

## Guía del Shooter

### Samuel paredes posada

#### 1 ¿Qué es una variable en game maker?

Una variable es algo que se utiliza para almacenar un valor para su uso en una o más operaciones o funciones.

#### 2 ¿Qué es el ámbito (scope) de una variable

**El Ámbito Local:** Si una variable se declara con la palabra clave var entonces es sólo visible dentro de este script o evento y es en el ámbito local.

**Alcance Global :** Si una variable es declarada utilizando global-var o se conoce con el prefijo global , entonces es en el ámbito global y es visible desde todas las áreas de código

**Instancia Alcance** - si una variable no ha sido declarada con var o globular y no se hace referencia al uso global o cualquier otro prefijo, entonces es una variable de instancia y se encuentra en el ámbito de instancia.

**Alcance explícita** -Si una variable se denomina con un prefijo con el DOT sintaxis es decir obj.varname , entonces esa variable está utilizando alcance explícito, el propio prefijo es el contexto en el que la variable se hace referencia desde .

#### 3 ¿Cuáles son los tipos de variables en game maker?

**Instancia:** La categoría más común. Las variables son definidas dentro de la instancia. Son únicas y propias de su instancia y se les puede usar en cualquier evento y cualquier función dentro de esa instancia.

**Local:** Estas variables se declaran mediante la función "var". Una variable local sólo es válida dentro del evento o script en el cual se crea.

En este caso GameMaker:Studio creará la variable, la usará el tiempo que dure el evento y luego se "olvidará" de ella, lo que causará que se obtenga el error "unknown variable" si se intenta usarla posteriormente.

**Global:** Una variable global pertenece a todo el entorno de juego, y no a una instancia en específico (a pesar que se declare dentro de una instancia). En un principio, tiene que declararse como global, pero después de esto, cualquier instancia puede leer o cambiar su valor. El valor de la variable siempre reflejará la última operación de la que fue objeto, sin importar qué instancia desarrolló la operación.

**Variables internas:** Son variables especiales inherentes a los objetos y habitaciones en el entorno. Están ahí desde que una habitación u objeto es creado. Pueden tener alcance de instancia o global, pero nunca local. Hay una gran cantidad de estas variables y cada una tiene usos muy específicos, se les describe con detalle a lo largo del manual en las secciones correspondientes.

**4** ¿Qué tipo de variable utilizaría si debe acceder a información desde varios eventos de una misma instancia?

Se utilizaría la variable global que nos permite realizar dicha función.

**5** ¿Qué tipo de variable utilizaría si sólo necesita acceder a información en un evento?

Se utilizaría la variable local que solo es permitida dentro de un evento

**6** ¿Qué tipo de variable utilizaría si debe acceder a información desde varias instancias?

Se debe utilizar la variable global que es un tipo de variable que una vez declarada no pertenece a ninguna instancia, pero puede ser leída por todas ellas.

**7** Indique 5 variables globales en game maker:

5 variables globales que se usaran en la construcción de todo el

aplicativo.

8: Indique 8 variables de instancia en game maker

• Food=15; • Munición=8; • Vidas=3; • Armadura=1; • Armas=3;

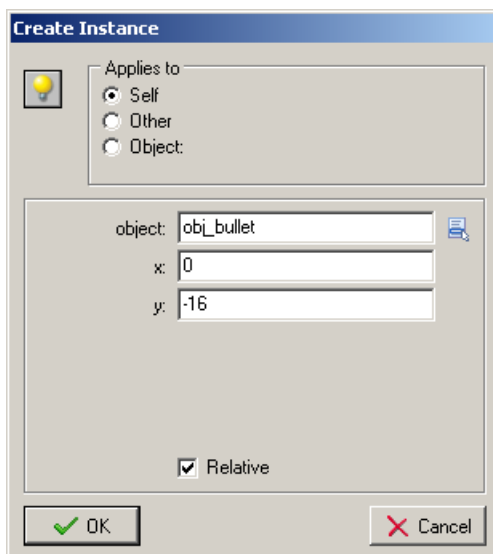
9 :Considere dos variables en game maker: Vidas y vidas. Son iguales estas dos variables?

