



Componente formativo

## **Programación de eventos de personajes y entornos**

---

### **Breve descripción:**

El componente formativo se enfoca en el diseño de interfaces de usuario y eventos para facilitar la interacción de las diferentes acciones que ocurrirán en los niveles de juego.

### **Área ocupacional:**

Ciencias Naturales.

---

**Mayo 2023**

## Tabla de contenido

1. Introducción .....	3
1. Programación de comportamientos de personajes controlados por el usuario.....	3
1.1. Configuración de animaciones de los personajes.....	7
1.2. Máquina de estado finito.....	10
2. Programación de eventos de interfaz HUD.....	14
2.1. Componentes Canvas .....	16
2.2. Interfaces de usuario responsivas .....	17
2.3. Eventos.....	18
3. Programación de eventos disparadores (interacción con el entorno) .....	20
4. Programación de comportamientos relativos a la mecánica de videojuego.....	26
5. Programación de inteligencia artificial.....	27
Síntesis .....	29
Material complementario .....	30
Glosario.....	31
Referencias bibliográficas .....	32
Créditos .....	33

## 1. Introducción

Los contenidos desarrollados en este componente formativo están enfocados en el diseño de interfaces de usuario y la programación de eventos y personajes para facilitar la interacción de las diferentes acciones que se dan en un videojuego.

Se invita a ver el video a continuación para conocer un poco más sobre los contenidos a desarrollar en este componente:

### 1. Programación de comportamientos de personajes controlados por el usuario

La programación de los comportamientos de los personajes permitirá definir las diferentes acciones que serán ejecutadas por el personaje cuando el jugador interactúe con los diferentes dispositivos de juego (joystick, teclado, mouse).

#### Input manager

El input manager es una ventana especializada en el motor de desarrollo Unity3D que permite definir los diferentes ejes (axes) que se encuentren asociados a las diferentes acciones del juego y/o personaje.

El input manager en Unity3D utiliza los siguientes tipos de controles:

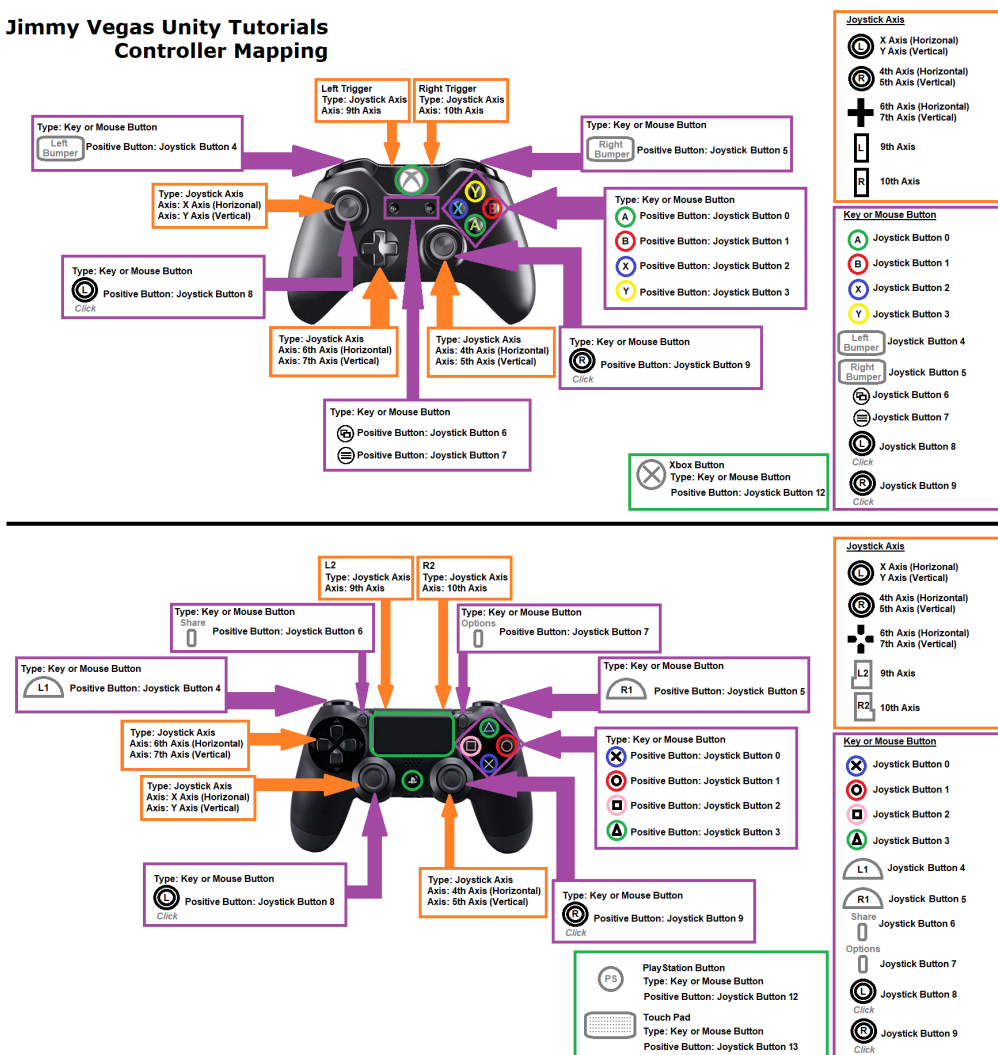
<b>Key</b>	Hace referencia a cualquier tecla (key) que se encuentre en teclado físico, como por ejemplo X, Ctrl, Shift.
------------	--

Button	Hace referencia a cualquier tipo de botón que se encuentre en un control físico como un gamepad, joystick.
Virtual Axis	Permite realizar el mapeo de los ejes de un control físico. Por ejemplo, cuando el jugador mueve las palancas de un gamepad.

En la siguiente figura se puede observar el mapeo de controles de Unity:

**Figura 1.** Mapeados controles de juego

## Jimmy Vegas Unity Tutorials Controller Mapping

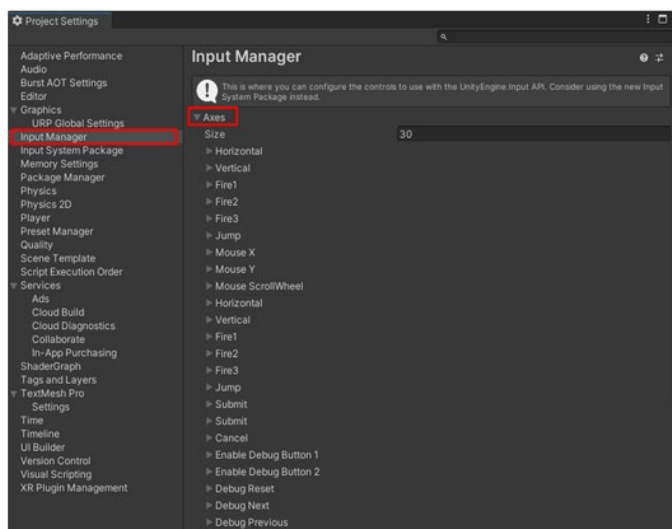
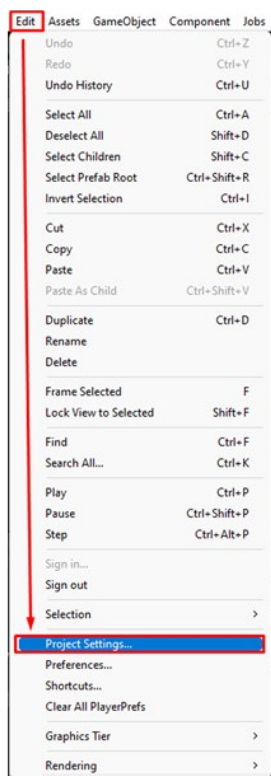


Nota. Tomada de Vegas (s. f.).

