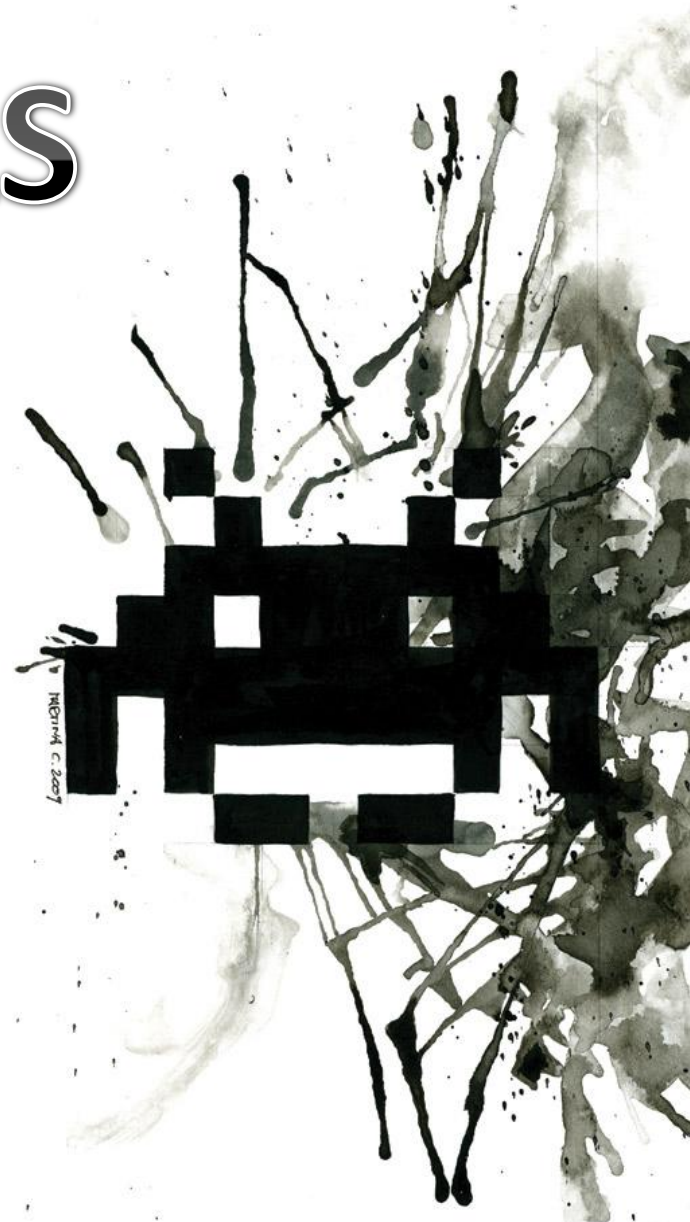


# SPACE INVADERS

Antonio Luis Moreno Castro.

Daniel Gómez Martínez.

Ismael Núñez Carrión.



# ÍNDICE

1. Introducción
2. Instrucciones
3. Objetivos propuestos
4. Bloques. Divisor frecuencia genérico
5. Bloques. Dibuja genérico
6. Movimiento (bala, alien, nave, bala alien)
7. Bloques de memoria
8. Diagrama de bloques de los Tops
9. Máquina de estados
10. Problemas encontrados
11. Simulaciones y calendario

# 1.Introducción

Se dispone a realizar el juego Space Invaders en la placa Spartan III con un kit de programación VHDL.

Space Invaders es un videojuego tipo arcade lanzado al mercado en 1978. Este videojuego muy relevante debido a que es uno de los precursores de los juegos de "shoot 'em up". Tiene como objetivo eliminar oleadas alienígenas disparando un cañón de láser (porque es verde).



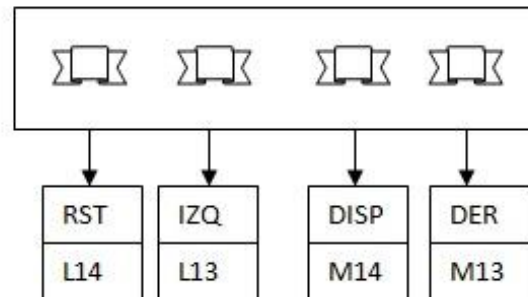
## 2.Instrucciones

El juego consta de una nave que se moverá de izquierda a derecha y disparará un proyectil con los botones indicados en la imagen.

Consta también de una matriz de aliens a los que hay que matar mediante el impacto del proyectil.