No son iguales porque el código del gamemaker distingue entre mayúsculas y minúsculas y las

toma como variables independientes que funciones totalmente diferentes

10 Considere la siguiente figura donde Self se refiere al objeto obj\_myplane.

- €Globalvar food;
- €Globalvar bullet;
- €Globalvar Health;
- €Globalvar Score;
- €Globalvar Live;



**10.1** ¿En qué posición de la pantalla se crea el objeto obj\_bullet?

Se crea en la posición 0,-16 es decir tomando en cuenta el plano seria en el centro X desplazando 16 espacios hacia abajo como lo indica el (-).

**10.2** ¿Qué indica la opción Relative en este caso?

El valor dado se añade al valor actual de la variable.

**11.** ¿Por qué las islas del escenario tienen un Depth de 10000?

Para asegurarse que los objetos queden por encima de las islas

**12** Explique qué hace la siguiente función random(room\_width).

Esta función creará un valor aleatorio entre 0 y el ancho de la habitación

**13** ¿Por qué las islas siempre aparecen en el mismo lugar cada nuevo juego, sabiendo que se está empleando la función random?

**14** ¿Qué hace la función randomize?

Hace que las cosas sean calculadas de manera aleatoria

**15** ¿Por qué no debería utilizarse randomize si se está depurando o probando el juego?

**16** ¿En qué parte del programa debe llamarse randomize?

Porque en la función random utilizamos -65 para la posición Y para asegurarnos de que la isla se inicia completamente por encima de la habitación. Debido a su velocidad vertical esta se moverá de nuevo a la vista.

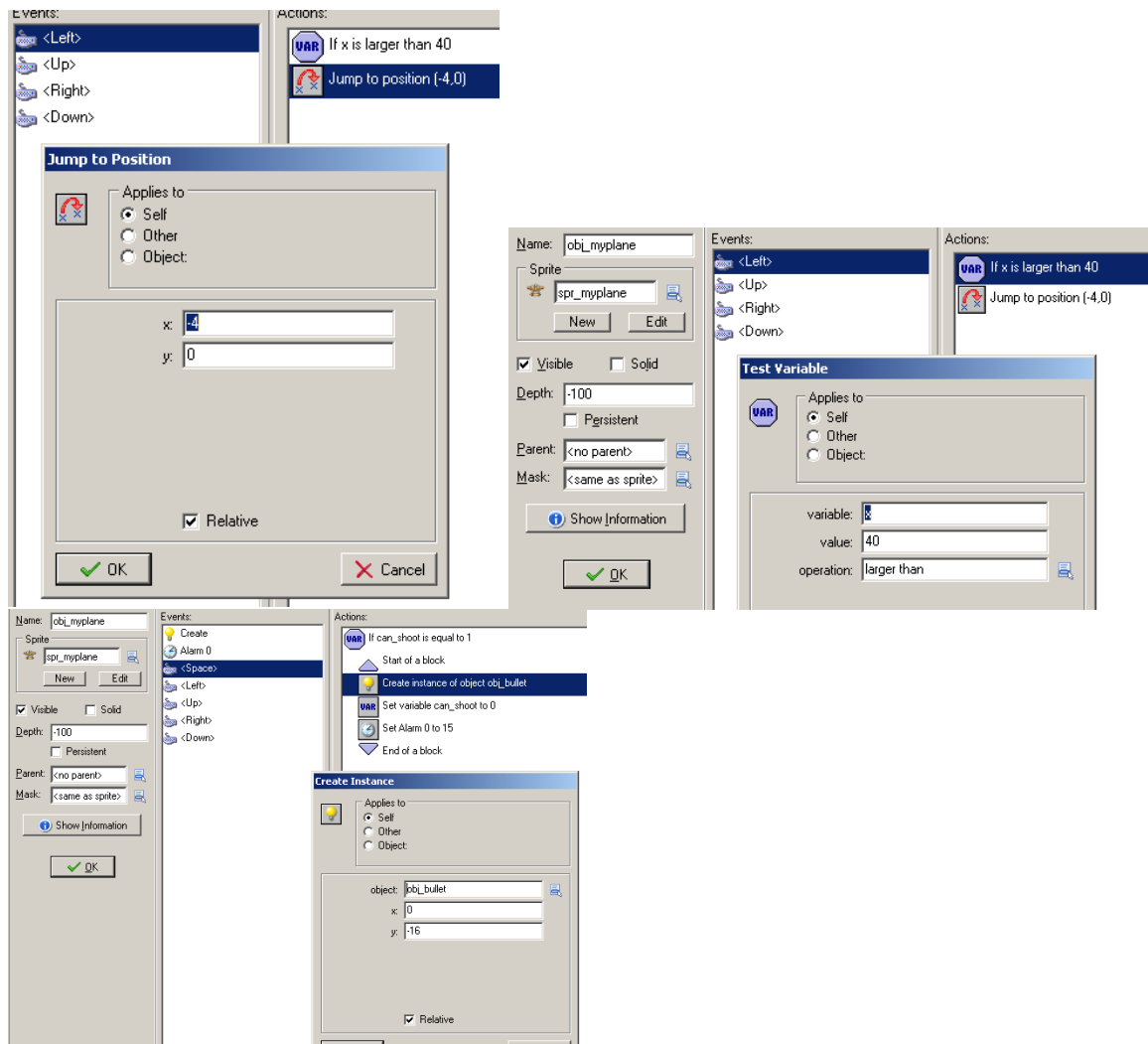
**17** Explique que hace el siguiente código:

Este código nos indica las posiciones en las cuales se va a desplazar el avión y nos da una condición que si la variable X es mayor que 40 el avión salte a la posición -4,0.

**18.** Al crear el sprite del avión principal se definió el origen en la

posición 32,32. Cuando se crea la instancia de la bala, indique la posición exacta donde se coloca en el room

La bala se coloca en toda la parte superior del avión (0,-16) con el fin de darle un estilo más real en el momento de dispararla.



19. Explique por qué razón en el evento step de la bala que lanza el avión principal, es necesario ejecutar el siguiente código:

Lo que hacemos con este código es decir que si la bala supera esta posición será eliminada así consumiremos menos memoria porque se eliminara la bala al no ser vista de nuevo.

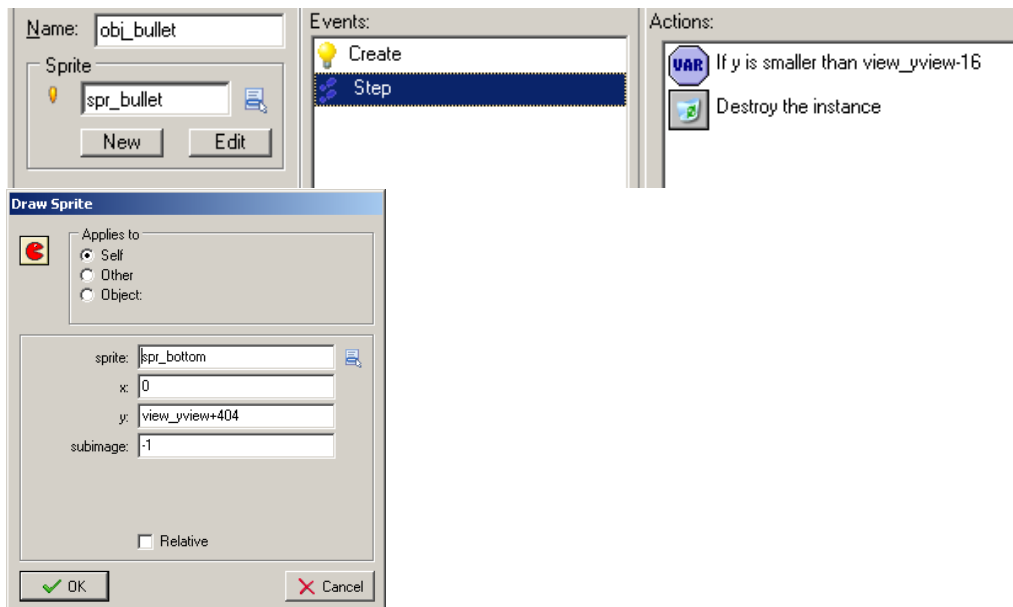
20. Explique cómo se implementa la funcionalidad que permite al avión principal disparar sólo dos balas por segundo.

