

1. Terminar nivel

Para este paso utilizamos el tag de los enemigos ("enemies" y "Tower1") donde en el main cámara en el script de GameController utilizamos una variable de texto, a la cual se le asignaron un toString de la suma de los array.length de los enemigos con tag Enemies y Tower 1.

Por medio de canvas y creamos dos imágenes, las cuales tenía texto y un botón de reinicio, con un if en el script de GameController, el cual tenía como condición que el .length de los dos array de los enemigos fueran igual a 0, se compró que el enemigo haya eliminado a todos los enemigos y se le mostró la pantalla de victoria.

Para cuando moría, se creó una variable bool en el script megaman () la cual tiene de nombre "vivo", en la cortina de muerte antes de que se destruyera el megaman se le puso un delay de 1 segundo para que justo después de la animación de muerte la variable vivo pase a ser true y ya en el script de gameControl, con ayuda de un if el cual tiene como condicion que vivo sea verdadero, se le muestra la pantalla de derrota al jugador

2. Tres tipos de enemigos.

Para el enemigo volador se utilizó la variable bool "canMove" de AIPath para que cuando el megaman esté en el rango este comience a perseguirlo. Con un trigger en el animator y nuevamente con el canMove se ejecuta la muerte cuando el enemigo colisione con dos de nuestras balas, esto por medio de el tag bullet.

Para el enemigo estático 1 utilizamos un raycast para que cuando éste colisione con el collider de megaman el enemigo dispare, el tiempo de disparo funciona igual que el cooldown de megaman para seguir disparando. Para este enemigo se creó la bala enemy bullet la cual no colisiona con enemigos pero si con megaman

Para el enemigo estático 2 utilizamos la función de Physics2D.OverlapCircle para crear el rango de ataque, para los disparos instanciamos dos balas enemy las cuales rotamos cada una 45 grados hacia la derecha y hacia la izquierda, utilizando Quaternion.Euler.

Para todos los enemigos la muerte se ejecutó por medio de un trigger.

3. Estado de derrota

Para esto se agregó en el if del método OnCollisionEnter2D que cuando la bala enemy colisionara con megaman se ejecutara su muerte y una variable bool llamada vivo se pasa a verdadera, esto para sacar la pantalla de perdiste.

Datos extra:

Se retiró la habilidad de salto doble y se disminuyó el tiempo de disparo de megaman, también el enemigo estático dos tiene un rango mucho más amplio que los demás.

Reto del desarrollador: Pasarse el juego en menos de 10 min (Premio de \$5.000 pesos COP), escribir a este discord *Juan V.P. Andres*#0252 para reclamar premio.

Links:

<https://juanvsandres720.itch.io/megaman-javs>

<https://github.com/Dap20040812/Proyecto-Final-Megaman-Prado-Valderrama.git>