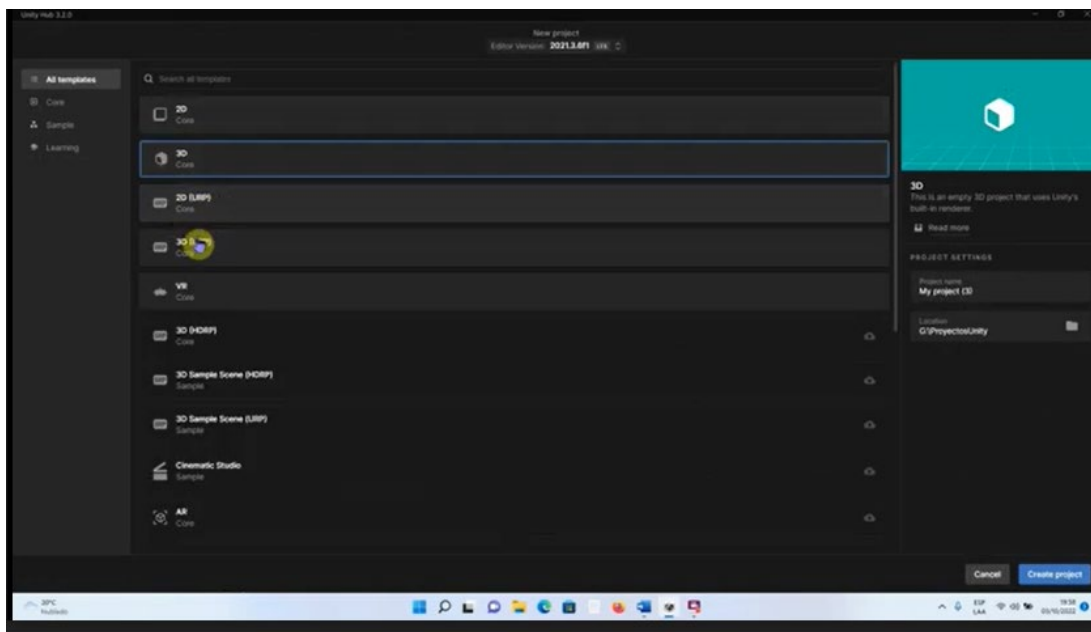
<http://www.jvunity.weebly.com/uploads/4/7/6/0/47606749/controlmap.png>

Para acceder a la ventana Input Manager dar clic en el menú Edit > Project Settings como se muestra en la siguiente imagen:

**Figura 2.** Input Manager Unity3D



En el siguiente video tutorial se observa la programación de personajes controlados por el usuario y la creación de un nuevo proyecto.

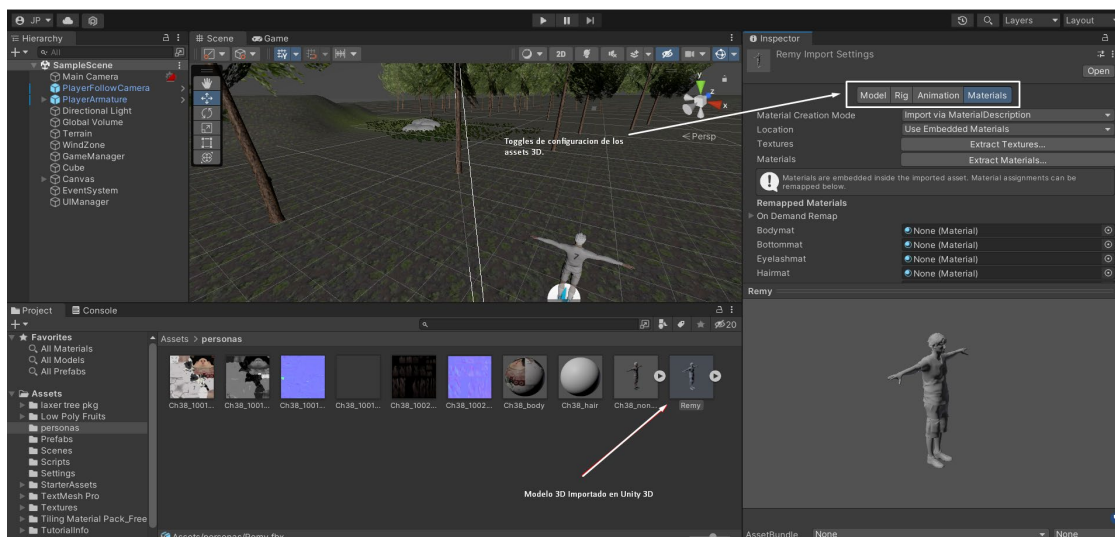


Input Manager

## 1.1. Configuración de animaciones de los personajes

El motor de desarrollo Unity 3D posee diferentes herramientas en su editor que permiten manipular las animaciones de los personajes importados. Las animaciones de los personajes pueden encontrarse embebidas en el personaje o externas al personaje.

**Figura 3.** Editor Unity 3D



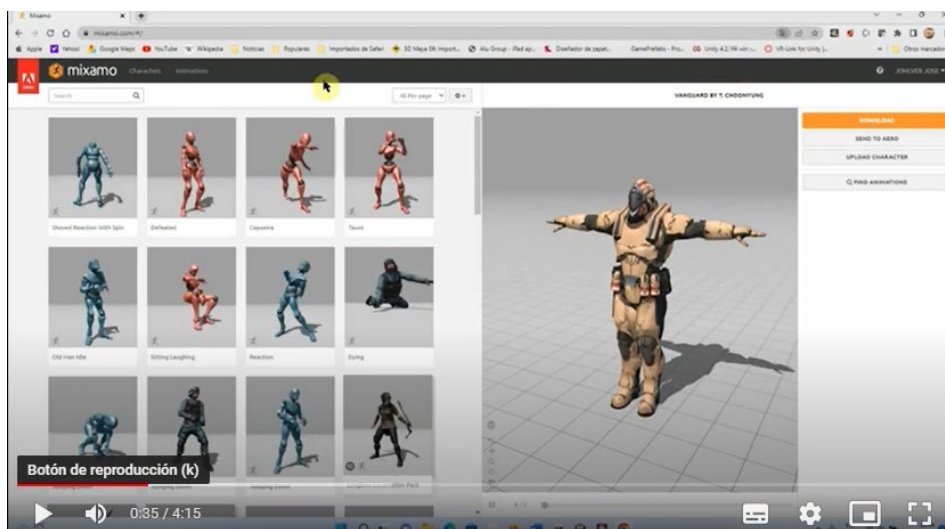
En la web existen un gran número de recursos que facilitan los procesos de animación de personajes si no se tienen las habilidades o no se tienen las animaciones para llevar a cabo este proceso. Para este componente se va a trabajar con la página Mixamo ([www.mixamo.com](http://www.mixamo.com)), que permite realizar las animaciones de personajes de un juego.

Para acceder a la página de Mixamo y comenzar a hacer las animaciones se deben seguir estos pasos.

En los siguientes videos tutoriales se explica cómo se pueden generar personajes y darles diferentes animaciones.

