

Trabajo Práctico - Programación

2)A) A partir de la línea 128 en adelante, en la función **PerformMovement()** se puede identificar un llamado a la función **body.Cast()**, en la que **body** es una instancia de la clase **Rigidbody2D** que se adjunta al proyecto (línea 35, línea 75). Agregar un componente **Rigidbody2D** a un sprite hace que este se vea afectado por el motor de físicas, lo cual significa que será susceptible a la gravedad y podrá controlarse vía scripts usando fuerzas.

A su vez, la función **Cast()** sería la implementación de la técnica de **raycast**, esta función debe recibir cuatro parámetros: **Cast(Vector2 move, ContactFilter2D contactFilter, RaycastHit2D[] hitBuffer, float distance = Mathf.Infinity);**

move: Vector2 que indica dirección del raycast

contactFilter: Filtros de contacto, o colisión a excluir

hitBuffer: Array de RaycastHit2D para almacenar los hits encontrados

distance: Distancia máxima del raycast

En la variable **hitBuffer** es devuelto un array, el cual contiene una colección de objetos **RaycastHit2D**, cada uno de los cuales contiene los datos relacionados a cada colisión encontrada. Cada elemento de **hitBuffer** será un objeto RaycastHit2D que contiene datos como:

point: el punto exacto de la colisión en coordenadas mundiales

normal: la normal de la superficie en el punto de colisión

distance: la distancia a la que ocurrió la colisión

collider: referencia al collider con el que se colisionó

rigidbody: referencia al rigidbody asociado al collider

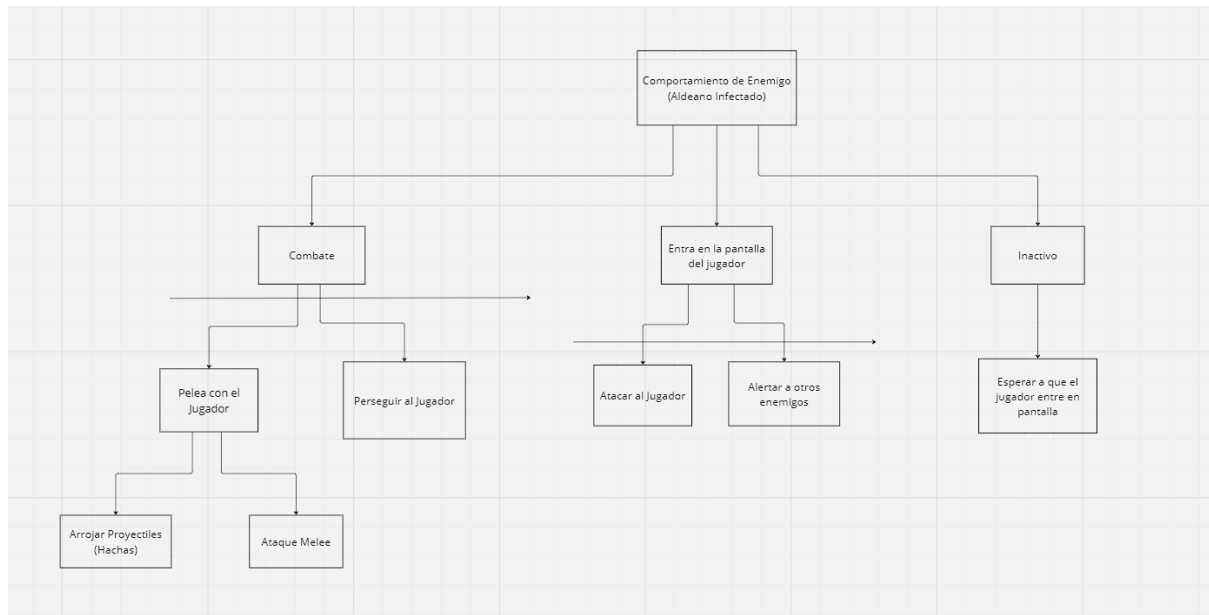
transform: transform del GameObject con el que hubo colisión

B) Los tokens se almacenan en un array de tipo **TokenInstance[]**. La variable **tokens** es un arreglo de objetos de la clase **TokenInstance**. Un arreglo es una estructura de datos que puede contener múltiples elementos del mismo tipo. En este caso, el arreglo **tokens** almacena todas las instancias de **TokenInstance** que se animan.

Cada elemento en el arreglo **tokens** es una instancia de la clase **TokenInstance**. Los elementos del arreglo se pueden acceder individualmente usando índices. Por ejemplo **tokens[0]** se refiere al primer elemento del arreglo y **tokens[1]** al segundo elemento.

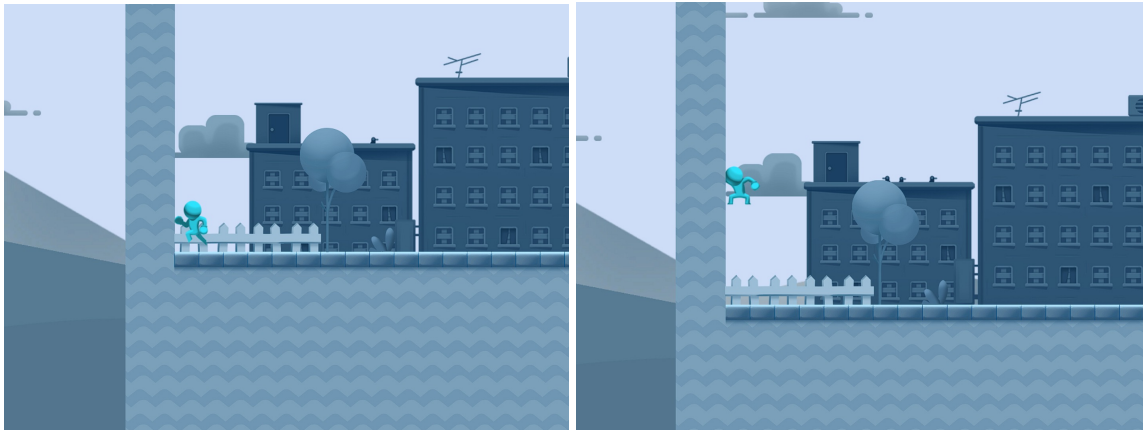
En la función **Update()** se recorre este arreglo usando un bucle **for** para animar todos los tokens al mismo tiempo. El bucle se repite a través de cada elemento en el arreglo **tokens** y actualiza su animación en cada cuadro.

C) Resident Evil 4 (2005): Aldeano Infectado

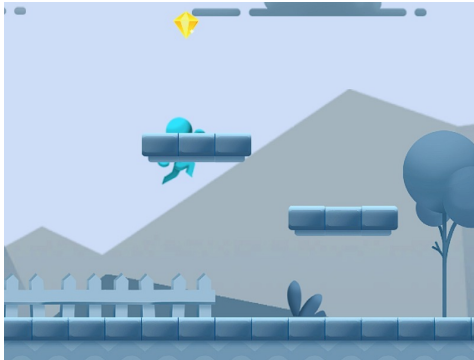


D)

Misaligned Collision: El error ocurre cuando la colisión puede tener la forma correcta pero no está alineada con el modelo 3D. En este caso al existir una plataforma secreta invisible se puede ver como el personaje flota sobre ella.



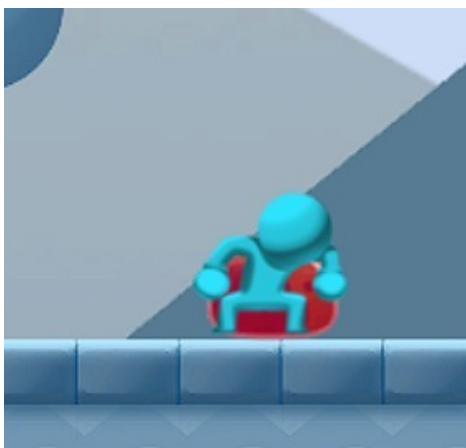
Missing Collision: Cuando a un objeto no se le aplica una colisión, el jugador puede atravesarlo o caer al suelo. En este caso la plataforma al no tener colisión el jugador puede traspasarla.



Missing Collision: Al no tener una colisión adecuada para la moneda el juego no detecta la interacción entre el jugador y la moneda



Missing Collision: Al no tener una colisión adecuada para el enemigo el juego no detecta la interacción entre el jugador y el enemigo



Gameplay Cues: Son un tipo de SFX que juega un papel clave en la experiencia del jugador, en este caso sería el de verificar la obtención de la moneda por un medio no visual, ya que al tocar la moneda esta desaparece, pero a diferencia de otras monedas no tiene sonido.