El objetivo del juego es destruir todos los aliens con tu nave.

El juego acaba un vez que todos los aliens sean eliminados o la nave sea destruida por los aliens mediante el choque directo o el disparo que generan.



# 3.Objetivos propuestos

Los objetivos principales fueron:

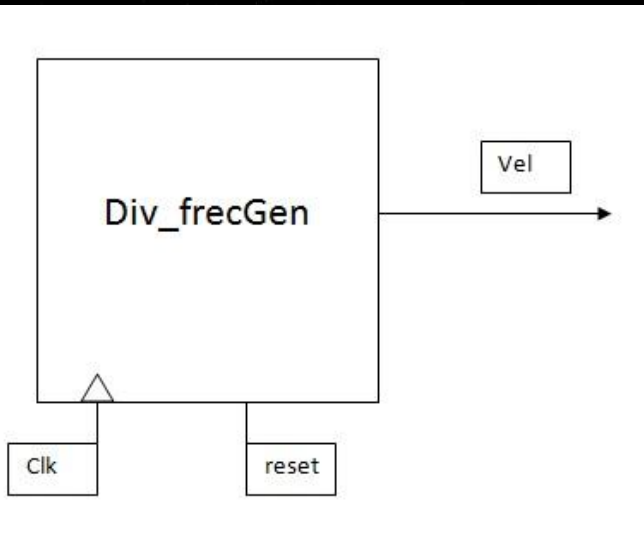
- Al menos que un marciano se comporte correctamente.
- Poder controlar la nave y tenga vidas.
- Conseguir la posibilidad de disparar y matar marcianos.

Los objetivos adicionales :

- Poner más marcianos.
- Una pantalla de reintento.
- Caso extremo: Te devuelva la bala en función del reloj.

Se han cumplido todos los objetivos principales y adicionales excepto la pantalla de reintento, y además, se ha añadido:

## 4. Bloques. Divisor de frecuencia genérico



Es un divisor de frecuencia genérico que utilizaremos para los enable de la mayoría de bloques.

Su composición es la misma que se vio en las

Prácticas: Cada "X" pulsos de reloj generará una salida llamada "vel".



# 5. Bloques. Dibuja genérico

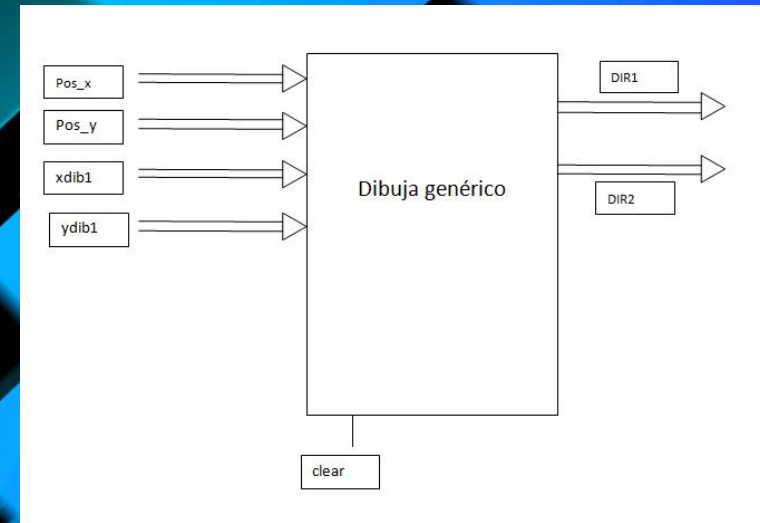
Todos los vectores utilizados son de tamaño 10.

Su funcionamiento es el siguiente:

Recibe las posiciones x e y por donde va pintando el driver vga.

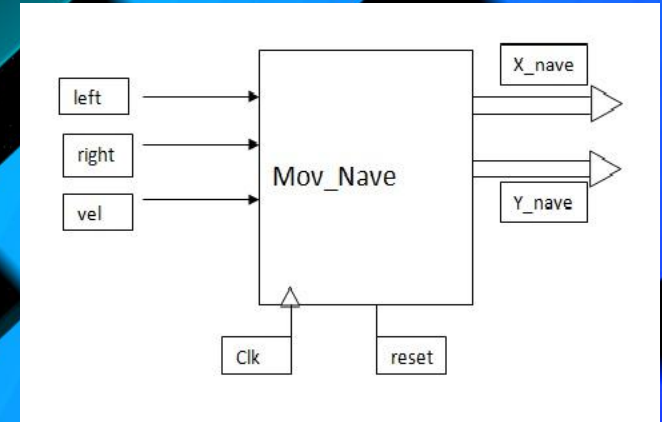
Compara con xdib e ydib y genera xrel e yrel que son posiciones relativas en el sprite.

