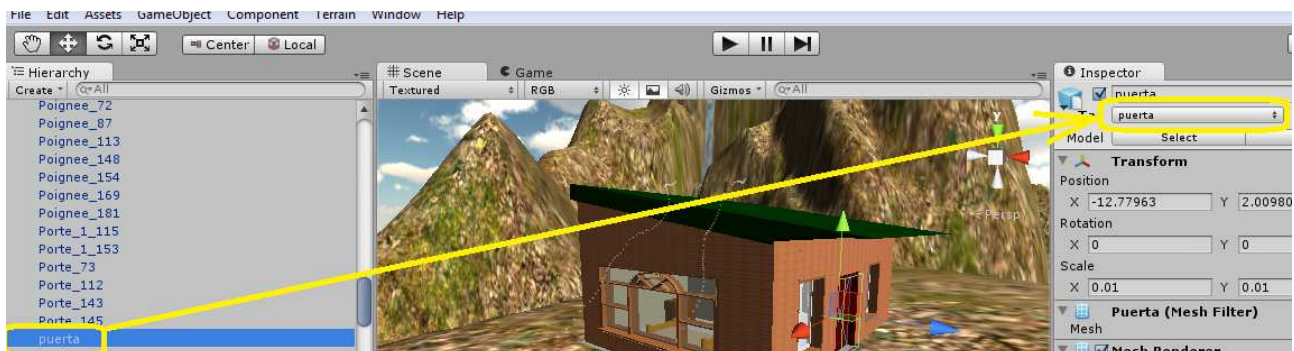
36.2. Presionamos debajo del 1 en la línea de Elemento 0



36.3. Escribimos la palabra puerta en el casillero que aparece

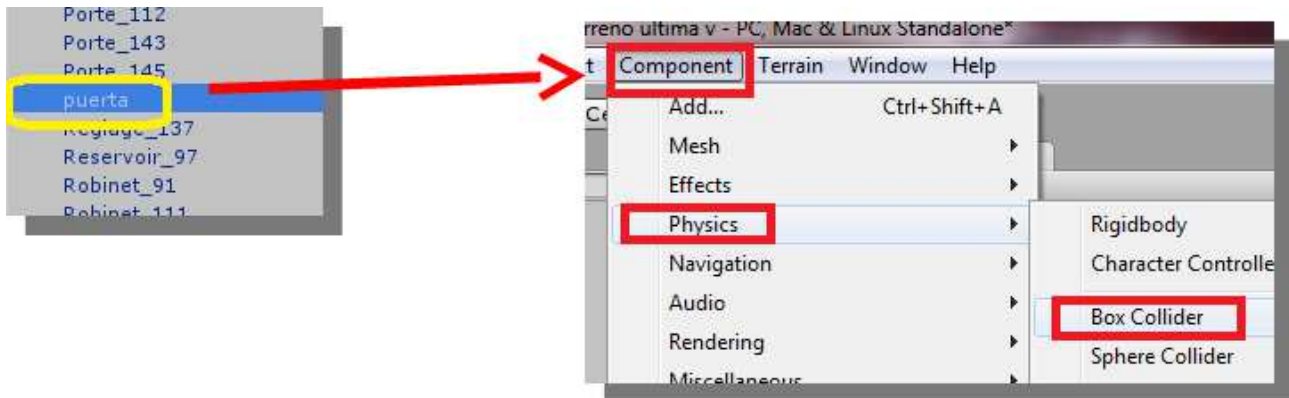


Paso 37: Seleccionamos nuestra casa en Hierarchy, y en Tag, seleccionamos el recién creado “puerta”

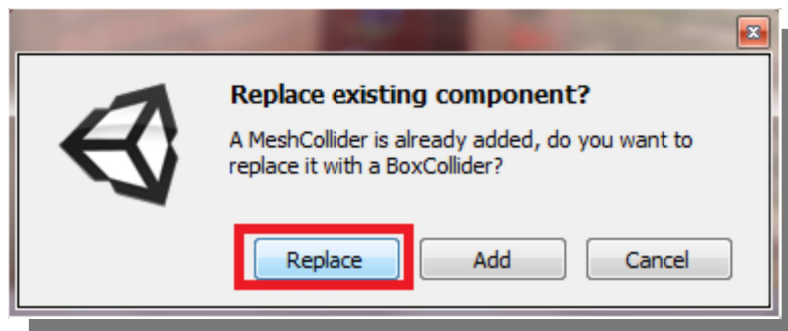


Paso 38: Seleccionamos la puerta en el panel Hierarchy y vamos a:

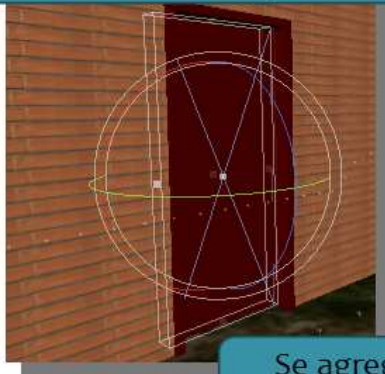
Component > Physics > Box Collider



Paso 39: Presionamos en “Replace” en el cartel desplegable que nos aparece.



