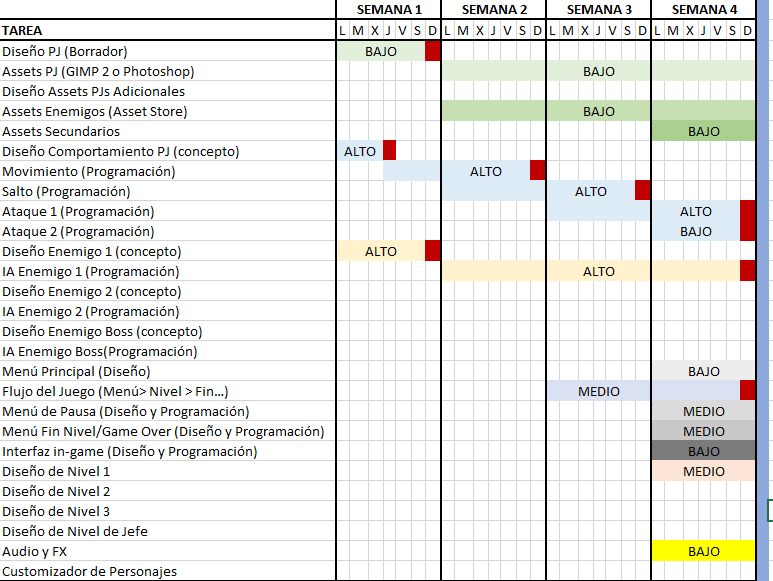


**PEC2 – VERSIÓN PARCIAL**



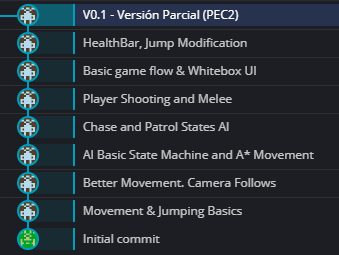
EL PROCESO HASTA EL MOMENTO

Para la versión parcial nos habíamos propuesto tener las mecánicas base ya establecidas; dejando de lado todo el arte y assets innecesarios.



Gantt para la Versión Parcial

Si miramos el árbol Git de avance veremos que empezamos desarrollando el movimiento base del jugador; y eso se llevó bastante tiempo hasta conseguir un resultado satisfactorio.

****

Árbol de GitKraken

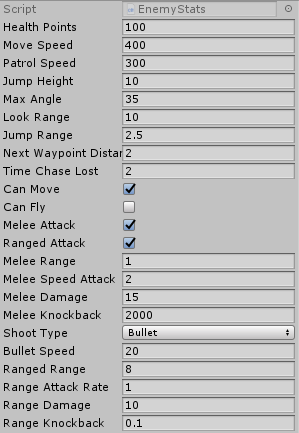
Usamos el RigidBody2D para aplicar movimiento al personaje a través de un CharacterController2D.

Aún teniendo un resultado bastante satisfactorio aún queda trabajo por hacer en esta área; pues el jugador no tiene control de lo alto que el PJ puede saltar; es decir, pulsar el botón de salto o mantenerlo pulsado actualmente da el mismo resultado.

Tras acabar con el movimiento base del jugador; incluyendo el movimiento de la cámara para seguir al jugador, pasamos a la IA de los enemigos; que es donde se ha ido la mayor parte del tiempo de desarrollo de este mes.

A través de una FSM (Finite State Machine) de 3 estados hemos dado vida a los enemigos básicos; estos estados son:

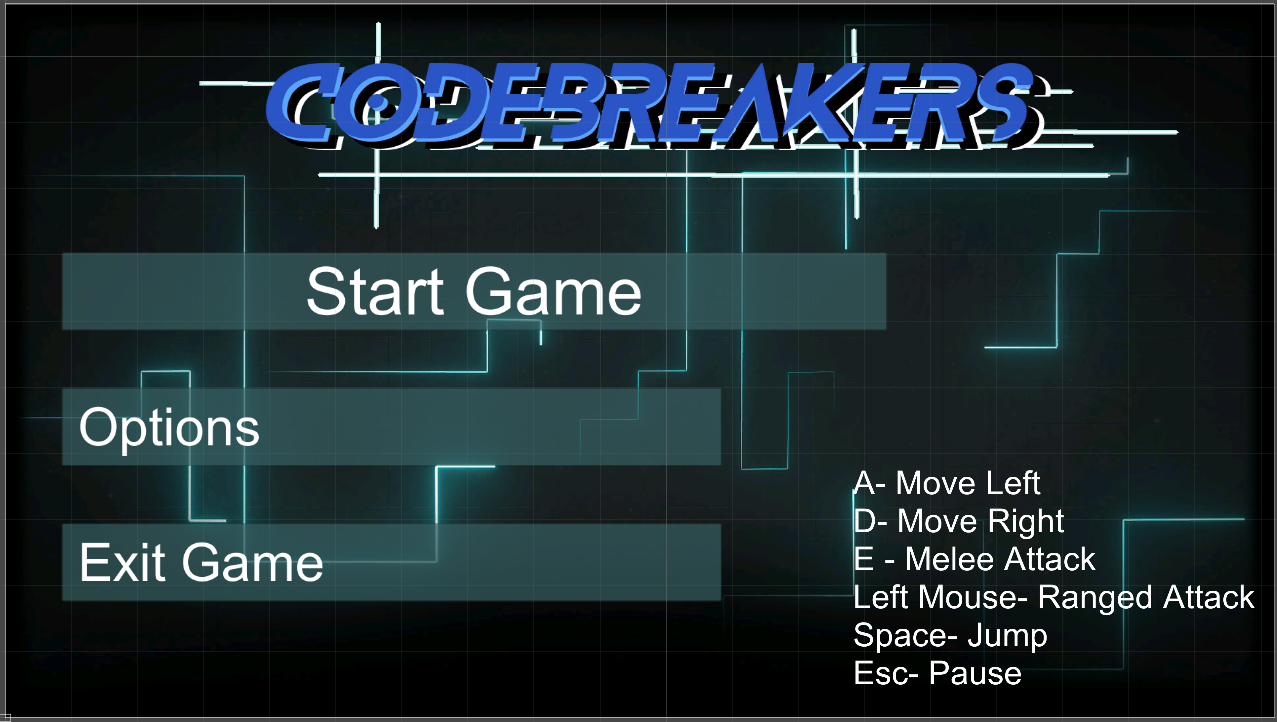
* PATROL STATE: El enemigo se mueve de un *waypoint* al siguiente. Algunos enemigos son estáticos; y su “patrol state” será mantenerse en el sitio.
* CHASE STATE: El enemigo encuentra al jugador y se disponer a perseguirle o atacarle si está a rango. Es importante destacar el orden de prioridades en este estado:
  + Si puede, y está a rango, golpeará cuerpo a cuerpo al jugador, con el objetivo de apartarlo.
  + Si puede, y está a rango, atacará al jugador disparándole. Actualmente, al igual que el jugador, disparan “balas”, que pueden esquivarse con facilidad.
  + Si puede, persigue al jugador hasta tenerlo a rango de un ataque o perderlo de vista.
* SEARCH STATE: Cuando pierde de vista al jugador; antes de volver al estado de patrulla, mira alrededor durante un tiempo a ver si lo encuentra.

****Actualmente se han creado distintos tipos de enemigos; aunque por falta de tiempo, son indistinguibles desde fuera. Para realizar los distintos tipos de enemigos se han empleado ScriptableObjects que contienen la información, estadísticas y parámetros de cada tipo de enemigo.

Estadísticas modificables para cada tipo de enemigo

Para esta versión se ha recreado, muy a grandes rasgos, un posible nivel con variedad de enemigos en el que el jugador deberá llegar al objetivo sin morir.

Al iniciar el juego iremos a la pantalla del menú inicial.



Menú Inicial de la Versión Parcial

Cómo se puede apreciar hay algunos assets ya desarrollados; los cuales corresponden a parte de la Práctica Final de la asignatura de *Experiencia de usuario e interfaces*.

Actualmente, el botón de Opciones no está habilitado.