A partir de éstas últimas señales se genera DIR1 y DIR2 que corresponden a las señales xrel e yrel, estos dos vectores son de tamaño 5 que entran en la memoria como vector DIR de tamaño 10.

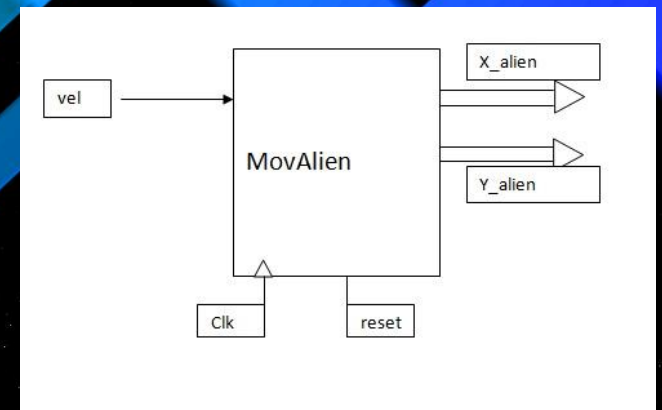


# 6. Movimiento (bala, alien, nave, bala alien)

El bloque nave recibe las señales de los botones left y right, a partir de estas dos calcula la posición de la nave. Vel es la señal que viene del divisor de frecuencia (enable).



El bloque movimiento alien solo recibe la entrada vel (enable) y automáticamente genera la secuencia de movimiento del alien.

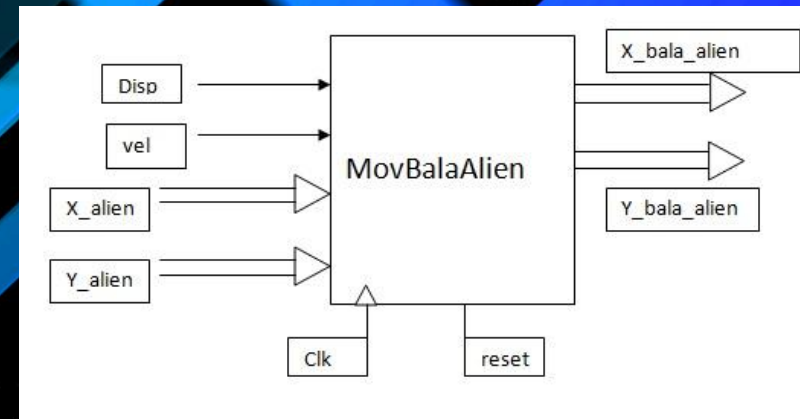
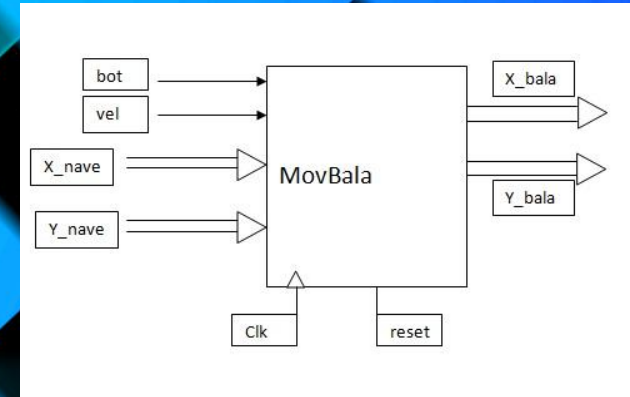




## 6. Movimiento (bala, alien, nave, bala alien)

Éstos bloques reciben las posiciones de nave o alien y una señal de disparo con su correspondiente nombre.