**Figura 4.** Obtener modelo 3D en Mixamo





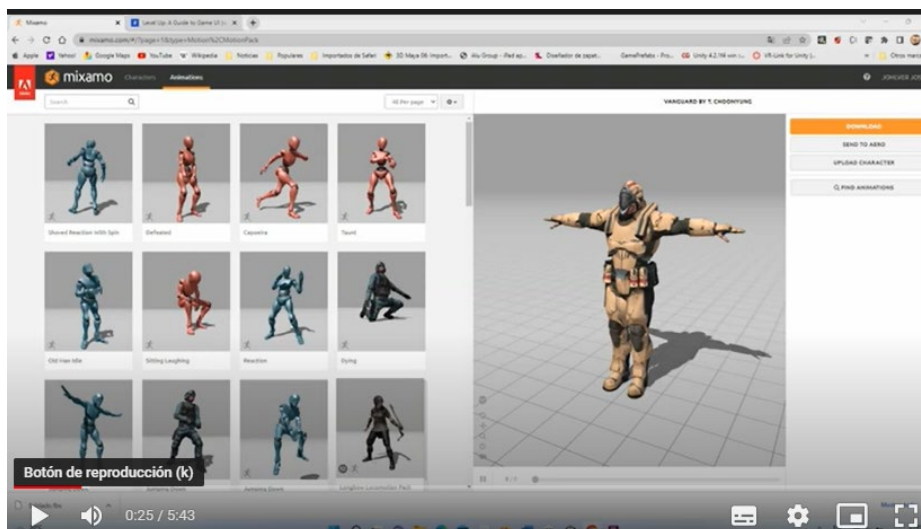
Obtener modelo 3D en Mixamo

**Figura 5.** Importar personaje



Importar personaje

**Figura 6.** Asignar animaciones



Asignar animaciones

## 1.2. Máquina de estado finito

### (Animator Controller)

El Animator Controller es un panel del editor de Unity3D que permite organizar y mantener un conjunto de clips de animación y asociar estas animaciones por medio de transiciones. En la gran mayoría de los casos, el Animator Controller puede contener varias animaciones y cambiar entre cada una de ellas dependiendo de las condiciones que ocurran en el juego, como se puede observar en el siguiente video.

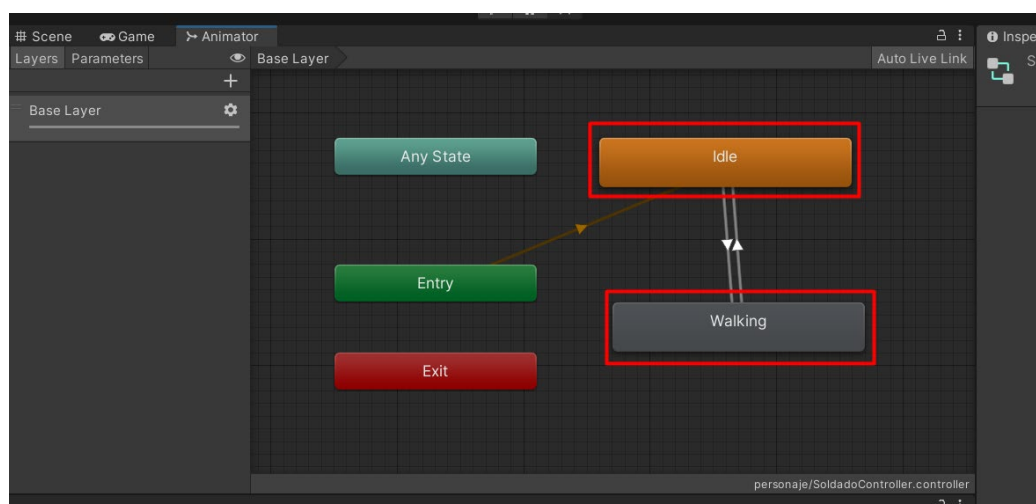


## Animator Controller

### Estados de animación

Los estados en la máquina de estados finito (Animator Controller) son la representación de los comportamientos que el personaje va a tener (reposo, caminar, correr y saltar).

**Figura 7.** Estados de la FMS



## Parámetros

Los parámetros en la máquina de estados finito representan una variable que permite controlar el cambio de un estado a otro haciendo uso de una transición. Los parámetros cuentan con cuatro tipos:

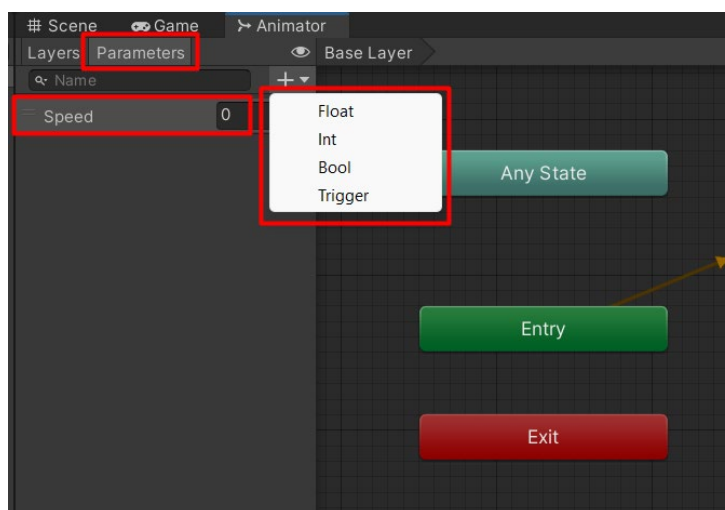
Float.

Int.

Bool.

Trigger.

**Figura 8. Parámetros FMS**

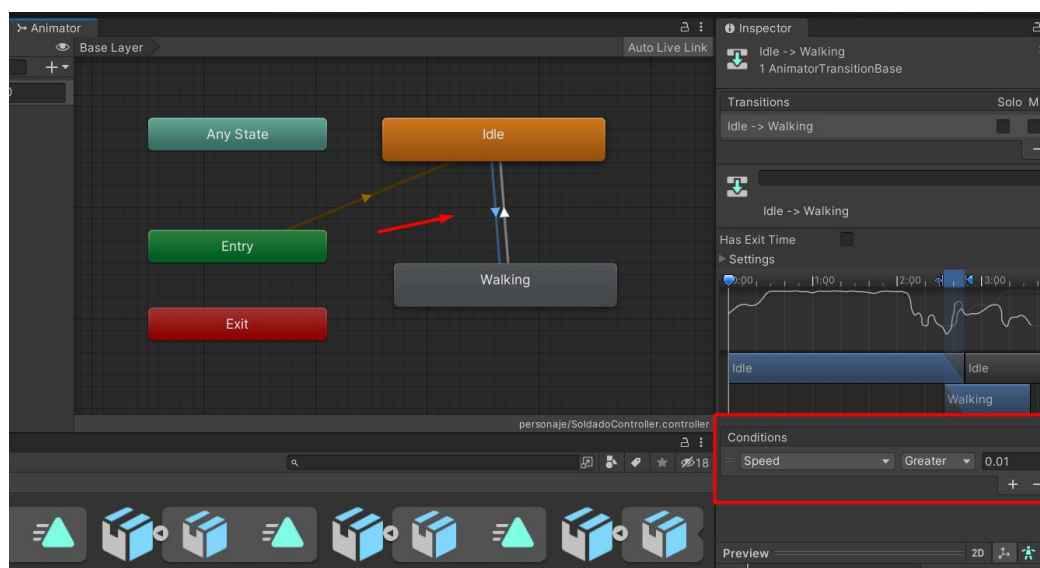


## Transiciones

Las State Machine Transitions existen para simplificar el proceso de transición entre los diferentes estados del Animator Controller. Las transiciones permiten tener un mayor nivel

de complejidad sobre la lógica de la máquina de estados; y las transiciones poseen condiciones que permiten controlar el proceso de cambio de un estado a otro.

