

Índice

# Trello:

## Primeramente en el proyecto nos hablaron sobre Trello, y sobre que esta podría ser una buena herramienta a la hora de gestionar las tareas pendientes.

(Adjuntamos el [enlace](https://trello.com/invite/b/AMQHqnQq/ATTIad03bda52e639e5119a5ffd8cb4eb551CFF08768/juego-kodular-fish-invaders) pero pedimos que no toquen las cosas dentro de la web, ya que para poder hacer que el enlace sea una invitación de solo lectura hay que pagar)

No hemos llegado a investigar todas las funcionalidades de trello, ya que simplemente lo hemos utilizado como un tablero en el cual poner posits de trabajos pendientes y para apuntar una guía de lo que había hecho o no cada uno.

Simplemente ponemos cosas como “hacer diseño x” o “programar x” y una vez hecho pasamos al siguiente.

## 

# Kodular:

## Planteamiento del juego

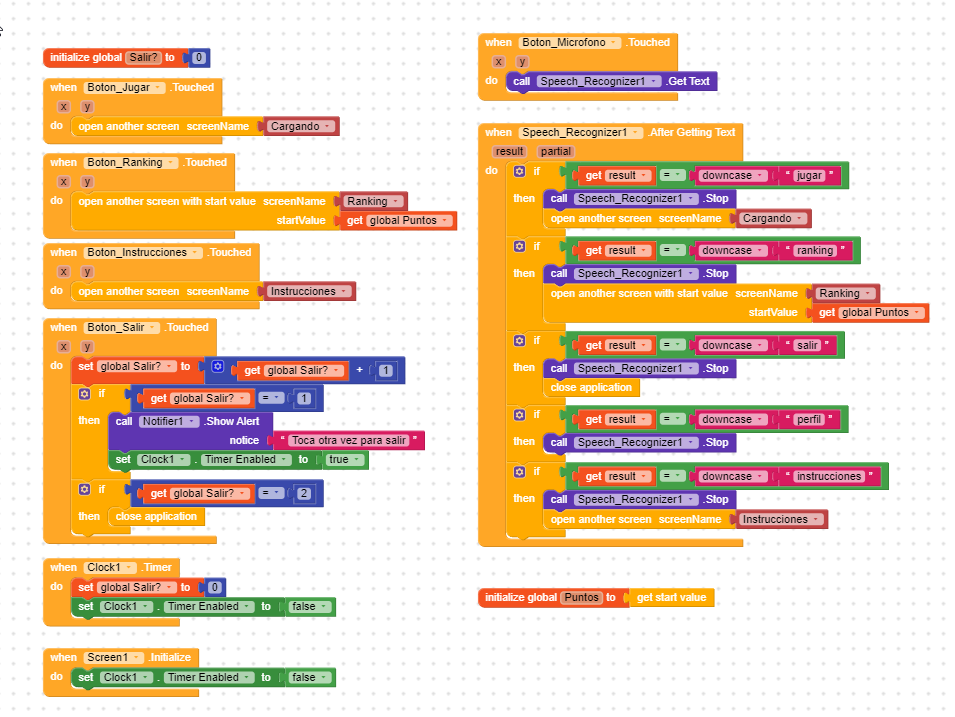
Primero habíamos pensado en hacer algo como un bricks breaker, pero deliberando con los tutores Toni y David hablamos de que sería un trabajo complicado, dadas nuestras capacidades, el hecho de calcular cómo se movería la pelota que rompe los bloques y que también sería complicado implementar las funciones y requerimientos mínimos que nos pedían.

Así que directamente pensamos en otro juego con una mecánica similar, Space Invaders. Una nave que se mueve de lado a lado y destruye alienígenas con un proyectil. Así que pensamos un buen rato y decidimos que podría ser una buena opción.

Decidimos la tarea de cada uno, Alexis como diseñador y Alex como programador y empezamos a desvelar qué debíamos hacer. Por la parte de programación empezamos a ver qué funcionalidades implementar como hacer que tuviera reconocimiento de voz para efectuar acciones, o similares. Y una vez con las ideas despejadas y apuntadas pensamos en la temática del juego. Y decidimos que podría quedar bien un diseño como de debajo del mar. Entonces planeamos cómo deberían ser los botones, la pantalla de carga, los personajes, etc.

Y cada uno se puso con su trabajo.