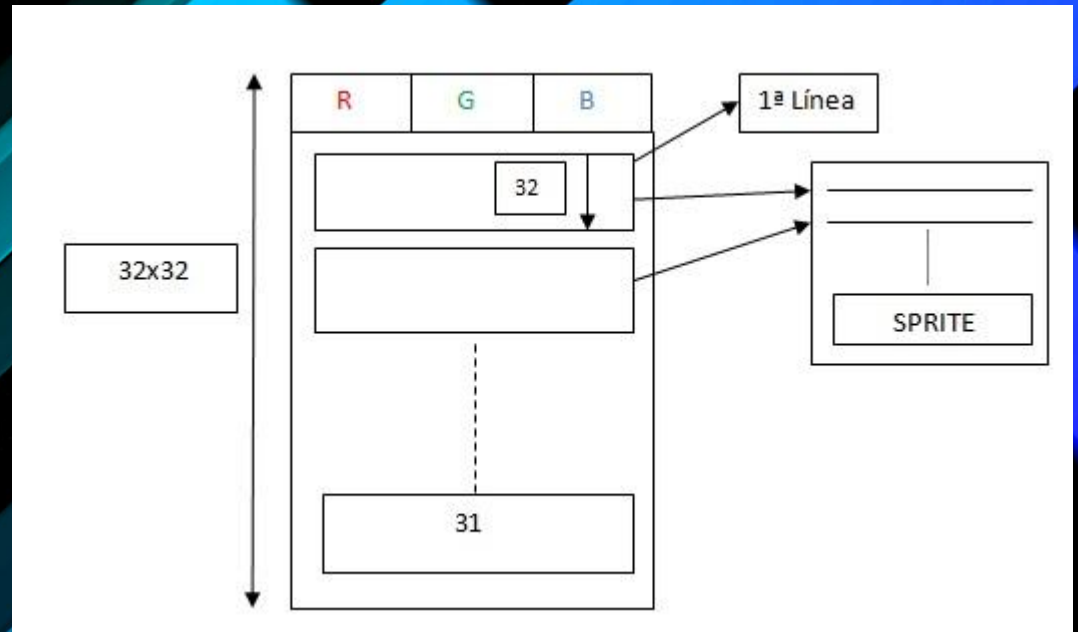
Se mueve siempre en línea recta.



# 7. Bloques de memoria

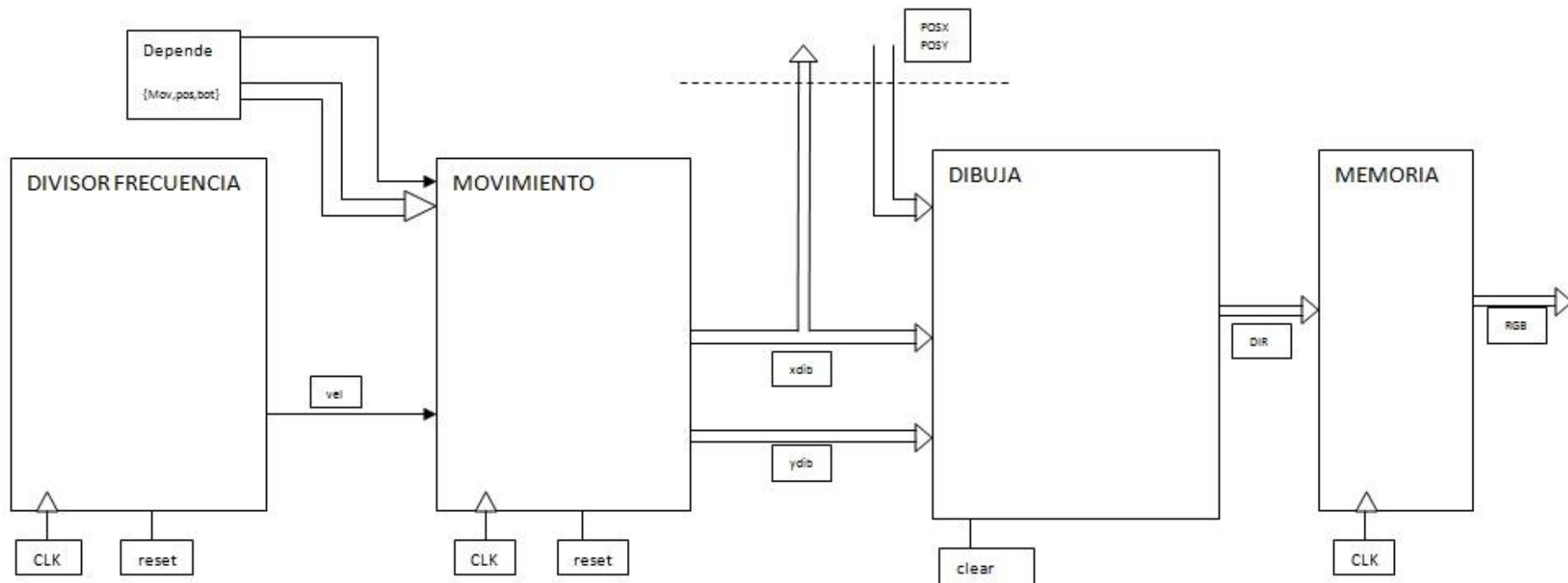
Usamos los siguientes bloques de memoria:

- Nave
- Alien
- Alien2
- Bala nave
- Bala alien
- Ganaste
- Perdiste
- Inicio



# 8. Diagrama de bloques de los tops

Aquí se presenta el diagrama de bloques genérico para el top de nave, alien, bala alien, bala.





