**INFORME DEL PROYECTO:**

**MAZE RUNNER**

**Ejecucion del proyecto :**

Para la creación de este proyecto se empleo el programa Godot en su versión 4.3 empleando el lenguaje de programación C#, esta version de Godot funciona con .Net 6, por lo cual tendrá que tenerlo instalado, para la ejecución del proyecto deberá importar la carpeta del proyecto a una versión de Godot igual o superior y ejecutarlo desde el editor.

**Estructura del proyecto :** La lógica del proyecto se maneja principalmente de manera separada de la parte gráfica, ésta se encuentra contenida fundamentalmente en los scripts Player.cs, donde se controla la lógica asocida con la creación de los jugadores y Maze.cs, donde se crea, modifica y se almacena el estado del laberinto. En cuanto a la parte gráfica, esta se controla a través del script GameGenerator.cs, que está asociado al nodo de mismo nombre, que es el nodo raíz de la escena del juego.

**Algoritmo de generación del laberinto :**

**Algoritmo de Backtracker Recursivo :** El funcionamiento de este algoritmo es bastante sencillo, primeramente se elige una casilla de salida en un tablero cuadrado de dimensión NxM y luego se desplaza aleatoriamente de manera horizontal o vertical a una casilla vecina válida, creando un camino entre ellas. Para que una casilla sea considerada válida esta debe encontrarse dentro de los límites del tablero y no haber sido visitada previamente, en caso de no encontrarse caillas válidas se regresa recursivamente a la casilla anterior. El algoritmo se ejecuta hasta que se hagan agotado todos los caminos posibles.

**Principio de funcionamiento :** Ya sabemos como funciona el algoritmo ahora toca entender por que, y como podemos asegurar que toda casilla del tablero es accesible desde cualquier otra. La eficacia de este algoritmo se basa en 2 hechos: primeramente que al pasar de una casilla a otra se crea un camino entre estas por lo que toda casilla visitada será accesible, y el hecho de que se recorren todas las casillas del tablero. Como podemos asegurar esto? Pues supongamos que existe una casilla o un grupo de estas que no son accesibles, la única forma de que esto suceda es que no sean válidas o sea que ya hallan sido visitadas ( lo cual es imposible ya que en ese caso serían accesibles) o que no existan casillas vecinas que sean accesibles ( lo que implica que estas son exteriores al tablero lo que tambien es imposible). Por lo tanto como todas las casillas visitadas son accesibles y todas las casillas son visitadas, entonces todas son accesibles.

**Disposición de las trampas** : Las trampas son dispuestas de manera completamente aleatoria a través de las casillas libres del laberinto, ignorando por supuesto las posiciones importantes como la salida y la meta. Existen 3 tipos de trampa : una que aumenta los puntos de vida del personaje, otra que causa daño y por último una que congela los movimientos e impide moverse por el resto del turno.