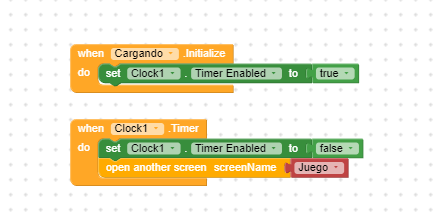
## Primera pantalla explicación bloques:



Botones iniciales: jugar, instrucciones, ver puntuación, etc.

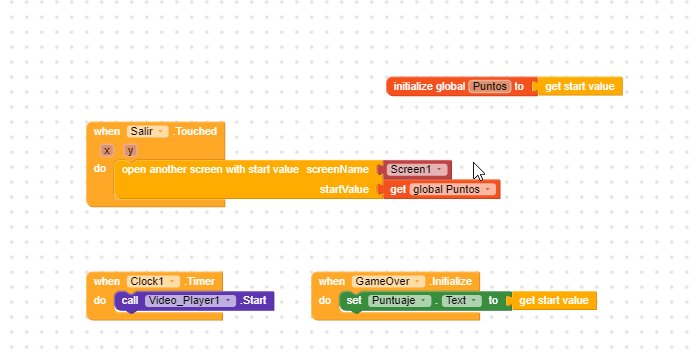
Añadiendo función de speaker para seleccionar los botones mediante el micrófono.

## Pantalla de carga al entrar al juego



Al probar el juego consideramos que en el momento de iniciar la partida todo iba como muy rápido, ya que directamente al darle al botón “jugar” el personaje empezaba a disparar. Entonces pusimos una “pantalla de carga” con un contador para que se hiciera un poco más lento.

## Game Over

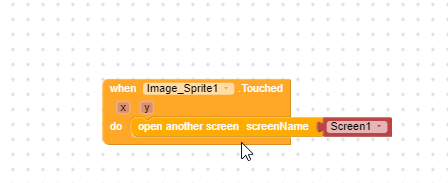


En la pantalla de Game Over se reproduce un video cuando la pantalla se inicializa.

Hay un botón de salir que nos redirige a la pantalla del menú principal.

Hay un “Label” con el nombre de “Puntaje” que convertimos su texto a los puntos obtenidos.

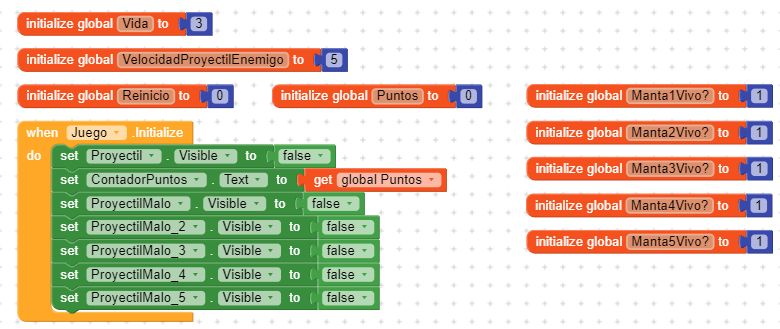
## Instrucciones



En la pantalla de instrucciones, aparece explicado brevemente como funciona el juego añadiendo un botón para volver a la pantalla principal. en el bloque se puede ver que básicamente cuando pulsas el botón vuelves atrás.

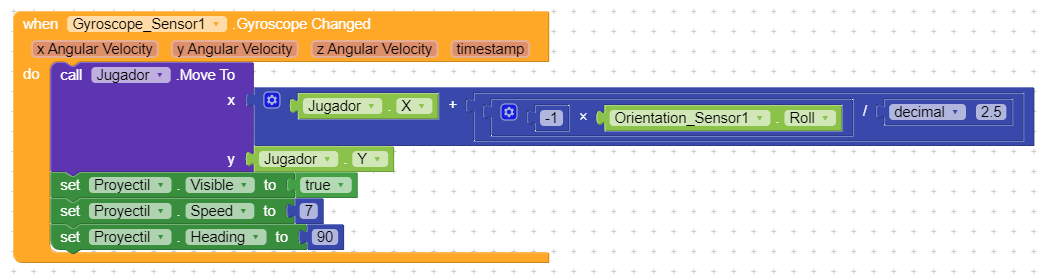
## Juego

Al iniciar el juego



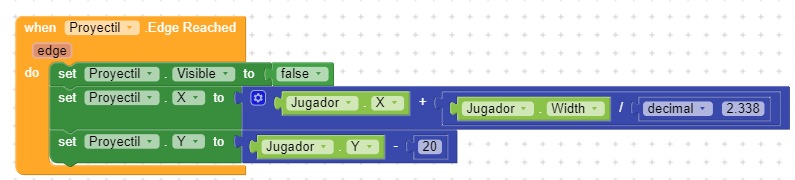
Al iniciar el juego se ocultan todos los proyectiles y se declaran todas las variables que necesitamos para comprobar las vidas, el escalado de la velocidad de los proyectiles enemigos, si están vivos o no y el reinicio de los enemigos.