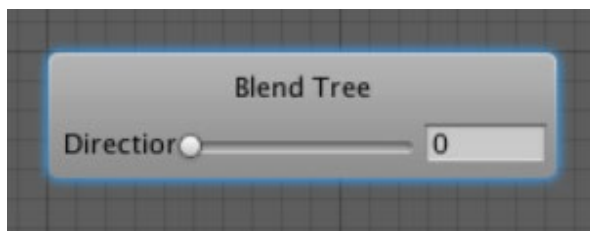
**Figura 9.** Transiciones FMS



## Árboles de mezclado (Blend Tree)

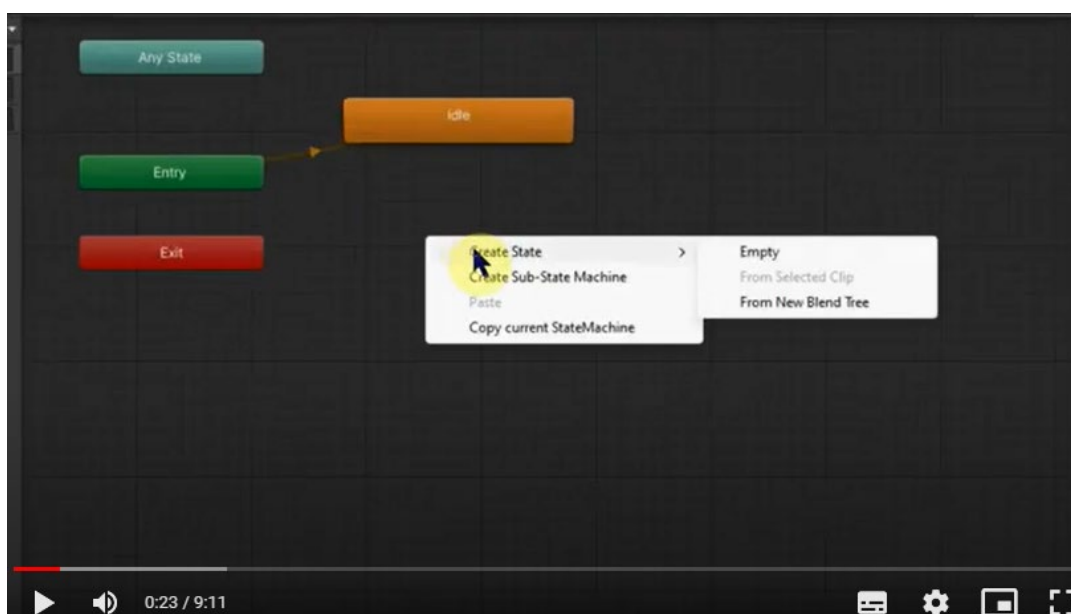
El Blend Tree permite realizar combinaciones entre los diferentes clips de animaciones y así generar comportamientos complejos.

**Figura 10.** Blend Tree Unity



blendTree.png

En el video que sigue a continuación se observa cómo se trabaja con el árbol de mezclado o Blend Tree.



Blend tree

## 2. Programación de eventos de interfaz HUD

La interfaz de usuario es uno de los principales elementos de los niveles de juego ya que por medio de estos, los usuarios pueden monitorear el comportamiento del nivel de

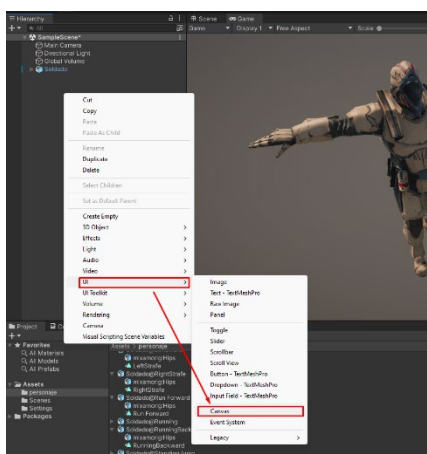
juegos. Por ejemplo, pueden conocer cuánto tiempo les queda de juego, cuánto nivel de salud les queda, etc.

## Canvas

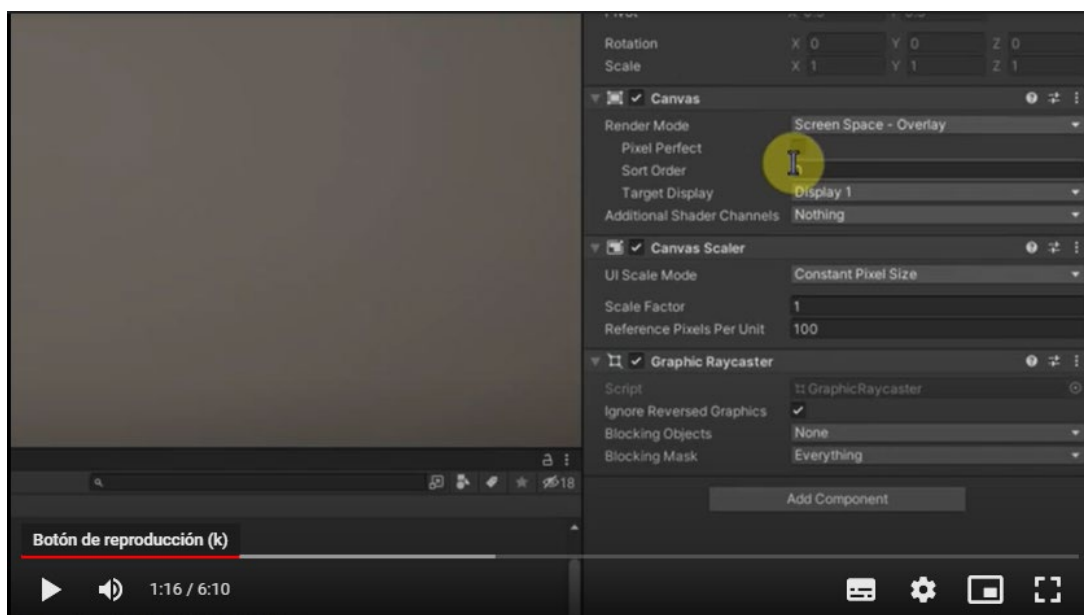
Unity3D en su conjunto de GameObject posee uno llamado Canvas que es un lienzo que permite organizar de forma detallada la interfaz de usuario. El Canvas trabaja con un componente adicional llamado el EventSystem el cual permite detectar los eventos que se ejecutan cuando el usuario interactúa con los elementos de la UI.

Para agregar el Canvas a la escena hacer clic derecho sobre el panel de jerarquía y seleccionar la opción UI>Canvas.

**Figura 11.**Canvas



Para manipular y crear interfaces de usuario con Canvas, se invita a ver este video:

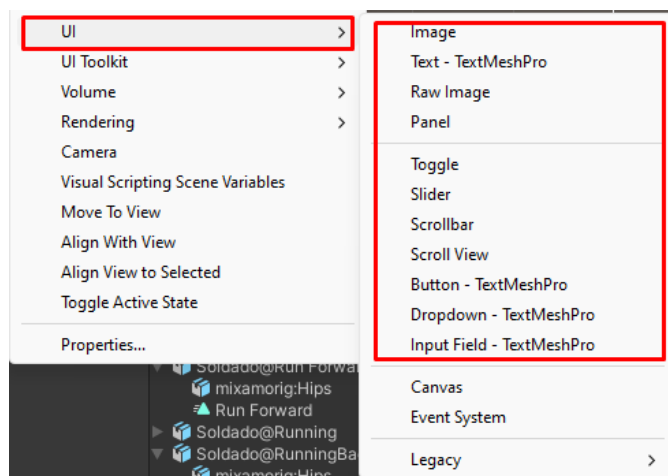


Iniciando Canvas

## 2.1. Componentes Canvas

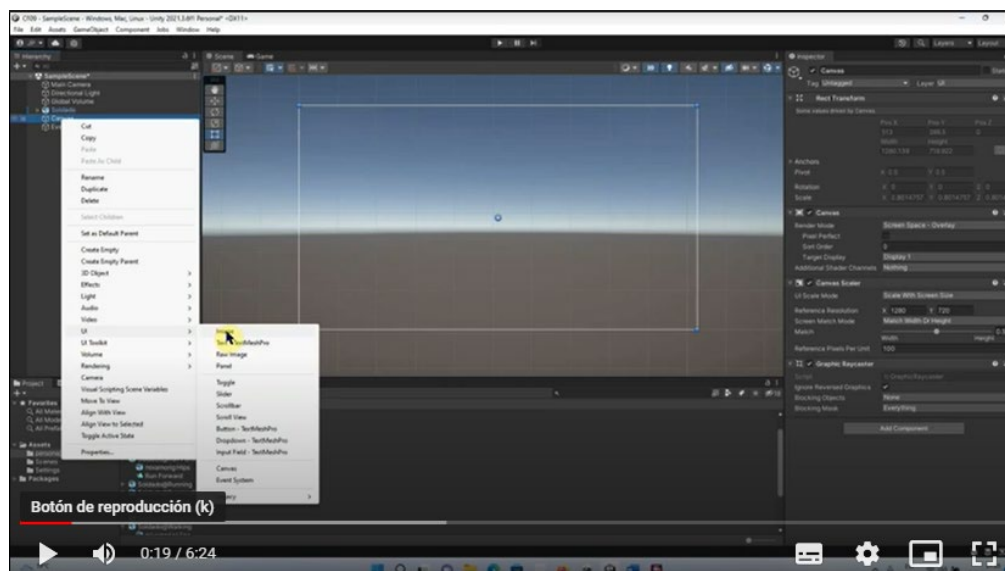
Los componentes del Canvas son los elementos usados para realizar el diseño de la interfaz de usuario. Los principales componentes de la clase UI son los siguientes:

**Figura 12.**Componentes Canvas





En el siguiente video se analizan cada uno de los componentes del Canvas.

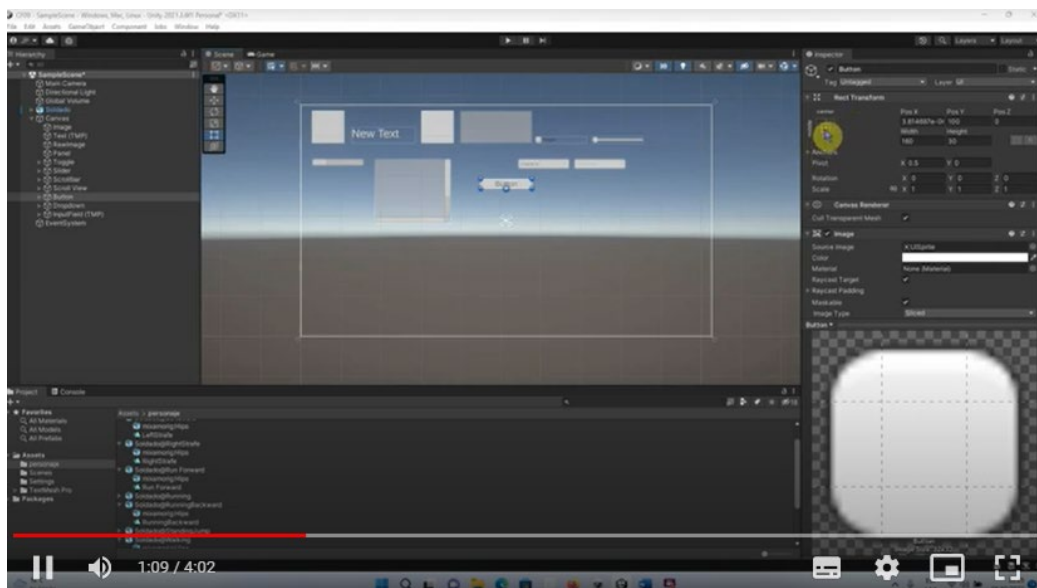


Componentes Canvas

## 2.2. Interfaces de usuario responsivas

Las interfaces de usuario responsivas son aquellas que se ajustan de forma automática a cualquier resolución de pantalla. Unity3D tiene una serie de elementos que permiten realizar este tipo de configuraciones en las interfaces de usuario, a saber:

En este video se observa cómo funciona el componente Rec transformer para manipular cada uno de los elementos del Canvas.



Rec transformer en Canvas

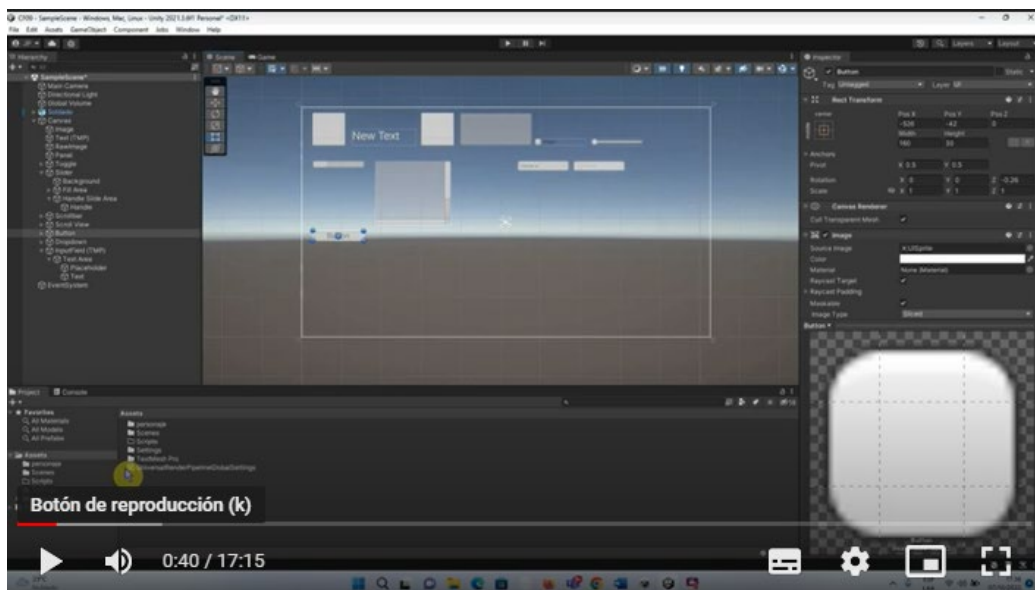
## 2.3. Eventos

Los eventos en la programación son acciones que se ejecutan cuando los usuarios interactúan con los diferentes elementos de la interfaz de usuario. En la siguiente tabla se describe cada uno de esos elementos.

Evento	Elemento UI	Descripción
OnClic	Button	Permite ejecutar una acción cuando el usuario interactúa con algún botón de la UI.

OnValueChange	DropDown	Ejecuta una acción cuando el jugador selecciona un elemento de la lista desplegable.
OnValueChange	InputField	Ejecuta una acción cuando cambia el valor de la caja de texto.
OnEndEdit		Ejecuta una acción cuando se finaliza el proceso de edición del texto del elemento.
OnSelect		Ejecuta una acción cuando se selecciona la caja de texto.
OnDeselect		Ejecuta una acción cuando se deselecciona la caja de texto.
OnValueChanged	Slider	Ejecuta una acción cuando el valor del slider ha cambiado.
OnChangeValue	Toggle	Ejecuta una acción cuando el valor del elemento ha cambiado. Retorna un valor booleano (true o false)

En el siguiente video se explica cada uno de los elementos de la interfaz de usuario, que componen la UI.