# 8. Diagrama de bloques de los tops

Presentamos a continuación el diagrama de bloque de vidas.

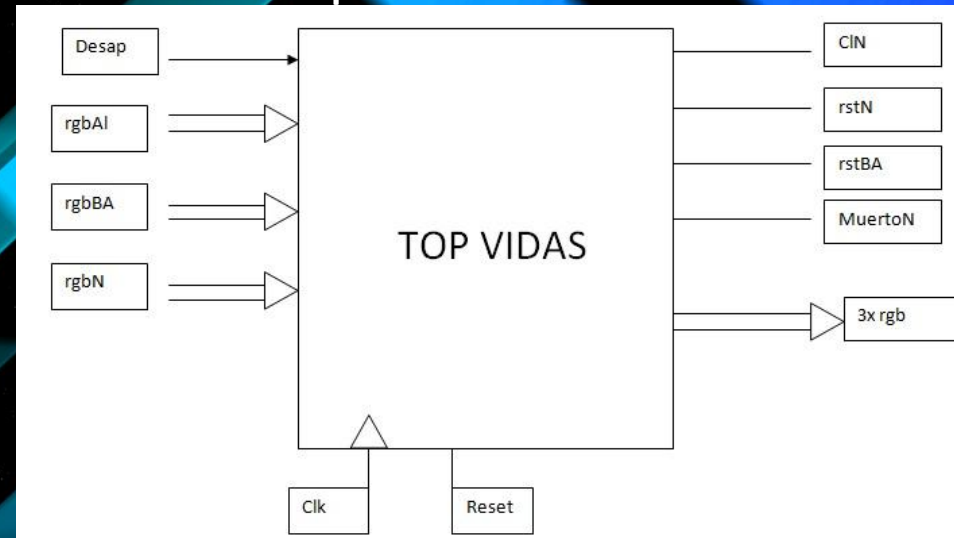
Desap es la señal que utilizaremos como clear para que no se dibujen las vidas.

Las tres señales de rgb entrantes se utilizan para identificar cuando los enemigos han impactado la nave.

Genera el clear y reset de la nave.

Genera la señal bala alien cuando impacta la nave.

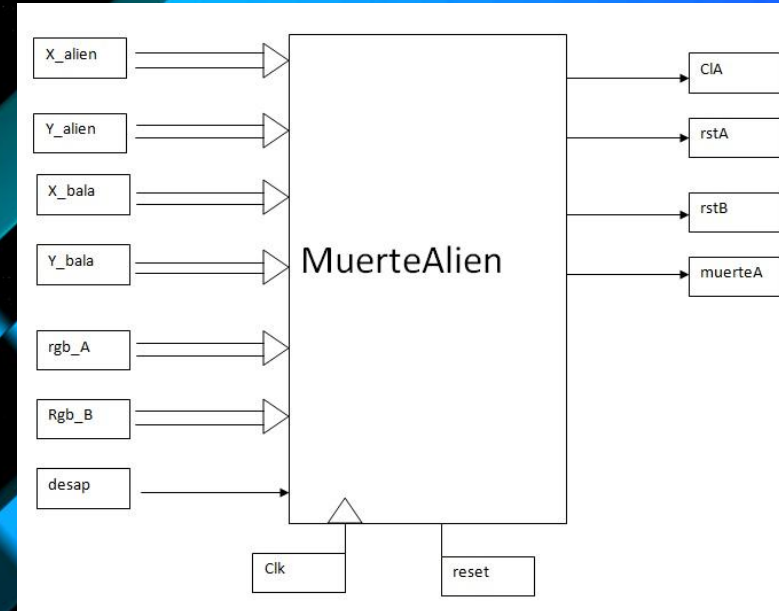
Genera la señal muerto y el rgb para cada corazón (3).