Movimiento del jugador



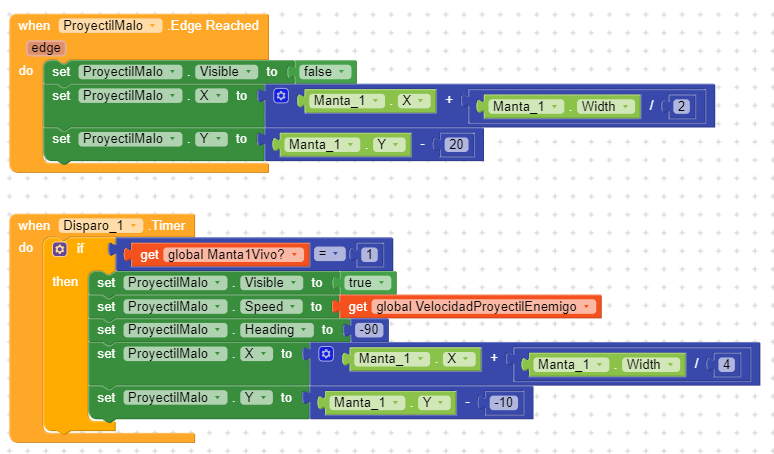
Como el movimiento del jugador es por el giroscopio, hacemos que cuando se detecte un cambio en el giroscopio la posición X del jugador se mueva. También controla el disparo del jugador.

Disparo



Cuando el proyectil impacta contra el borde de la pantalla, lo oculta y lo envía de nuevo hacia la posición del jugador.

Disparo de los enemigos



Hay 5 bloques como estos para controlar de forma independiente cada proyectil que se dispara.

Hay un contador que cada X segundos comprueba si la mantarraya que dispara está viva o no.

En el caso de que siga viva se muestra el proyectil y se dispara desde la posición de la Manta correspondiente.

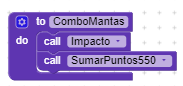
Cuando el proyectil impacta contra el borde de la pantalla desaparece y se mueve hacia la posición de la manta correspondiente.

Comprobación de colisiones del proyectil



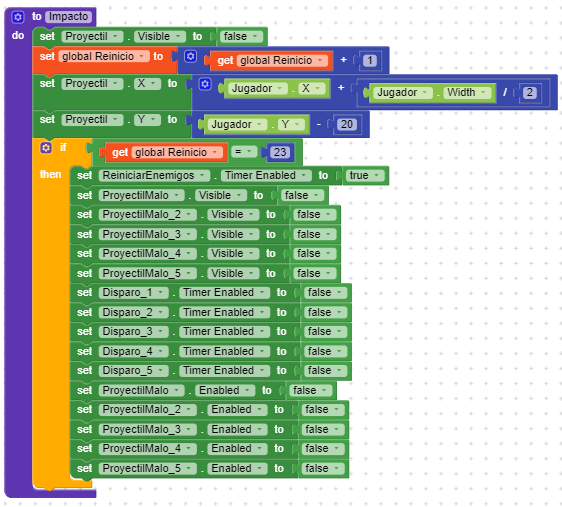
Aquí comprobamos si el proyectil impacta contra algo.

Hay una cadena muy larga de ifs que comprueban cada enemigo para saber con cual ha impactado, una vez lo encuentra hace desaparecer al enemigo y llama a la función Combo(Tipo de enemigo.



Esta función se encarga de llamar a dos funciones distintas que son:

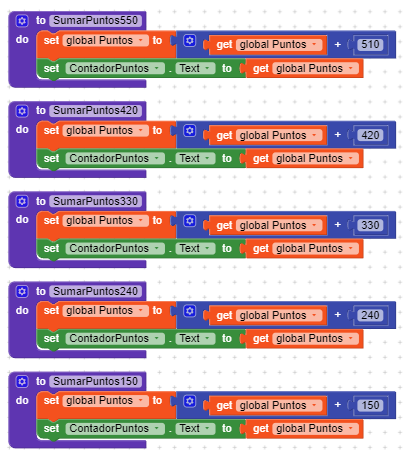
Impacto



Esta función lo que hace es ocultar el proyectil y sumar 1 a una variable que se llama “Reinicio” que se encargará de comprobar cuando los enemigos volverán a aparecer cuando mueran todos y hacer que inicie un contador que se encargará de hacer que los enemigos se reinicien pasados unos segundos.

