

## GAME MAKER LANGUAGE CHEATCHEET

### Literales

Reales (3, -5.42, -199, 5/4) Cadenas/strings ("Hola mundo", 'Esto es texto')

Booleanos (true/1, false/0)

### Visualizar variables

show\_message(valor)

show\_debug\_message(valor)

### Sobre sintaxis

Comentario en línea: //

Comentarios en bloque: /\* \*/

; opcional al final de las sentencias

### Variables

Caract.: 64 car. máx; no especiales ni espacios, no empezar con número.

Notaciones: snake (puntos\_vida), camel case (PuntosVida, puntosVida)

Ámbito: global (global.nombre), de instancia (nombre), local (var nombre)

Nomenclatura del punto: objeto.variable\_instancia

### Macros

All configurations: para todos los proyectos.

Default: propias del proyecto actual.

Generación de semilla: randomize()

### Operadores

Aritméticos: + - \* / div mod

Unitarios y de asignación: = - ++ -- += -= \*= /=

Relacionales: == != < > <= >=

Lógicos: ! and or

### Expresiones

Orden de resolución: paréntesis > aritméticos > relacionales > lógicos

### Sentencias condicionales: if

```
if (condición==true) {
    instrucciones1;
}
else if (condicion2==true) {
    instrucciones2;
}
else {
    instrucciones3;
}
```

## GAME MAKER LANGUAGE CHEATSHEET

### Sentencias condicionales: switch

```
switch (opción) {
    case 0:
        instrucciones1;
        break;
    case 1:
        instrucciones2;
        break;
    default:
        instrucciones3;
        break;
}
```

### Sentencias iterativas: do-until

```
do {
    instrucciones;
} until (condición==true);
```

### Sentencias iterativas: while

```
while (condición==true) {
    instrucciones;
}
```

### Sentencias iterativas: repeat

```
repeat (nº_veces) {
    instrucciones;
}
```

### Sentencias iterativas: for

```
for (var i=0; i<5; i++) {
    instrucciones;
}
```

### Scripts (funciones)

Script "saludar":

show\_message(texto);

Uso de variables externas:

texto="Hola a todos!";

show\_message(texto);

Paso de argumentos:

Script "saludar":

show\_message(argument[0], argument[1], ...);

show\_message(argument0, argument1, ...);

Script ejemplo:

nombre=argument[0];

repeticiones=argument[1];

```
repeat (repeticiones) {
    show_message(nombre);
}
```

Luis Hernández López

Llamada al script: saludar("Luis",3);

Documentación, poner al principio:

///sumar(num1, num2)

Retorno de valores: return resultado;

Evitar conflicto entre variables de script y de instancia: poner locales en script.

NOTA DE BUENA PRÁCTICA: Cada script debe realizar una tarea concreta y simple.

### Funciones de redondeo

round(num)

floor(num)

ceil(num)

sign(num)

abs(num)

clamp(num,min,max)

### Funciones de aleatoriedad

choose(valor1, valor2, ...)

random(num)

irandom(num)

random\_range(num1,num2)

irandom\_range(num1,num2)

### Funciones de conversión

string (real)

real ("string")

### Objetos e instancias

id

object\_index

x

y

hspeed

vspeed

direction

speed

xstart

ystart

xprevious

yprevious

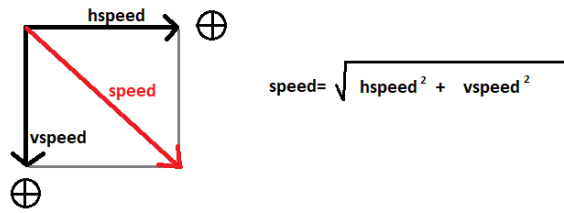
visible

depth

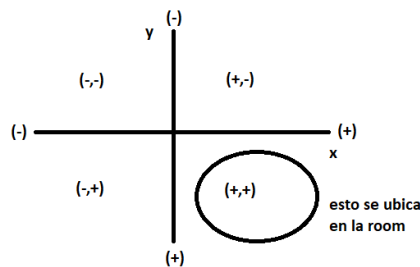
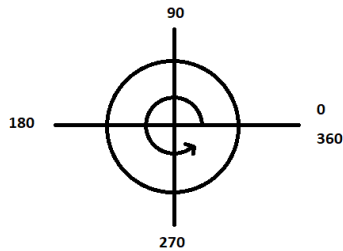
persistent