## Eventos Interfaz de Usuario

Para ampliar el tema sobre la programación de eventos de interfaz HUD, se sugiere consultar en el material complementario la página Unity Learn el tema de IU Components.

### 3. Programación de eventos disparadores (interacción con el entorno)

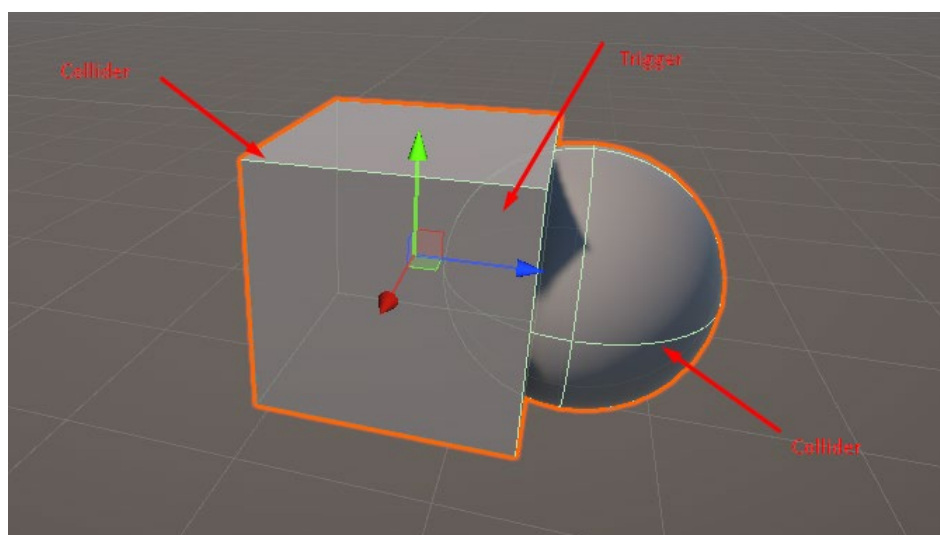
En el diseño de las mecánicas del videojuego es necesario crear interacciones con el entorno, es en este punto donde los eventos ocurridos por condiciones físicas permiten a los jugadores recoger ítems de los niveles de juego, sufrir daño por pasar sobre zonas no permitidas, etc.

Los motores de videojuegos permiten simular los comportamientos físicos del entorno real haciendo uso de diferentes componentes como rigidbodies, materiales físicos, colisionadores etc., permitiendo que las mecánicas sean dinámicas y parecidas a la realidad.

## Triggers

Los triggers son eventos que ocurren cuando los objetos de videojuego se entrelazan entre sí como se observa en la figura.

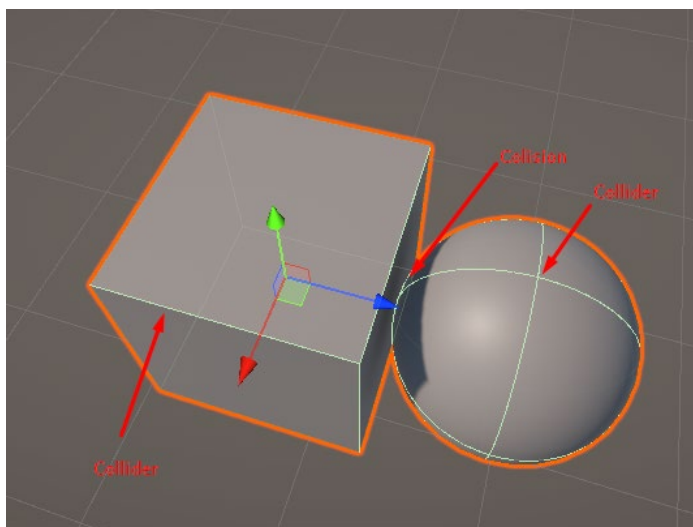
**Figura 13.** Trigger entre gameobjects



## Collision

La colisión es un evento que ocurre cuando dos gameobjects se tocan entre sí, este evento es muy utilizado para provocar daño a los enemigos en un videojuego.

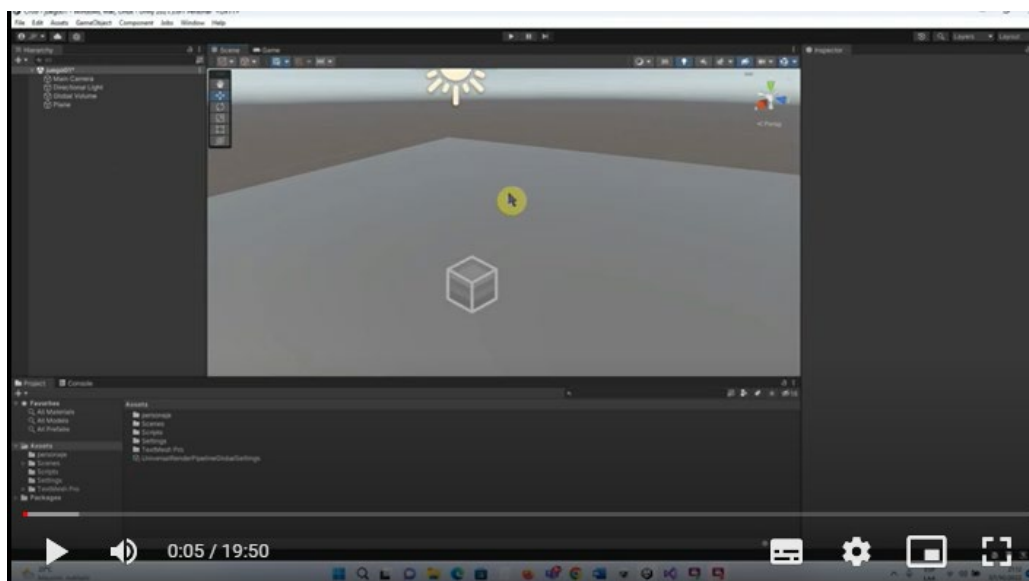
**Figura 14.** Collision



Los tipos de colliders son:

Sphere Collider	Box Collider	Capsule Collider
Este tipo de collider es muy usado en formas redondas.	Es empleado en formas cubicas, edificios, carretera, etc.	Utilizado en los personajes para crear zonas de daño; es muy utilizado en los troncos de los árboles.

En el siguiente video se estudian las físicas y colisiones en el motor de desarrollo.

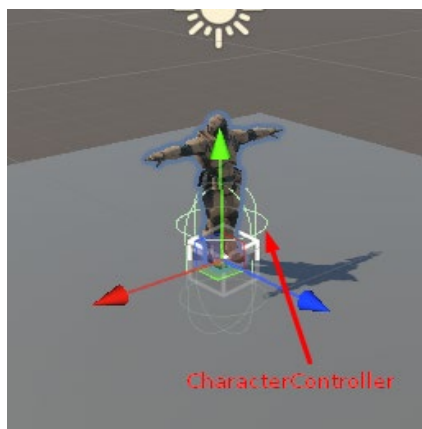


Físicas y colisiones

OnControllerHit

Este tipo de colisión es característico en los personajes de los videojuegos, ya que permite detectar cuando el jugador recoge elementos del nivel de juego.