También hace que el proyectil vuelva a dispararse desde la posición del jugador cuando impacta contra un enemigo.

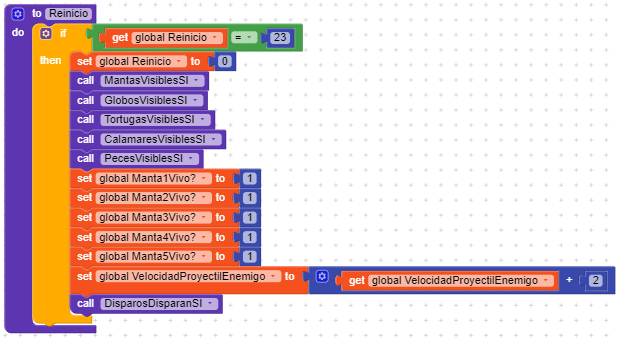
Sumar puntos:



Hay 5 variantes porque cada tipo de enemigo da una cantidad de puntos distinta

Lo único que hace esta función es sumar la cantidad correspondiente de puntos a la variable “Puntos” y actualizar el contador.

Reinicio de los enemigos

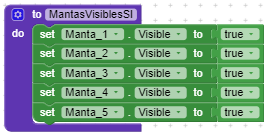


Esta función comprueba que cuando el contador de enemigos muertos sea igual a 23 vuelva a poner la variable a 0 y llame a la función “(Enemigo)VisiblesSI” y vuelva a poner en 1 la variable que comprueba si las mantas están vivas para que puedan salir los proyectiles enemigos.

También se aumenta la velocidad de los proyectiles enemigos cada vez que se reinicia.

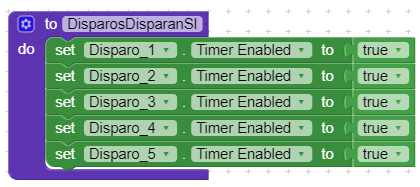
También se llama a la función “DisparosDisparanSI”

Función “(Enemigo)VisiblesSI”



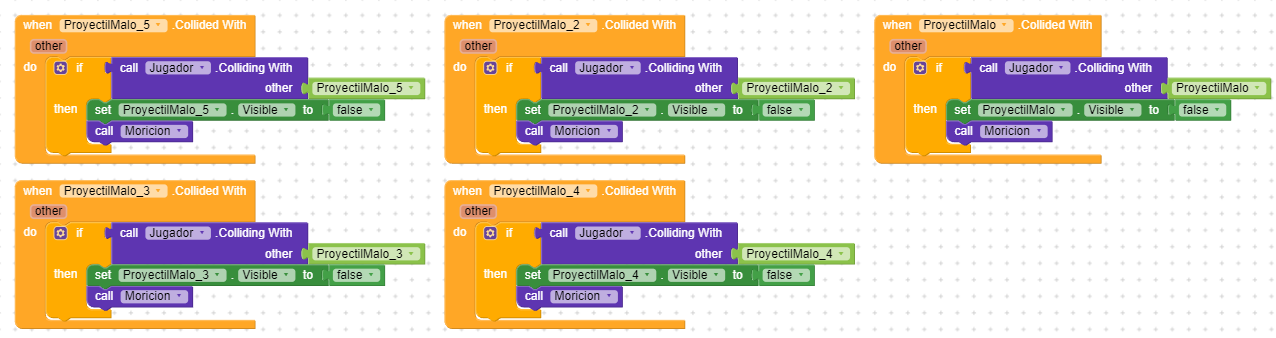
Hay 5 funciones que corresponden a cada tipo de enemigo que lo único que hace es hacer que todos los enemigos de ese tipo sean visibles.

Función “DisparosDisparanSI”



Lo único que hace esta función es volver a habilitar los relojes que controlan cada disparo.

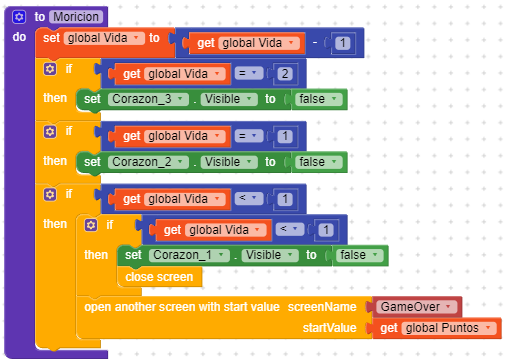
Detectar colision con el jugador



Hay 5 bloques de código independientes para cada proyectil enemigo que lo que hace es comprobar cuando un proyectil malo impacta contra algo es el jugador o no. En caso afirmativo el proyectil se oculta y llama a la función “Moricion”.