# 8. Diagrama de bloques de los tops

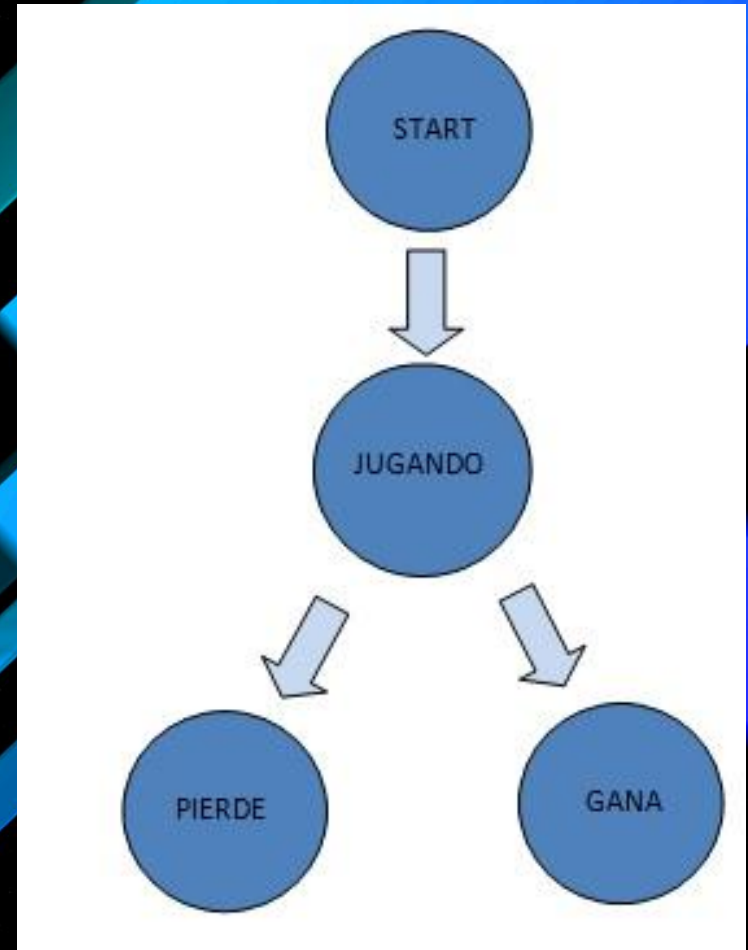
Este bloque es MuereAlien y se encarga de detectar cuando el alien recibe un impacto procedente de la bala comparando los rgb de la bala y del alien.

En caso de tener una matriz de aliens se compara las posiciones del alien y de la bala para saber en que alien ha impactado.



# 9. Máquina de estados

La máquina de estados empieza en el estado start que es una pantalla de inicio prediseñada. Al pulsar el botón (1er interruptor) pasa al estado jugando que es el juego en sí. Después de ese estado si se gana se verá una pantalla de victoria y si se pierde se verá una pantalla de derrota. Para comenzar de nuevo el juego es necesario pulsar el botón reset.





# 10.Problemas encontrados

- Nos quedamos sin memoria al pintar la pantalla inicio y las dos pantallas del final.
- Terminando ya el proyecto teníamos muchos errores que no éramos capaces de arreglar o determinar el origen, por lo que decidimos rehacer el proyecto bloque a bloque, reestructurando el contenido y optimizando la variable, aunque en esencia era el mismo código.
- El alien se pintaba en el borde derecho de la pantalla dejando una estela tras de sí, se solucionó disminuyendo la velocidad al pintarlo.
- Problema con las vidas: no se pintaban debido a problemas de sincronismo. Gran retraso en la elaboración del proyecto.

# 11.Simulaciones y calendario

1. Una vez definida la nave y su movimiento se prueba en la placa y se obtiene el resultado esperado: se mueve y se pinta correctamente.
2. Pintamos y definimos el movimiento del alien. Se prueba en la placa y funciona correctamente.
3. Implementamos la bala y comprobamos que funcionaba correctamente aunque no mataba al alien.
4. Arreglamos el problema con la muerte del alien.
5. Elaboramos tres vidas y comprobamos en la placa, no se pintan.
6. Arreglamos el problema con las vidas.
7. Elaboramos una matriz de marcianos y vemos que en la placa funciona correctamente.

En resumen el calendario propuesto se cumple hasta el momento en el que nos topamos con los errores que solucionamos rehaciendo el trabajo y a partir de ahí nos retrasamos mucho con respecto al calendario inicial.