Si pulsamos sobre *Start Game* se iniciará el nivel y tomaremos control sobre el personaje (el cuadrado blanco)

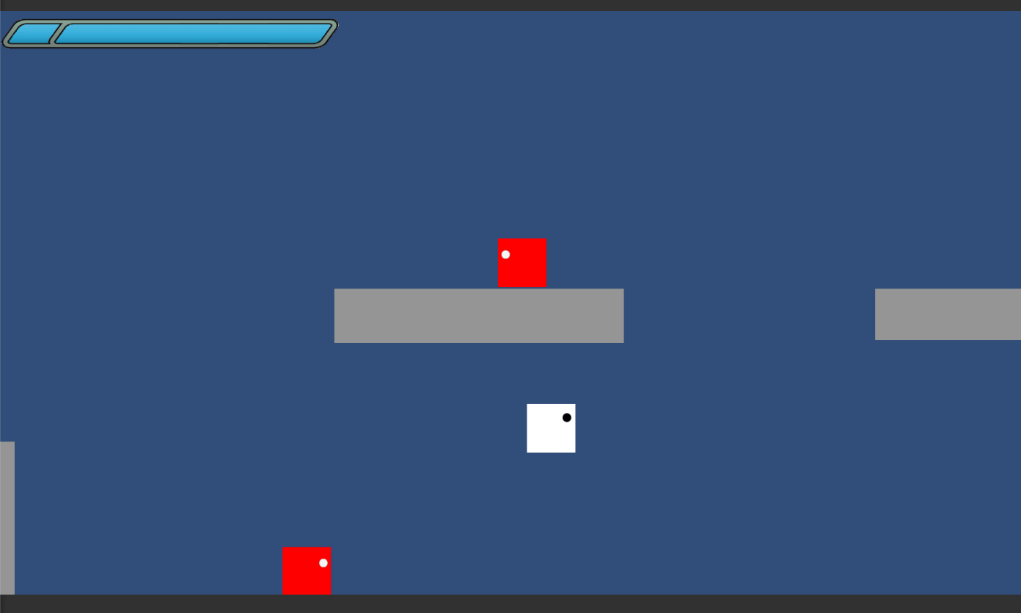
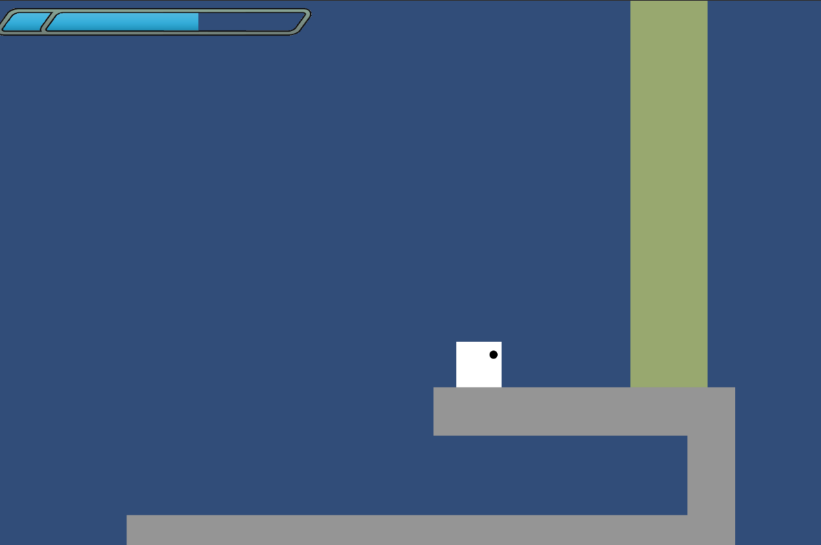


Imagen del nivel de juego

Los cuadrados rojos representan a los enemigos; los cuales nos perseguirán y nos dispararán al vernos.

En la esquina superior derecha se muestra nuestra barra de salud. Si perdemos toda, saltará la pantalla de Game Over.

Esta pantalla también aparecerá cuando superemos el nivel, es decir, cuando lleguemos a la zona marcada en amarillo:



Final del Nivel

Desde la pantalla de Game Over podemos reiniciar el nivel; ir al menú de inicio o salir del juego. Para las siguientes versiones cuando dispongamos de más de un nivel aparecerá la opción de pasar al siguiente nivel.



Pantalla de Game Over

De cara a la próxima entrega necesitaremos centrarnos en el ámbito artístico, visual y sonoro; habrá que realizar los personajes, los enemigos, animaciones, efectos de sonido y música de fondo; al menos de forma general; puesto que ahora mismo no se pueden esquivar los ataques cuerpo a cuerpo porque no hay animación que permita reaccionar al ataque.

También habrá que mejorar el ámbito del combate, y crear diferentes opciones de personaje jugable; al igual que con los enemigos, mediante ScriptableObjects.