Estos bloques se hicieron porque los proyectiles enemigos mataban enemigos, así que tuvimos que hacer esto para que solo pudiesen impactar contra el jugador.

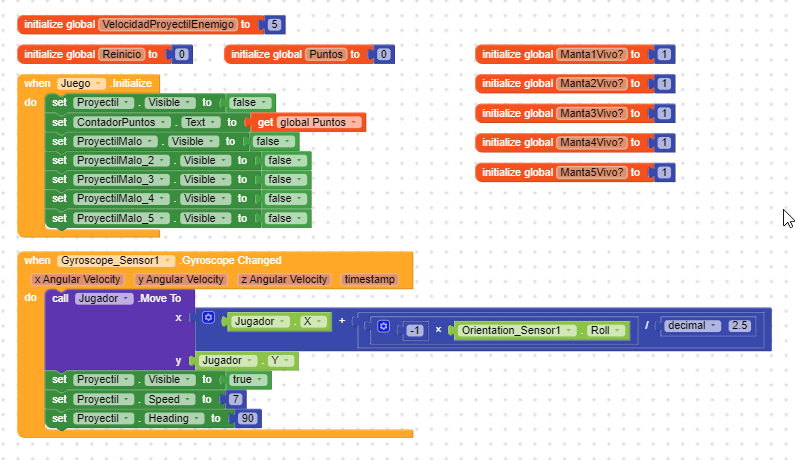
Función “Moricion”

