



# CrabPang

<https://github.com/juanantc/CrabPang.gmx>

Juan Antonio Moraleda Ocón

76628694P

## Herramienta de desarrollo utilizada

Para el desarrollo de este videojuego se ha utilizado GameMaker: Studio. Es un motor muy potente para el desarrollo de videojuegos 2D y que se puede obtener de forma gratuita.

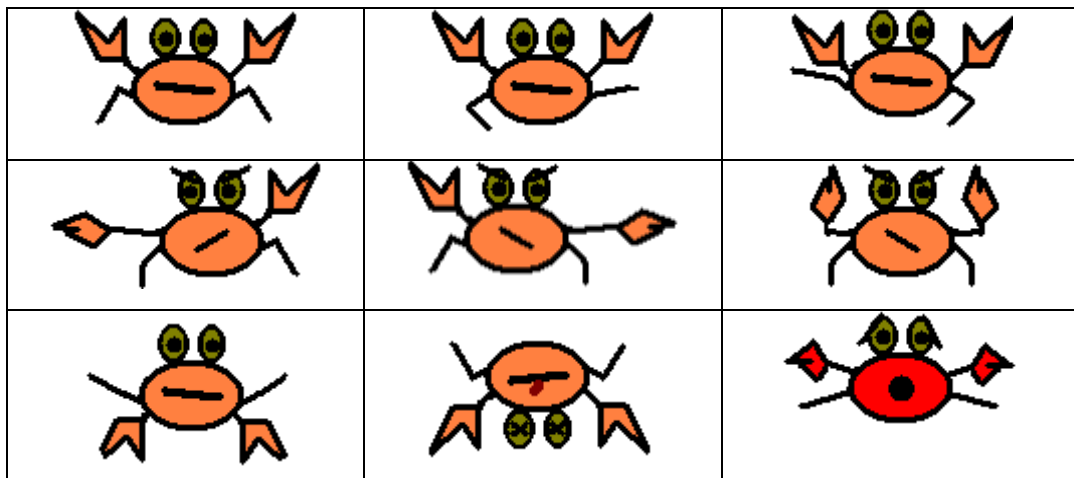
Esta herramienta de desarrollo está basada en un lenguaje de programación (Game Make Language) y un kit de desarrollo de software (SDK).

GML es un lenguaje de programación interpretado desarrollado para usarse como complemento de Game Maker. Su sintaxis es similar a la de C++, aunque también se ve influenciado por la sintaxis de otros lenguajes como Pascal.

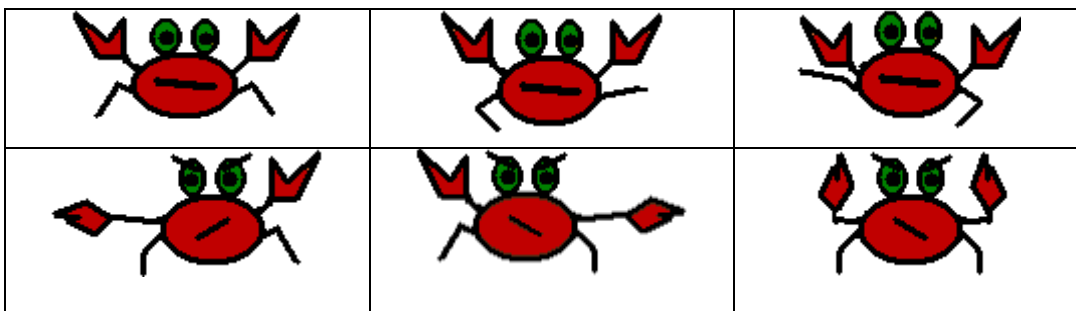
## Sprites

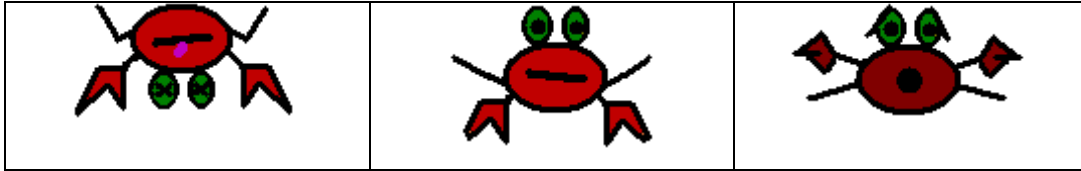
El primer paso en el desarrollo del videojuego ha sido la creación de los sprites (permiten crear animaciones). Los principales son los siguientes:

### Personaje:

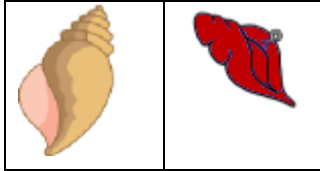


### Enemigo:





Otros:



Cada uno de los sprites del personaje permiten construir animaciones que posteriormente estarán asociadas a los eventos correspondientes.

Aunque los sprites del enemigo no se han usado en esta versión del videojuego, están creados para utilizarlos en una versión futura de éste.

Los dos últimos sprites representarán a los objetos que serán objetivo de los disparos de nuestro personaje.

## **Mecánica de juego**

Primer nivel:

El jugador deberá conseguir la mayor puntuación posible en el tiempo establecido moviendo el personaje y disparando a los objetos que caen. Si el jugador consigue destruir un objeto antes de que llegue al suelo obtendrá tres puntos. El objeto permanecerá unos segundos en el suelo y si en ese intervalo de tiempo no recibe ningún disparo desaparecerá. En caso de recibir un disparo solo sumará un punto.

Si un objeto en movimiento colisiona con el personaje, se restará un punto, en caso contrario restará tres puntos.

Se disponen de tres disparos consecutivos. Hasta que un disparo no colisione con el objeto a destruir o una pared, no volverá a estar disponible (Evitar ráfagas continuas)

Segundo nivel:

Se trata de una variante del nivel anterior en la que al disparar a un objeto en movimiento, éste se divide en dos objetos más pequeños. El primer objeto suma tres puntos, mientras que los dos siguientes valen un punto cada uno.

## **Controles del personaje**

- Desplazamiento horizontal:

- Desplazamiento hacia la izquierda: flecha izquierda
- Desplazamiento hacia la derecha: flecha derecha.
- Disparo:
  - A: disparo a la izquierda.
  - W: disparo arriba.
  - D: disparo a la derecha.

## Inteligencia artificial

Se ha desarrollado un algoritmo para la generación de los objetos que caen.

El código es el siguiente:

```

1 min_x = 45
2 max_x = room_width - 45
3
4 i = obj_personaje.x - (score+10)*1.5
5 d = obj_personaje.x + (score+10)*1.5
6
7 if i < min_x
8     i = min_x
9
10 if d > max_x
11     d = max_x
12
13 x = random_range(i,d)
14
15 if room == nivel1
16     instance_create (x,0,obj_caracola1)
17 else if room == nivel2
18     instance_create (x,0,obj_caracola1_nivel2)
19
20 //show_debug_message("i: " + string(i) + ", " + "d: " + string(d))

```

Este código genera nuevas instancias de los objetos a disparar de forma aleatoria dentro de un rango que viene determinado por la posición del personaje y la puntuación. Así, cuando el jugador tiene una puntuación baja, los objetos caerán cerca del personaje y a medida que aumente su puntuación, el rango donde pueden caer irá aumentando. De esta forma cada vez será más difícil conseguir puntos en la partida.

## Documentación consultada

<http://docs.yoyogames.com/>

<http://gamedesign.wikidot.com/gamemaker:game-maker>