# UNIVERSIDAD TECNOLÓGICA DEL NORTE DE GUANAJUATO

# Texto El contenido generado por IA puede ser incorrecto.

Ingeniería en Desarrollo y Gestión de Software

# Unidad II.

**Programación orientada a objetos para videojuegos**

**Asignatura:**

Creación de Videojuegos

**PRESENTA**:

# Herrera Ramiro

**Grupo**:

GIDS5101-e

**A 10 de octubre del 2025**

**Dolores Hidalgo Cuna de la Independencia Nacional, Guanajuato.**

**TABLA DE CONTENIDO**

[Tutorial 02 1](#_Toc210383647)

Tutorial 02 - 🛠 PROCEDIMIENTO - PASO A PASO 🛠

**COLISIONADORES (COLLIDERS)**

**Paso 1:** Seleccione **PlayerObject** y luego seleccione el botón **Add Component** en la ventana del inspector. Busque y seleccione "**Box Collider 2D**" para agregar un Box Collider 2D al PlayerObject.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Paso 2:** Necesitaremos saber cuándo el jugador choca con un enemigo, así que agregue un **Box Collider 2D** al **EnemyObject** también.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**COMPONENTE RIGIDBODY 2D**

**Paso 3:** Con **PlayerObject** seleccionado, haga clic en el botón **"Add component"** en la ventana Inspector, busque "**Rigidbody 2D**" y agréguelo al PlayerObject.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 4:** En el menú desplegable establezca los valores a las siguientes propiedades:

* **Body Type**: Dynamic
* **Mass**: 1
* **Linear Drag**: 0
* **Angular Drag**: 0
* **Gravity Scale**: 0

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 5:** Seleccione el **EnemyObject** y agregue un Componente **Rigidbody 2D** de tipo **Dinámico** para él también con las mismas propiedades.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 6:** Vaya a la opción **Edit → Project settings → Physics 2D** y cambie el valor para la gravedad **Y** de **–9,81** a **0**.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**ETIQUETAS Y CAPAS**

**Paso 7:** Selecciona el **PlayerObject**. En el menú desplegable **Tag** en la parte superior izquierda del Inspector, seleccione la etiqueta de **Jugador** para agregar una etiqueta a nuestro PlayerObject.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 8:** Crear una nueva etiqueta llamada **"Enemy"** y úsala para configurar la etiqueta del objeto **EnemyObject**.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 9:** Seleccionar en el menú desplegable **Layers** y seleccionar **"Add layer"**. Debería ver la ventana Layers.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Paso 10:** Ahora seleccione el **PlayerObject** nuevamente para ver sus propiedades en el Inspector. Seleccione la capa de **bloqueo** que acabamos de crear en el menú desplegable.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**SORTING LAYERS**

**Paso 11:** En el componente **Sprite Renderer** en la ventana Inspector, seleccione la opción **Sorting layer** Menú desplegable de capa y seleccione **"Add Sorting Layer"**.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 12:** Agregue una capa de ordenamiento llamada **"Characters"**, y luego haga clic en **PlayerObject** nuevamente para ver su Inspector y seleccione la nueva **Capa de ordenamiento** de caracteres del menú desplegable.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 13:** Seleccione nuestro **EnemyObject** y establezca su **Capa de ordenamiento** en **Characters**, porque queremos que los enemigos también aparezcan en la parte superior de las cosas.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**PREFABS**

**Paso 14:** Primero, crea una carpeta llamada **"Prefabs"** en nuestra carpeta **Assets** en la vista Proyecto.

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

**Paso 15:** Luego seleccione nuestro **PlayerObject** de la vista de **Hierarchy** y simplemente arrástralo a la carpeta **Prefabs**.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 16:** Ahora puedes eliminar de forma segura el **PlayerObject** de la vista de Hierarchy.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Paso 17:** Haga lo mismo con **EnemyObject**: arrástralo a la carpeta **Prefabs** y eliminar el **EnemyObject** original de la vista Hierarchy.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**SCRIPTS: MOVIMIENTO DEL PERSONAJE**

**Paso 18:** Seleccione nuestro **PlayerObject Prefab** y arrástralo a la vista Hierarchy.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 19:** Desplácese hasta la parte inferior del Inspector y presione el botón **Add Component**. Escriba la palabra **Script** y seleccione **"New Script"** y nombrarlo como **"MovementController"**.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 20:** Cree una nueva carpeta llamada **"Scripts"** en la vista Project. El nuevo script se habrá creado en la carpeta Assets de nivel superior en la vista Project. Arrastre el script **MovementController** a la carpeta **Scripts** y luego haga doble clic en él para abrirlo en Visual Studio.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Paso 21:** Agreguemos las siguientes propiedades a la clase:

public class MovementController : MonoBehaviour

{

//Velocidad de los personajes

public float movementSpeed = 3.0f;

//Representa la ubicación del Player o Enemy

Vector2 movement = new Vector2();

//Referencia a RigidBody2D

Rigidbody2D rb2D;

}

**Paso 22:** Establezcamos la referencia RigidBody2D en el método Start:

void Start()

{

//Establece el componente Rigidbody2D enlazado

rb2D = GetComponent<Rigidbody2D>();

}

**Paso 23:** Programemos el método FixedUpdate:

private void FixedUpdate() {

//Captura los datos de entrada del usuario

movement.x = Input.GetAxisRaw("Horizontal");

movement.y = Input.GetAxisRaw("Vertical");

//Conserva el rango de velocidad

movement.Normalize();

rb2D.velocity = movement \* movementSpeed;

}

Interfaz de usuario gráfica, Texto, Aplicación

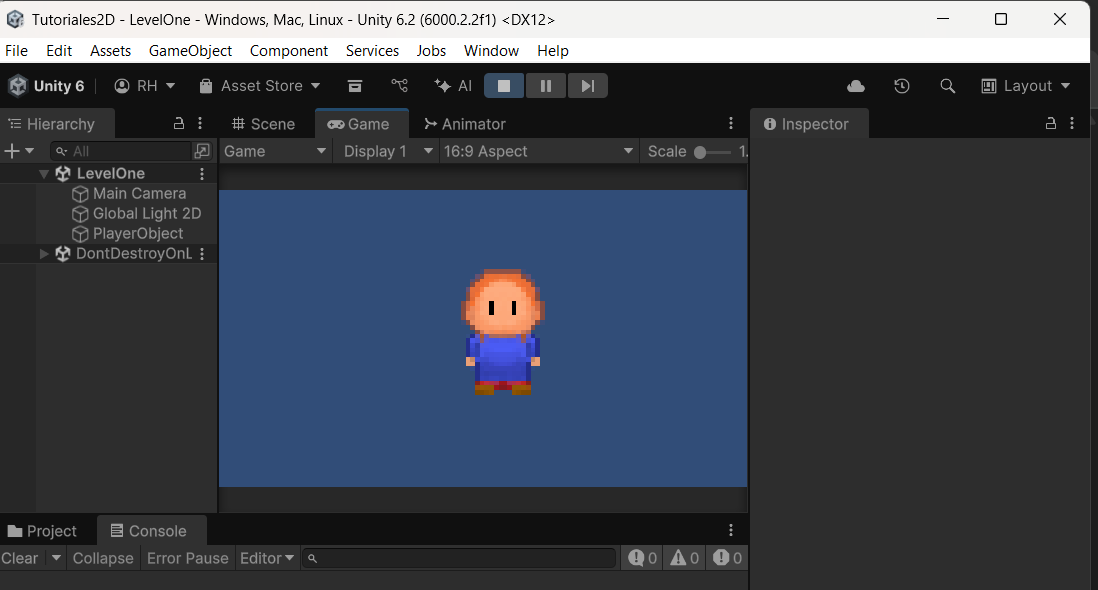
El contenido generado por IA puede ser incorrecto.

**Paso 24:** Regrese al Editor de Unity y asegúrese de ver nuestro PlayerObject en la vista de Hierarchy. Para agregar el script a nuestro PlayerObject, arrastre el **MovementController script** de la carpeta Scripts, en PlayerObject en la jerarquía, o arrástralo al Inspector cuando el PlayerObject esté seleccionado.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 25:** Ahora presione el botón **Play**. Deberías ver a nuestro personaje de jugador caminando en su lugar. Presione las teclas de flecha o **W, A, S, D** en su teclado y mírala moverse.



Si no funciona correctamente, es probable que sea un error en el código al momento de llamar los componentes/acciones y debas de configurar lo siguiente en Edit – Player – Active Input Handling

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**MÁQUINA DE ESTADOS DEL ANIMATOR**

**Paso 26:** Vaya a la carpeta **Animations → Controllers** y haga doble clic en el Objeto **PlayerController**.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 27:** Haga clic y arrastre sus objetos de estado de animación hasta que aparezcan en la pantalla, con el reproductor inactivo a un lado, y las animaciones de caminatas de jugador agrupadas.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Paso 28:** Seleccione y haga clic con el botón derecho en el estado de animación **"player-idle"** y seleccione **"Set as Layer Default State"**.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Paso 29:** Ahora seleccione y haga clic con el botón derecho en **"Any State"** y seleccione **"Make Transition"**. Aparecerá una línea con una flecha adjunta y siguiendo alrededor de su ratón. Haga click en **"player-walk-east"** para crear una transición.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Paso 30:** Ahora haga lo mismo para el resto de los estados de animación: haga clic con el botón derecho en **Any State → Make Transition** y seleccione cada uno de los Estados de animación para crear una transición.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Paso 31:** Debe crear un total de **cinco flechas de transición blancas** que apunten desde Any State a los cuatro estados de animación de caminata del jugador y al jugador inactivo player-idle.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**PARÁMETROS DE ANIMACIÓN**

**Paso 32:** Seleccione la pestaña **Parámetros** en el lado izquierdo de la Ventana Animator. Presione el símbolo **+** y seleccione **"Int"** en la lista desplegable. Cambie el nombre del parámetro de animación creado a **"AnimationState"**.

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

**Paso 33:** Seleccione la línea de transición blanca que conecta **Any State** con el estado en el Inspector, cambie la configuración para que coincida.

**Paso 34:** **Desmarque Has Exit Time** porque queremos interrumpir una animación si nuestro usuario presiona una tecla diferente.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Paso 35:** En la parte inferior del inspector, verá un área titulada **"Conditions"**. Haga clic en el símbolo **+** en la parte inferior derecha y seleccione **AnimationState → Equals**, e ingrese el valor correspondiente para cada transición:

|  |  |
| --- | --- |
| Transición | Condición |
| Any State to player-walk-east | AnimationState Equals = 1 |
| Any State to player-walk-west | AnimationState Equals = 3 |
| Any State to player-walk-north | AnimationState Equals = 4 |
| Any State to player-walk-south | AnimationState Equals = 2 |
| Any State to player-idle | AnimationState Equals = 0 |

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**ACTUALIZACIÓN DEL SCRIPT MOVEMENTCONTROLLER**

**Paso 36:** Agreguemos la referencia del objeto Animator y la definición de estados al script:

public class MovementController : MonoBehaviour

{

//... propiedades anteriores ...

Animator animator; //Referencia a componente animator

string animationState = "AnimationState"; //Variable en Animator

//Enumeración de los estados

enum CharStates

{

walkEast = 1,

walkSouth = 2,

walkWest = 3,

walkNorth = 4,

idleSouth = 0

}

}

**Paso 37:** Inicializamos en el método start el valor del componente Animator:

void Start()

{

//... código anterior ...

//Establece valor de componente Animator el objeto ligado

animator = GetComponent<Animator>();

}

**Paso 38:** Modifiquemos el método Update:

void Update()

{

this.UpdateState(); //Invoca al método

}

private void UpdateState() {

if (movement.x > 0) { //ESTE

animator.SetInteger(animationState, (int)CharStates.walkEast);

} else if (movement.x < 0) { //OESTE

animator.SetInteger(animationState, (int)CharStates.walkWest);

} else if (movement.y > 0) { //NORTE

animator.SetInteger(animationState, (int)CharStates.walkNorth);

} else if (movement.y < 0) { //SUR

animator.SetInteger(animationState, (int)CharStates.walkSouth);

} else { //IDLE

animator.SetInteger(animationState, (int)CharStates.idleSouth);

}

}

**Paso 39:** Modifiquemos el método FixedUpdate:

private void FixedUpdate() {

MoveCharacter(); //Método definido para ingresar la dirección

}

private void MoveCharacter() {

//Captura los datos de entrada del usuario

movement.x = Input.GetAxisRaw("Horizontal");

movement.y = Input.GetAxisRaw("Vertical");

//Conserva el rango de velocidad

movement.Normalize();

rb2D.velocity = movement \* movementSpeed;

}

**Paso 40:** Regresa al Animator de Unity y verifica cada estado de transición. A medida que avanza por cada flecha de transición, recuerde desmarcar cuadros como **Exit Time, Fixed Duration, Can Transition to Self** y establecer **Transition Duration (%)** a **0**.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Paso 41:** Seleccione cada estado de animación de caminata del jugador objeto y ajuste la **velocidad a 0,6**, y ajuste **player-idle a 0.25**.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**RESULTADO FINAL**

**Paso 42:** Ahora ha configurado una gran parte de las animaciones del reproductor necesarias para nuestro juego. Pulsamos el botón **Play** y movemos nuestro personaje por la pantalla con las teclas de flecha o **W, A, S, D**.

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.