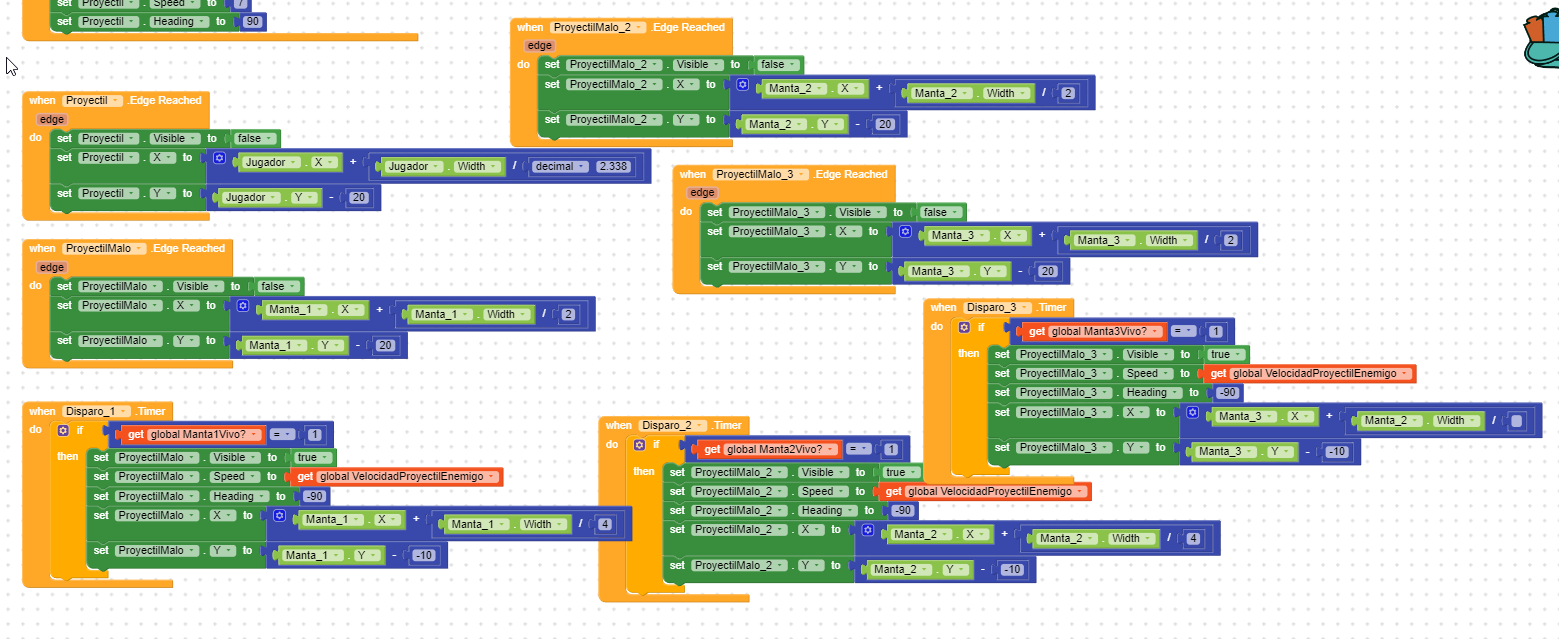


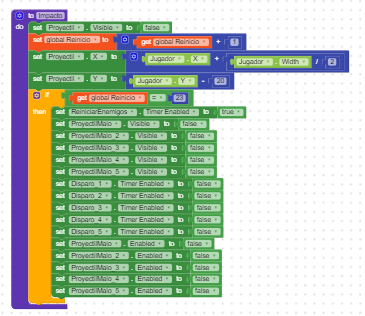
Lo que hace esta función es encargarse de gestionar las vidas del jugador y acabar la partida.

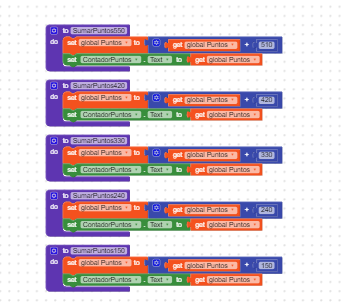
Cuando se llama a esta función se resta en 1 la vida del jugador. Luego hay unos ifs que lo que hacen es ocultar el icono del corazón para señalizar que se ha perdido esa vida.

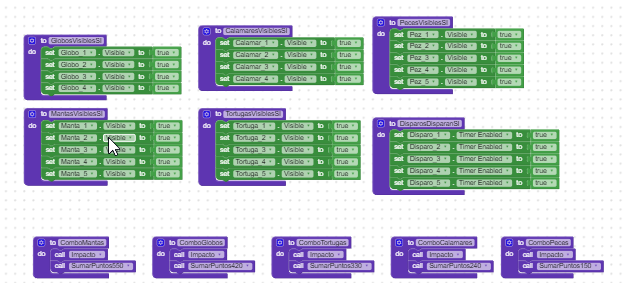
Una vez la vida llega a 0 se cierra la pantalla del juego y se cambia a la pantalla de GameOver guardando el valor de los puntos obtenidos.

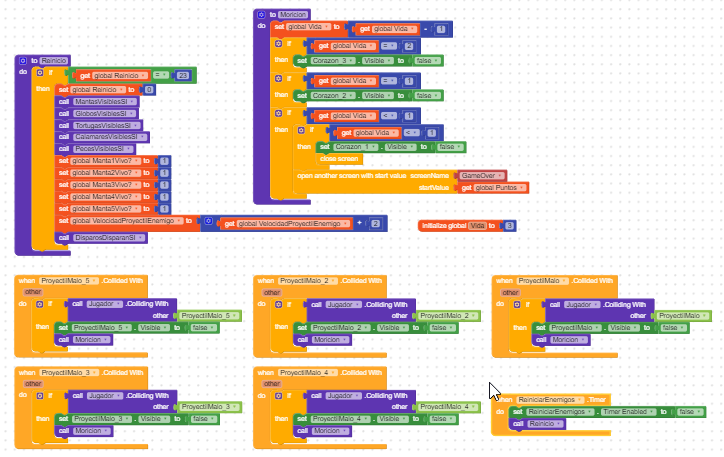






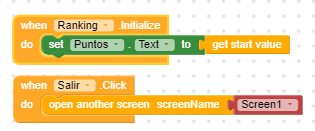






## 

## Ranking



Dado a que no pudimos llegar a hacer la parte de multiusuario, lo único que cambia en este apartado es que en vez de mostrar un ranking de todos los que han jugado es que muestra la puntuación conseguida en la partida anterior.

# Diseños:

## Entrega:

En la misma carpeta de entrega hay una carpeta que contiene los diseños de todas las cosas que hay implementadas en el juego. Desde los personajes hasta el logo y los fondos.

Todos han sido hechos originalmente por nosotros en una página web para crear dibujos en pixel art, escogimos que fuera en pixel art porque consideramos que le daría un mejor ambiente dadas sus funcionalidades.

Os dejamos aquí un link a la web utilizada y el diseño del personaje principal, un camarón pistola.

[LINK](https://www.pixilart.com/alexisvega)