**Figura 15.** Character controller



A través de los siguientes videos se explicará cómo controlar un personaje y aplicar animaciones:

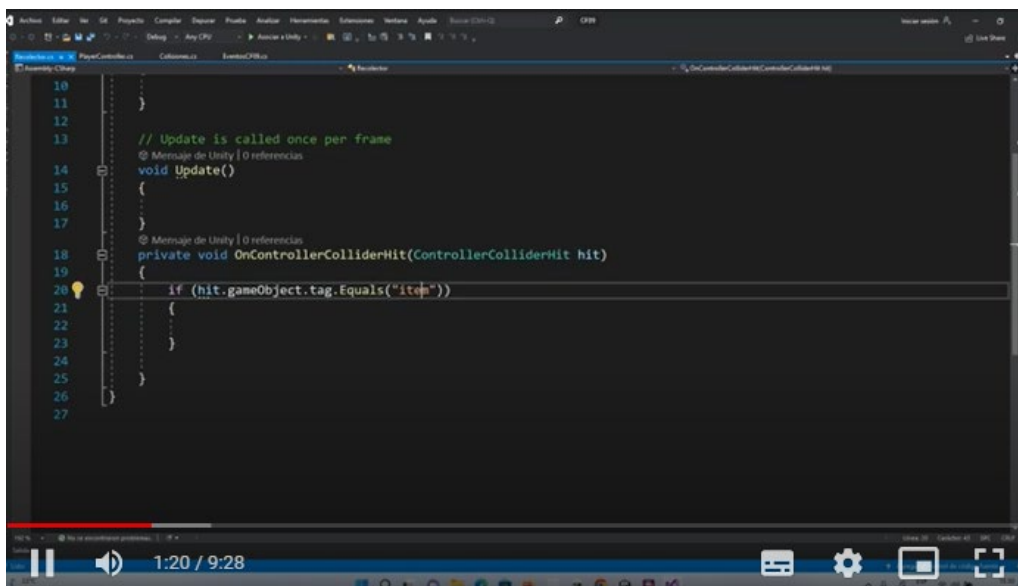
Character controller para controlar el movimiento de un personaje.



Character Controller

Colisión con otros objetos de la escena.





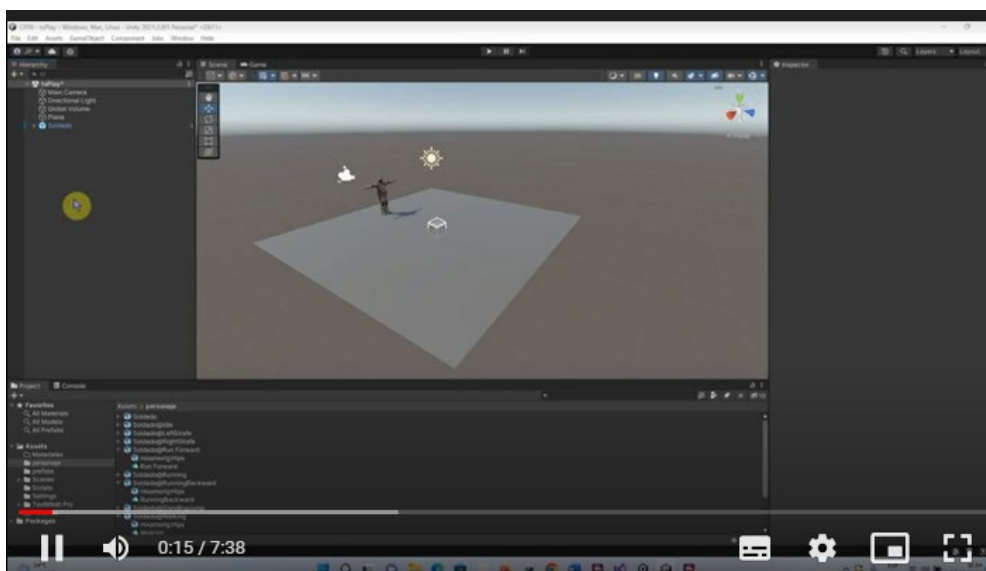
```

10
11
12
13 // Update is called once per frame
14 @ Mensaje de Unity | 0 referencias
15 void Update()
16 {
17 }
18 @ Mensaje de Unity | 0 referencias
19 private void OnControllerColliderHit(ControllerColliderHit hit)
20 {
21     if (hit.gameObject.tag.Equals("item"))
22     {
23     }
24 }
25
26
27

```

Collider Hit

Generación de cámara de seguimiento al personaje.



Cinemachine

Para ampliar el tema sobre la programación de eventos disparadores e interacción con el entorno, consultar en el material complementario la página Unity Learn el tema de Catapult Physics; Forces, and Energy.

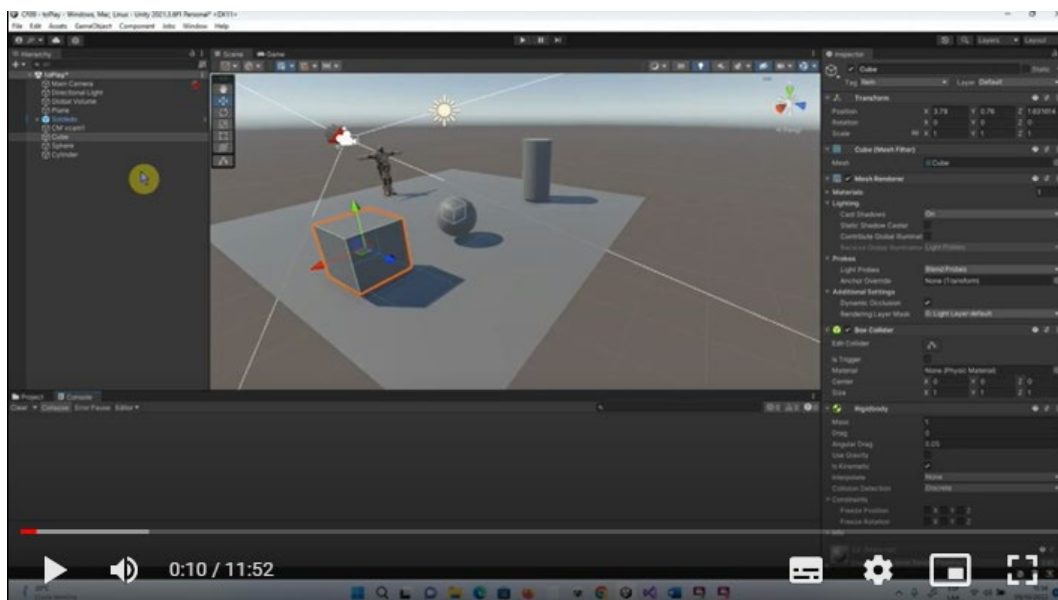
#### **4. Programación de comportamientos relativos a la mecánica de videojuego**

La interacción del jugador con los objetos del videojuego desempeña un papel muy importante en las mecánicas del videojuego, ya que así puede saber el estado actual del nivel, los elementos que ha recolectado para su inventario, qué nivel de vida le queda, entre otros muchos aspectos más.

##### **Interactividad UI Recolección de ítems de juego**

Una de las principales actividades que se deben cumplir en la gran mayoría de mecánicas es la recolección de objetos en el escenario; esto hace que el jugador pueda recoger pociones mágicas, pociones de salud e incluso municiones para sus armas.

Continuando con las colisiones del Character Controller, en el siguiente video se explica la creación de un sistema de recolección y conteo que se visualiza en la interfaz de usuario (UI).



Recolector

## 5. Programación de inteligencia artificial

Los desarrolladores de videojuegos viven preocupados por ofrecer juegos que contengan las mejores experiencias gráficas y una mayor inmersión en cada mecánica desarrollada. La inteligencia artificial en los videojuegos también juega un papel fundamental ya que permite tener grandes comportamientos de los jugadores NPC, haciendo que cada nivel de juego sea más atractivo.