La bala debe ser disparada cuando el usuario pulsa la tecla de espacio. Pero no queremos demasiadas balas en el mismo momento. Sólo permitimos al jugador disparar dos balas por segundo, es decir, una bala cada 15 pasos. Para ello utilizamos una variable `can_shoot`. Establecemos esta variable a 1, lo que indica que, efectivamente, podemos disparar una bala. Y una alarma que nos permita volver a disparar cada 15 pasos. Permitiendo que la variable `can_shoot` vuelva a retomar su valor en 1 permitiendo 2 balas por seg.

**21.** Explique cada uno de los pasos que ocurren desde que un enemigo choca el avión principal hasta que el avión aparece de nuevo.

El enemigo choca con el avión principal y reduce la barra de la salud cuando varios enemigos chocan con el avión principal este se queda sin salud y muere restándole 1 a sus vidas en el momento en que todas las vidas se terminen el juego vuelve a comenzar con todas las vidas.

**22.** ¿Qué hace el objeto `controller_enemy`? R:/ Este objeto controlara la creación de enemigos dentro del juego. **23.** ¿Cuál es la función del evento `Alarm`? R:/ Nos permita volver a crear enemigos dentro del juego. **24.** En la siguiente figura ¿Qué significa el parámetro `subimage`?



Esto significa que la subimagen actual se dibuja, Como sólo hay una subimagen en el sprite, realmente no nos preocupamos por esto, pero si un sprite consta de múltiples subimágenes se puede indicar aquí la subimagen que desee ver.

**25.** Explique cómo funciona el código de la bala del enemigo 3 que se muestra a continuación:

Utilizamos esta para comprobar si el plano principal está disponible y, si es así, dirigir la bala hacia el avión. De lo contrario, la bala va a ir hacia abajo.

**26.** ¿Qué hace el evento Outside Room? ¿Qué pasa si no se coloca?

El evento destruye el objeto, Si no se coloca tendríamos las balas recorriendo todo el mapa y

es precisamente lo que no queremos.

**27.** ¿Qué es una vista o View?

Las vistas son una de las más importantes de las propiedades de la pantalla que se pueden manipular con GameMaker. Son, básicamente, pequeñas ventanas en su mundo de juego que le permiten mostrar por partes al jugador dentro de una sala.

**28.** Cuando se crea la vista, ¿Por qué se coloca la velocidad del background en 0?

Para que el background permanezca estático y la vista sea la encargada de recorrer este

background.

**29.** ¿Qué quiere decir que la velocidad del avión principal se coloque a -2?

Que el avión ira a una velocidad tal que si el usuario no realiza ningún movimiento con las

teclas el avión continúe “moviéndose”

**30.** ¿Cómo se hace mover la vista? ¿A qué velocidad se mueve?

La velocidad será -2 para que se mueva paralelamente al avión y la vista se mueve habilitando

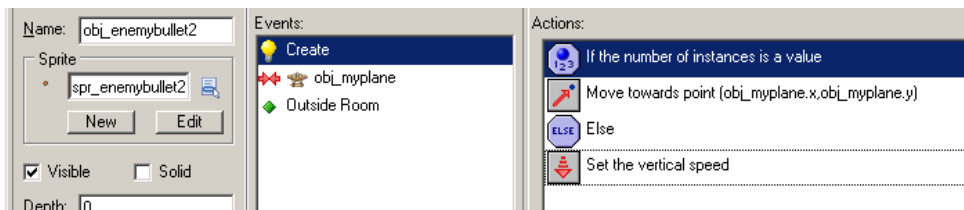
la opción de vistas

**31.** ¿Cómo se garantiza que el avión principal no se salga de la vista?

En las configuraciones de room, le das seguir al personaje y fijas el centro del punto de vista

para que maneje una cámara centrada Luego le das en activar vistas.

O Pones la velocidad horizontal a la misma rapidez vertical y con los límites x le das a una opción para que el objeto no se salga de los bordes.



**32.** ¿Cómo se garantiza que las balas se destruyen luego de abandonar la vista?

Se crea un evento que indique que cuando la bala no se vea se destruya eso se hace

previamente en un proceso anterior ya mencionado.

**33.** ¿Cuál es la diferencia entre la propiedad View in room y Port on Screen?

view in room: es considerada como un área de la sala que vamos a mostrar en la pantalla. Port on screen : es considerado como el área de la pantalla donde nos vamos a dibujar la vista.