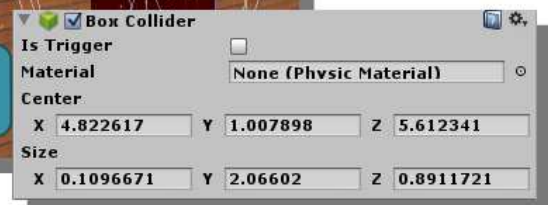
Podemos ver el Box Collider colocado en la puerta



39. Modificando los valores en el inspector, de "Extents" podemos aumentar la zona en la dirección saliente a la puerta, en la cual la puerta va a reaccionar con la cercanía del jugador y se va a abrir



Se agrega una solapa en el inspector



40. Presionamos sobre nuestro personaje, y arrastramos los archivos de sonido según se indica en la siguiente figura.



Ya podemos entrar a modo juego y ver el resultado

Bloque temático 3: Sonido

El sonido es una parte fundamental de un videojuego, así como lo es de una película o de una obra de teatro, en realidad es una parte fundamental de nuestras vidas, la música nos inspira y despierta emociones. El mundo cinematográfico lo comprendió hace mucho tiempo, prácticamente desde sus inicios, desde las fantasías animadas de los dibujos de Walt Disney, pasando por películas que han quedado en el recuerdo de todos por las emociones que representan, como la música de tiburón, ojo de tigre de la película Rocky, etc, etc,...

La música es también parte fundamental de un videojuego, y en esta área existen grupos musicales y artistas que se dedican especialmente a agregarle el sonido adecuado a un juego para lograr persistir su recuerdo en la mente de las personas.

La plataforma de Unity nos permite agregar sonido en los siguientes formatos:

- 1.- ogg
- 2.- wav
- 3.- mp3

Existen muchos sitios donde poder descargar sonidos. En el siguiente link pueden encontrar algunas páginas interesantes:

<http://www.genbeta.com/multimedia/10-sitios-para-descargar-gratis-efectos-de-sonido>

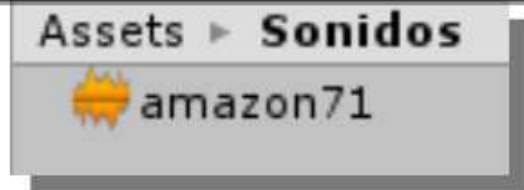
pueden servir para animar puertas, disparos, rayos, sonido ambiente, etc. sin embargo es muy probable que con el paso del tiempo y a medida que queramos darle mayor terminación a los efectos, necesitemos crear nuestros propios sonidos, y por que no contar con la ayuda de un profesional en esta área.

Veamos a continuación, como podemos agregar sonido en Unity:

1. Agregamos dentro de la carpeta Assets una carpeta llamada Sonido

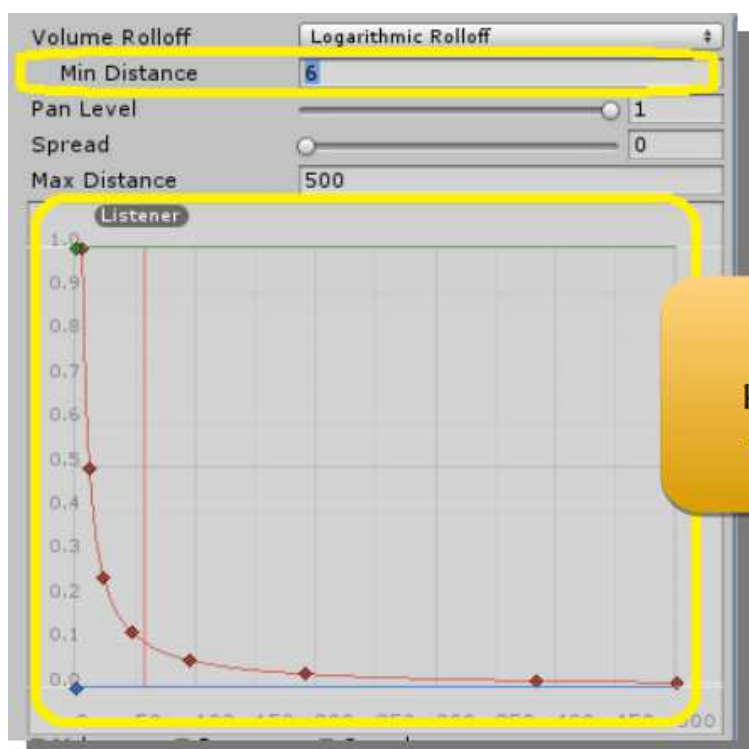
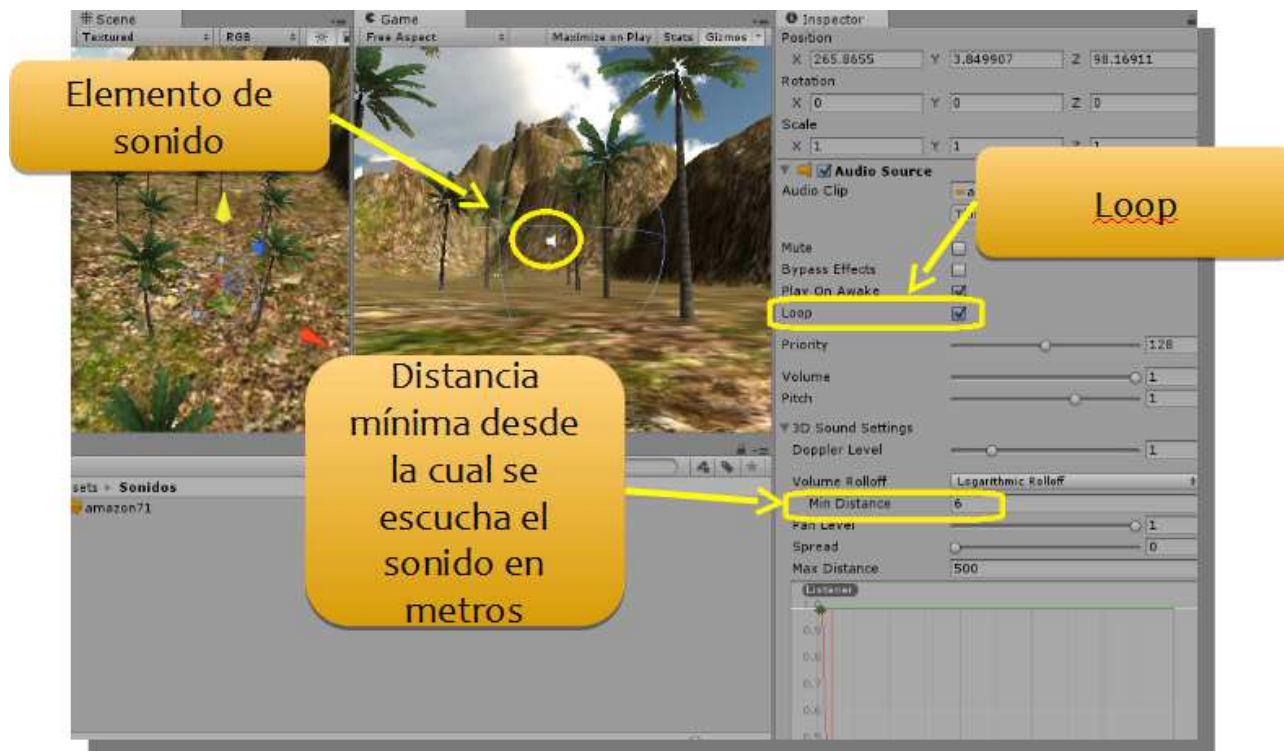


2. Dentro de la carpeta agregamos nuestro sonido



3. Arrastramos el sonido hasta la carpeta “Hierarchy”

4. Seleccionamos el sonido agregado y lo desplazamos en la escena hasta el punto en donde se originaría



Curva de atenuación editable.
Probemos modificarla, variando el valor de la distancia mínima.



Bibliografía utilizada y sugerida

Documentación oficial online -

<http://docs.unity3d.com/ScriptReference/Rigidbody.html>

<http://unity3d.com/es/unity/quality/audio>

Lo que vimos

En esta unidad hemos visto el concepto de RigidBody, como interaccionar con objetos y como trabajar con sonidos.



.....