direction

object\_get\_name(object\_index)



$$\text{speed} = \sqrt{\text{hspeed}^2 + \text{vspeed}^2}$$



Ángulo positivo en sentido antihorario (en grados).

x positivas hacia la derecha  
y positivas hacia abajo

#### Herencia

event\_inherited()

**Sprites (R: lectura; W: escritura)**

sprite\_index(index): -1 quita el sprite

sprite\_width: R

sprite\_height: R

sprite\_get\_name(sprite\_index)

image\_xscale: RW, 1=100%

image\_yscale: RW, 1=100%

image\_angle: RW

image\_blend: RW

image\_alpha: RW

Mirroring con los scales

#### GAME MAKER LANGUAGE CHEATSHEET

NOTA: no es lo mismo direction que

image\_angle

Colores predefinidos: c\_white, c\_red...

#### Animación

image\_index

image\_speed

image\_number: R

Para que la velocidad no dependa del

room\_speed:

image\_speed=(image\_number/room\_speed)/  
(segundos\_duracion)

#### Draw

draw\_self()

draw\_set\_halign(fa\_center)

draw\_rectangle(x1,y1,x2,y2,outline)

draw\_set\_valign(fa\_middle)

draw\_set\_colour(c\_color)

draw\_text(x,y,string)

#### Inputs teclado

keyboard\_check(key)

keyboard\_check\_pressed(key)

keyboard\_check\_released(key)

#### Identificadores de keys de teclado

Generales: vk\_nokey, vk\_anykey, vk\_left, ...

Caracteres: ord("W") // Ordinal UNICODE

#### Input ratón

mouse\_check\_button(key)

mouse\_check\_pressed(key)

mouse\_check\_released(key)

#### Identificadores de keys de ratón

mb\_left // mb\_right // mb\_middle

mb\_none // mb\_any

mb\_wheel\_up // mb\_wheel\_down

#### Variables intrínsecas de posición de ratón

mouse\_x

mouse\_y

#### Función interesante

point\_direction(x1,y1,x2,y2)

#### Rooms

room\_width

room\_height

room\_speed

room

room\_first

room\_last

Luis Hernández López

room\_previous(room)

room\_next(room)

room\_restart()

room\_exists(room)

room\_goto(room)

room\_goto\_next()

room\_goto\_previous()

room\_get\_name(room)

#### Alarma

creación>llamada>cuentaAtras>ejecución>

finalización

12 alarmas/objeto

alarm[0]=room\_speed\*3;

#### Sonidos - básico

audio\_play\_sound(sound\_id, prioridad, loop)

audio\_stop\_sound(sound\_id)

audio\_pause\_sound(sound\_id)

audio\_resume\_sound(sound\_id)

#### Sonidos - comprobaciones

audio\_is\_playing(sound\_id)

audio\_is\_paused(sound\_id)

#### Sonidos - operaciones globales

audio\_stop\_all()

audio\_pause\_all()

audio\_resume\_all()

#### Arrays 1D

arreglo[0]=1.25 // arreglo[3]="Hola"

array\_length\_1d(arreglo)

arreglo[array\_length\_1d(arreglo)] = valor

arreglo[posición] = noone

#### Arrays 2D

@ es el accesor

arreglo[0,2] // 0: fila ; 2: columna

array\_height\_2d

array\_length\_2d(arreglo, fila)