Unity 3D, al igual que muchos motores de desarrollo de videojuegos, posee una estructura bastante robusta que permite integrar de forma sencilla la inteligencia artificial a nuestros juegos. Para tener una visión general de la navegación en Unity, son necesarios los siguientes elementos:

**Slide Simple**

**Ver anexo: CF09\_5\_Slide\_Navegacion\_Unity**

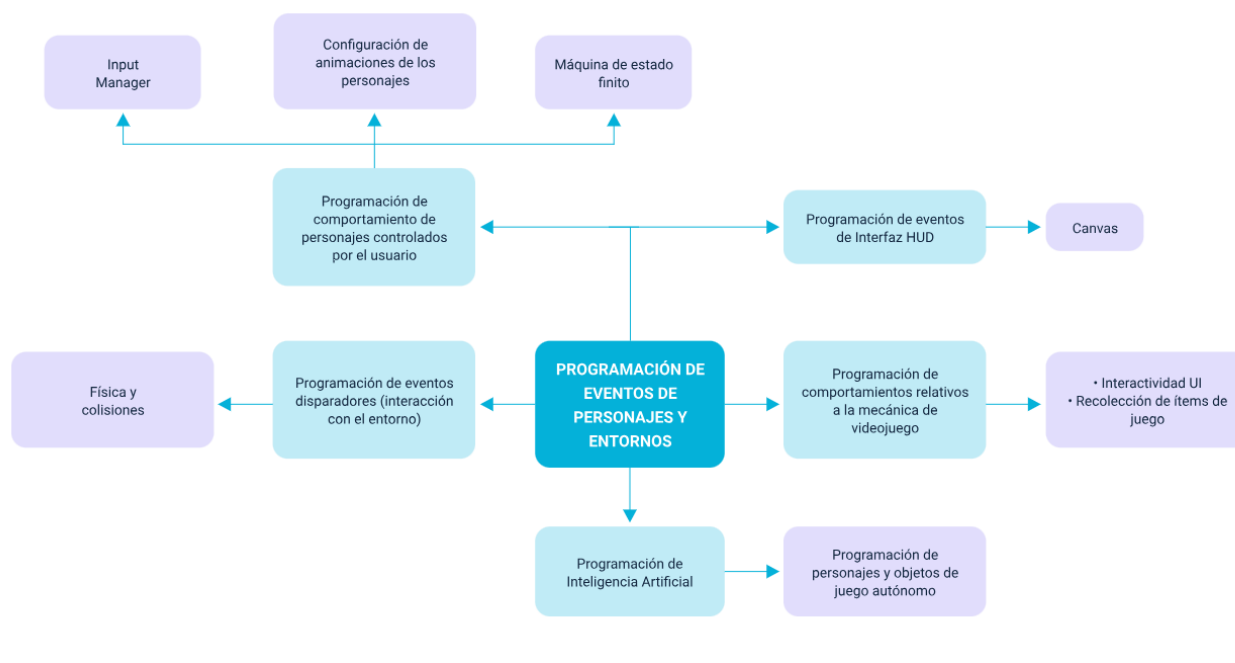
Agregar agente de navegación

Estos son los pasos para agregar el agente de navegación:

Para ampliar el tema sobre Programación de Inteligencia Artificial, consultar en el material complementario la página Unity Learn, el tema de Navigation Meshes.

## Síntesis

En el siguiente esquema se presenta una recopilación de los temas trabajados en este componente formativo “Programación de eventos de personajes y entornos”:



## Material complementario

Tema	Referencia APA del Material	Tipo de material	Enlace del Recurso o Archivo del documento material
Cómo iniciar las pruebas o “testing” en un videojuego.	Unity Manual. (s.f.). <i>Unity Test “Framework” overview.</i>	Otro	<a href="https://docs.unity3d.com/Packages/com.unity.test-framework@1.3/manual/index.html">https://docs.unity3d.com/Packages/com.unity.test-framework@1.3/manual/index.html</a>
Pruebas del sistema.	Kodeco. (2019). <i>Introduction To Unity Unit “testing”</i>	Otro	<a href="https://www.kodeco.com/9454-introduction-to-unity-unit-testing">https://www.kodeco.com/9454-introduction-to-unity-unit-testing</a>

## Glosario

**“Ad hoc”:** que es apropiado, adecuado o especialmente dispuesto para un determinado fin.

**CPU:** una unidad central de procesamiento o CPU, es una pieza de “hardware” que permite que tu computadora interactúe con todas las aplicaciones y programas instalados.

**“Framework”:** es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de “software”.

**QA:** un QA (Quality Assurance) o analista QA es el profesional responsable de asegurar la calidad del “software” y de prevenir fallos en él.

**SetUp:** es una herramienta de los sistemas operativos y los programas informáticos que permite configurar diversas opciones de acuerdo a las necesidades del usuario.

**Teastear / “testing”:** testear es obtener la información necesaria para mejorar el sistema que se está probando, pero también para mejorar los propios procesos de desarrollo y de pruebas. Hay diferentes objetivos en el “testing”: Buscar los defectos. Ganar confianza respecto al nivel de calidad.

**Test Runner:** un test runner es una utilidad que nos permite escribir y correr tests para una aplicación. Existen gran variedad de tests runner como por ejemplo mocha.

**UX:** el diseño UX se refiere al término «diseño de experiencia de usuario», mientras que UI significa «diseño de interfaz de usuario».

## Referencias bibliográficas

Awad, W. (2021). *Game “testing” Automation* Guidance.

<https://www.theseus.fi/bitstream/handle/10024/505977/Game%20Testing%20Automation%20Guidance.pdf?sequence=2&isAllowed=y>

Da Silva Lima, G. et al. (2021). *Devops methodology in game development with Unity*

3D. [https://www.ihci-conf.org/wp-content/uploads/2021/07/04\\_202105C029\\_Lima.pdf](https://www.ihci-conf.org/wp-content/uploads/2021/07/04_202105C029_Lima.pdf)

Koepke, B, Pelletier, B., Adair, D., Jhawar, R., Macaulay, I. & Bielecki, T. (2013). *Agile Game Development*.

<http://kremer.cpsc.ucalgary.ca/courses/seng403/W2013/papers/05GameDevelopment.pdf>



## Créditos

Nombre	Cargo	Regional y Centro de Formación
Claudia Patricia Aristizábal Gutiérrez	Responsable del equipo	Dirección General
Liliana Victoria Morales Gualdrón	Responsable de línea de producción	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Xiomara Becerra Aldana	Instructora Ambiental	Centro de Gestión Industrial
Lubin Andrés Hernández Sanabria	Instructor	Centro de Gestión Industrial
Jesús Ricardo Arias Munevar	Instructor	Centro de Gestión Industrial
Javier Ricardo Luna Pineda	Diseñador Instruccional	Centro para la Industria de la Comunicación Gráfica
Silvia Milena Sequeda Cárdenas	Evaluador Instruccional	Centro de diseño y metrología
Rafael Neftali Lizcano Reyes	Asesor Pedagógico	Centro Industrial del Diseño y la Manufactura
Martha Isabel Martínez Vargas	Productora audiovisual	Centro Industrial del Diseño y la Manufactura
Gloria Amparo López Escudero	Adecuadora instruccional - 2023	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Andrés Felipe Velandia Espitia	Metodólogo para la formación virtual - 2023	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Alix Cecilia Chinchilla Rueda	Asesor metodológico	Centro de gestión de Mercados, Logística y Tecnologías de la Información
Yuly Andrea Rey Quiñonez	Diseñadora web	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital

Manuel Felipe Echavarria Orozco	Desarrollador Fullstack	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Ernesto Navarro Jaimes	Animador y Producción audiovisual	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Carolina Coca Salazar	Evaluadora de contenidos inclusivos y accesibles	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Lina Marcela Pérez Manchego	Validadora de recursos educativos	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Leyson Fabian Castaño Pérez	Validadora de recursos educativos	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital