**Trabajo Práctico - Programación**

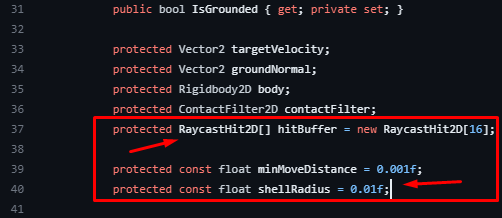
Leer la función ***PerformMovement*** del script /Assets/Scripts/Mechanics/**KinematicObject.cs** e identificar si se utilizan **colliders o raycasts** para detectar las superficies que el objeto tiene por delante.

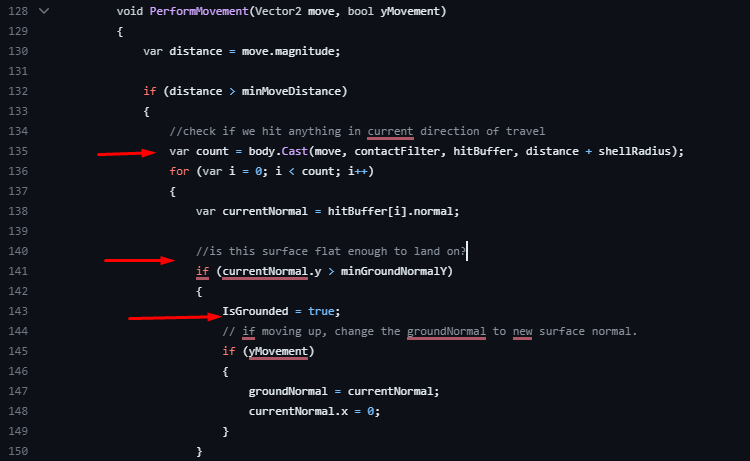
Tener en cuenta la variable “***hitBuffer***” y la llamada a la función ***Cast*** en la línea **135.**

Desde la linea 128 en adelante, en la función **“PerformMovement()”**se puede identificar un llamado a la función **“body.Cast()”** en donde se ve que el **“body”** es una instancia de la clase **“Rigidbody2D**”, lo cual se denota que se adjunta este componente al proyecto (linea 35)

La implementacion fue el raycast en el codigo, esto es requerido para detectar las distintas superficies y objetos que tiene dentro de un radio el personaje jugable, verificando si de esta manera existe colisiones a su alrededor o de los objetos.

Adicional la linea 128 hace un llamado a la funcion **body.cast(**) en donde vemos que el body es una instancia de la clase **rigibody2D**, agregar dicho componente a un cuerpo hace que se vea afectado por ls fisicas, como por ejemplo sera un cuerpo afectado por la gravedad.





¿Qué tipo de estructura de datos se utiliza para almacenar los tokens (objetos coleccionables) para que puedan ser animados todos al mismo tiempo en el archivo /Assets/Scripts/Mechanics/**TokenController.cs**

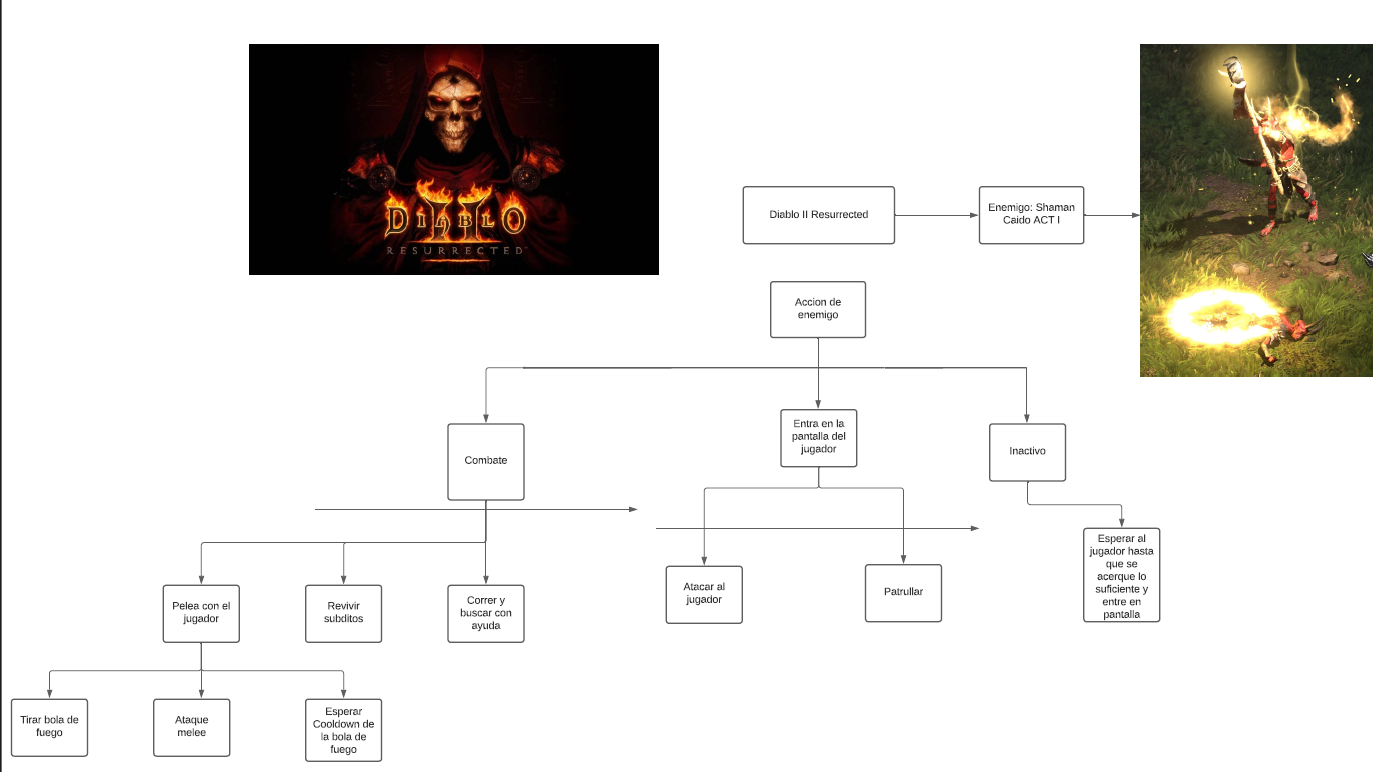
Tener en cuenta la variable “***tokens***” y que se puede acceder a ***cualquiera de sus elementos*** desde la función ***Update***()

Se utilizó un **array** como estructura de datos para almacenar los **Tokens**

**Diagrama de comportamiento**

**Diablo II resurrected**

El shaman caido de acto 1 de diablo II como este y muchos demas enemigos solo entran en accion cuando el jugador se acerca lo suficiente tenerlo en pantalla y al entrar en pantalla atacaran inmediatamente ya sea con una bola de fuego como ataque a distancia o melee si se acerca el jugador al shaman caido lo suficiente, o realizaran diferentes acciones, por ejemplo si alguno de sus subditos caidos se encuentra eliminado y su cuerpo en el mapa, procederan a revivirlo hasta que no haya ninguno muerto y continue con el ataque, si el mismo es herido y no es rematado por el jugador, este intentara alejarse del jugador tratando de activar otros enemigos en pantalla para que ataquen al jugador y se encontrara patrullando hasta que el jugador se acerque lo suficiente para volver a iniciar el ataque pero no se volvera inactivo a menos que el jugador se aleje lo suficiente para que se reinicie su estado.



Jugar el juego corriendo el ejecutable /Build/**Platformer.exe** y redactar los bugs encontrados y sus posibles causas.

**Missing collision**: el jugador se le es imposible permanecer en esta plataforma ya que muy probablemente el collider no este asignado para identificar al personaje



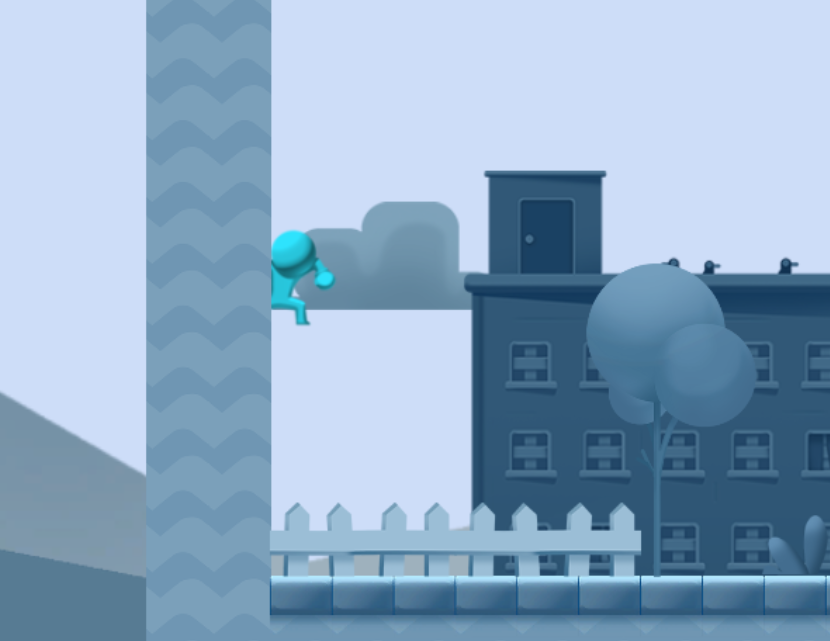
**Model misplaced:** Se denota durante todo el juego que el jugador personal no camina apoyandose en el suelo sino en el aire, a diferencia de los enemigos que su masa si se encuentra en contacto con el suelo.



**Object with missing collision:** Aca podemos apreciar como este objeto o coleccionable no tiene ningun tipo de colision lo que ocasiona que el jugador no pueda tomar este item o hacer uso del mismo.



**Missin collision:** Se puede apreciar otra falta de colision con la pared del lado izquierdo al inicio del juego, lo que puede ocasionar que el jugador pierda alguna vida con tal de querer explorar



Al ser un juego de plataformas es muy comun observar valga la redundancia, plataformas que no posea una colision o no se puedan subir en ellas por parte del game desing, ya sea para molestar al jugador haciendolo creer que dicha plataforma sirve y es utilizable para el transcurso del juego o solo para una trampa y provocar ya sea una muerte intencional o un breve retraso en su juego. sin embargo estos son los mas destacables a señalar.