

APRENDE GAMESMAKER

GUIA DE ACCIONES DRAG & DROP EN CODIGO GML

<http://www.aprendegamemaker.com/>

David López 2015

<http://www.aprendegamemaker.com/>

Si has empezado con Game Maker, es muy posible que hayas utilizado las acciones Drag&Drop para programar tus objetos, has podido hacer juegos sencillos y te has habituado a la [interfaz del programa](#).

Una vez hecho varios proyectos, es el momento de dar el salto, hacer cosas más complejas, y eso pasa por codificar en gml. Pero si no estás acostumbrado a programar, pasar de hacer acciones Drag and Drop a escribir en GML puede ser muy frustrante, ya que el cambio puede ser un poco confuso.


Una manera de hacerlo más fácil de entender es relacionar cada botón de acción con una función (a veces un grupo de [funciones o instrucciones](#)). De esta manera, sustituimos una acción por código GML, y la transición resulta más fácil.

En versiones anteriores de Game Maker, veía en internet documentos con la equivalencia entre acciones y GML. Como no lo he visto en Game Maker Studio, a continuación vamos a hacer una referencia dónde veremos la acción y su código equivalente.

Explicación de la tabla de referencia

La intención es ayudar a los usuarios para familiarizarse con el código, y así utilizar el lenguaje de Game Maker. No explicaremos las funciones, ni sus parámetros de entrada o su salida, ni poner ejemplos de cómo usarlos. Eso ya lo iremos haciendo en futuros posts, además de la [documentación de Game Maker](#).

Para cada acción, haré una tabla con 3 columnas: la primera con el icono y nombre de la acción, la segunda las opciones que aparecen para modificar dicha acción y la tercera su código equivalente. Tendrá un aspecto parecido a éste:

Acción	Opciones	Código GML
 Move fixed	Direction: seleccionable con fechas Speed: número	<pre>///Código GML speed = numero;</pre>

La columna de **Opciones** concretaremos el tipo de datos, número o texto principalmente, que espera como entrada, y veremos que tendrá una equivalencia en el código.

El código GML puede ser, como hemos dicho anteriormente, una función o una serie de instrucciones varias. A veces pondrá dos posibles soluciones, para que podamos comparar. Es posible que veamos más parámetros en las funciones GML que en las

<http://www.aprendegamemaker.com/>

acciones, simplemente pondré los que den el mismo resultado, sabiendo que un cambio en ellos hará otros efectos. Por ejemplo:

```
draw_text(x, y, "Hola");
draw_text_ext(x, y, "Hola", 1, 1);
draw_text_colour(x, y, "Hola", c_white, c_white, c_white,
c_white, 1);
draw_text_transformed(x, y, "Hola", 1, 1, 0);
```

Son iguales, pero usar una función u otra es para poder usar los parámetros que más se adecuan a cada situación. En esta referencia sólo pondré una función, y si existiesen más parámetros dejaré el valor por defecto.

Código GML de los atributos comunes

Ya comentamos en su momento, en los [detalles de una acción en Game Maker](#), que existen propiedades comunes a la mayoría de las acciones, y que no explicaré en cada acción. Nos referimos al apartado Applies to, al check Relative y el check NOT (de negación).

Applies to / Aplica a

En este apartado vemos tres opciones, para saber que dicha acción se ejecuta en *Self*, el objeto dónde se añade, *Other*, el objeto que afecta a la colisión y *Object*: cualquier objeto que le digamos.

Para Self no hay que hacer nada extra, pero para las otras opciones usaremos la instrucción with.

```
with (other) {
    //Código GML de la tercera columna
}
```


```
with (nombre_objeto) {
    //Código GML de la tercera columna
}
```

Relative

Si damos un valor a una variable, por ejemplo score, con la acción correspondiente, perderemos el valor que ya tenía. Esta casilla se utiliza para no perder ese valor y

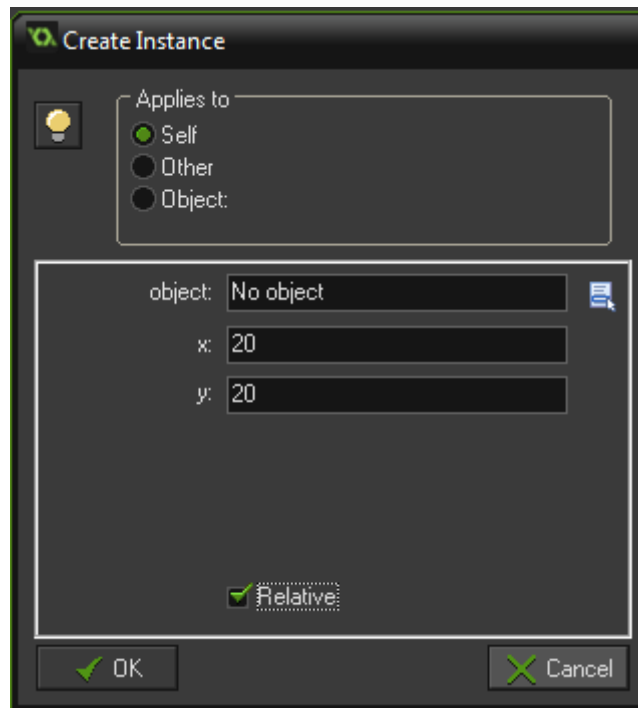
<http://www.aprendegamemaker.com/>

aumentarlo/incrementarlo. Su equivalencia se hace con el operador `+=`. Por ejemplo,

para aumentar en 10 puntos la variable `score`, usamos *Set Variable* , en value ponemos 10 y marcamos el check *Relative*. El código equivalente en GML es:

```
score += 10;
```

Otro ejemplo que se usaría en una función, si creamos un objeto con el check *Relative*, sería así:



```
instance_create(x + 20, y + 20, object_enemy);
```

NOT

En las acciones comparación, se suele distinguir fácilmente porque tienen forma de hexágono, suele verse el check **NOT**. Se utiliza para hacer la negación de la pregunta, y se usa el símbolo de exclamación. Por ejemplo

```
if (x != 50)
```

```
if (!place_free(x, y))
```

Recordemos que en funciones que devuelven si es verdad o falso, booleanas, tenemos varias maneras de escribir la condición.

<http://www.aprendegamemaker.com/>

```
//Todo esto es lo mismo  
if (!place_free(x, y))  
  
if (place_free(x, y) == false)  
  
if (place_free(x, y) != true)
```


Cuando programamos podemos hacerlo de la manera en que nos sintamos más a gusto.


Y ahora sí, empieza la guía de acciones para convertir a código GML.


Guía de acciones a código GML


Tienes a continuación la guía completa de todas las acciones y su código. Verás un apartado para cada pestaña.


La pestaña move


Acción	Opciones	Código GML
 Move fixed	direction: seleccionable con flechas speed: número	Usando variables <pre> flecha1 = 0; flecha2 = 45; flecha3 = 90; flecha4 = 135; flecha5 = 180; flecha6 = 225; flecha7 = 270; flecha8 = 315; //Añadir por cada flecha marcada/elegida. direction = choose(arrow1, arrow2, arrow3, ...); speed = numero; Equivalente al anterior, usando las flechas y la misma variable de direction motion_set(direction, speed); </pre>


Acción	Opciones	Código GML
 Move free	direction: numero speed: número	Usando variables <pre> direction = numero; speed = numero; Usando una función motion_set(direction, speed); </pre>


Acción	Opciones	Código GML
 Move Towards	x: número y: número speed: número	<code>move_towards_point(x, y, speed);</code>


Acción	Opciones	Código GML
 Speed Horizontal	Hor. speed: número	<code>hspeed = numero;</code>


Acción	Opciones	Código GML
 Speed Vertical	Vert. speed: número	<code>vspeed = numero;</code>


Acción	Opciones	Código GML
 Set Gravity	direction: número gravity: número	<code>gravity_direction = numero;</code> <code>gravity = numero;</code>


Acción	Opciones	Código GML
 Reverse Horizontal		<code>hspeed = -hspeed;</code>


Acción	Opciones	Código GML
 Reverse Vertical		<code>vspeed = -vspeed;</code>


Acción	Opciones	Código GML
 Set Friction	Friction: número	<code>friction = numero;</code>


Acción	Opciones	Código GML
 Jump to Position	x: número y: número	<code>x = numero;</code> <code>y = numero</code>


Acción	Opciones	Código GML
 Jump to Start		<code>x = xstart;</code> <code>y = ystart;</code>


Acción	Opciones	Código GML
 Jump to Random	Snap hor: número Snap vert: número	<code>move_random(snap_hor, snap_vert);</code>

Acción	Opciones	Código GML
 Align to grid	Snap hor: número Snap vert: número	<code>move_snap(snap_hor, snap_vert);</code>


Acción	Opciones	Código GML
 Wrap Screen	direction: horizontal vertical in both directions	Horizontal <pre>move_wrap(true, false, sprite_width / 2);</pre> Vertical <pre>move_wrap(false, true, sprite_width / 2);</pre> En ambas direcciones <pre>move_wrap(true, false, sprite_width / 2); move_wrap(false, true, sprite_height / 2);</pre>


Acción	Opciones	Código GML
 Move to Contact	direction: número maximum: número against: solid objects all objects	Objetos sólidos <pre>move_contact_solid(direction, maximum);</pre> Todos los objetos <pre>move_contact_all(direction, maximum);</pre>


Acción	Opciones	Código GML
 Bounce	precise: booleano against: solid objects all objects	Objetos sólidos <pre>move_bounce_solid(precise);</pre> Todos los objetos <pre>move_bounce_all(precise);</pre>


Acción	Opciones	Código GML
 Set Path	path: ruta speed: número at end (número): 0-stop 1-continue from start 2-continue from here 3-reverse relative: booleano	<code>path_start(path, speed, at_end, relative);</code>

Acción	Opciones	Código GML
 End Path		<code>path_end();</code>


Acción	Opciones	Código GML
 Path Position	position: número 0-1	<code>path_position = numero;</code>


Acción	Opciones	Código GML
 Path Speed	speed: número	<code>path_speed = numero;</code>


Acción	Opciones	Código GML
 Step Towards	x: número y: número speed: número stop at: booleano	<code>mp_linear_step(x, y, speed, stop_at);</code>


Acción	Opciones	Código GML
 Step Avoiding	x: número y: número speed: número avoid: booleano	<code>mp_potential_step(x, y, speed, avoid);</code>


La pestaña main1


Acción	Opciones	Código GML
 Create Instance	object: objeto x: número y: número	<code>instance_create(x, y, objeto);</code>


Acción	Opciones	Código GML
 Create Moving	object: objeto x: número y: número speed: número direction: número	Utilizando función <code>obj = instance_create(x, y, objeto);</code> <code>with (obj) {motion_set(direction, speed);}</code> Utilizando variables <code>obj = instance_create(x, y, objeto);</code> <code>obj.speed = numero;</code> <code>obj.direction = numero;</code>


Acción	Opciones	Código GML
 Create Random	object1: objeto object2: objeto object3: objeto object4: objeto x: número y: número	<code>//en GML, la función choose permite ¡hasta 15 parámetros!</code> <code>obj = choose(object1, object2, object3, object4);</code> <code>instance_create(x, y, obj);</code>


Acción	Opciones	Código GML
 Change Instance	change into: objeto perform events: booleano	<pre>instance_change(change_into, perform_events);</pre>


Acción	Opciones	Código GML
 Destroy Instance		<pre>instance_destroy();</pre>


Acción	Opciones	Código GML
 Destroy at Position	x: número y: número	<pre>position_destroy(x, y);</pre>


Acción	Opciones	Código GML
 Change Sprite	sprite: sprite subimage: número speed: número	<pre>sprite_index = nombre_sprite; image_index = numero; image_speed = numero;</pre>


Acción	Opciones	Código GML
 Transform Sprite	xscale: número yscale: número angle: número mirror: no mirroring mirror horizontal flip vertical mirror and flip	<pre>image_xscale = numero; image_yscale = numero; image_angle = numero; //mirror horizontal image_xscale *= -1; //flip vertical image_yscale *= -1;</pre>


Acción	Opciones	Código GML
 Color Sprite	color: color alpha: número	<pre>image_blend = color; image_alpha = numero;</pre>


Acción	Opciones	Código GML
 Play Sound	sound: sound loop: booleano	<pre>audio_play_sound(nombre_sonido, 0, loop);</pre>


Acción	Opciones	Código GML
 Stop Sound	sound: sound	<pre>audio_stop_music(nombre_sonido);</pre>


Acción	Opciones	Código GML
 Check Sound	sound: sound	<pre>if sound_isplaying(nombre_sonido)</pre>


Acción	Opciones	Código GML
 Previous Room		<pre>room_goto_previous();</pre>

Acción	Opciones	Código GML
 Next Room		<pre>room_goto_next();</pre>


Acción	Opciones	Código GML
 Restart Room		<code>room_restart();</code>


Acción	Opciones	Código GML
 Different Room	room: room	<code>room_goto(room);</code>


Acción	Opciones	Código GML
 Check Previous		<code>if (room_previous(room) != -1)</code>


Acción	Opciones	Código GML
 Check Next		<code>if (room_next(room) != -1)</code>


La pestaña main2


Acción	Opciones	Código GML
 Set Alarm	number of steps: número in alarm no: 0-11	<code>alarm[0..11] = numero;</code>


Acción	Opciones	Código GML
 Set Time Line	time line: timeline position: número start: booleano loop: booleano	<pre> timeline_index = nombre_timeline: timeline_running = booleano; timeline_lop = booleano; </pre>


Acción	Opciones	Código GML
 Time Line Position	position: número	<pre> timeline_position = numero; </pre>


Acción	Opciones	Código GML
 Time Line Speed	speed: número	<pre> timeline_speed = numero; </pre>


Acción	Opciones	Código GML
 Start Time Line		<pre> timeline_running = true; </pre>


Acción	Opciones	Código GML
 Pause time Line		<pre> timeline_running = false; </pre>

Acción	Opciones	Código GML
 Stop Time Line		<pre> timeline_running = false; timeline_position = 0; </pre>

Acción	Opciones	Código GML
 Display Message	message: cadena	<code>show_message(texto);</code>


Acción	Opciones	Código GML
 Open URL	URL: url	<code>url_open(url);</code>


Acción	Opciones	Código GML
 Restart Game		<code>game_restart();</code>

Acción	Opciones	Código GML
 End Game		<code>game_end();</code>


Acción	Opciones	Código GML
 Save Game	filename: nombre archivo	<code>game_save(nombre_archivo);</code>


Acción	Opciones	Código GML
 Load Game	file name: nombre archivo	<code>game_load(nombre_archivo);</code>


Acción	Opciones	Código GML
 Replace Sprite	sprite: sprite filename: nombre archivo images: número	<code>sprite_replace(sprite, nombre_archivo, numero, 0, 0);</code>


Acción	Opciones	Código GML
 Replace Background	background: background filename: nombre archivo	<code>background_replace(nombre_background, nombre_archivo);</code>


La pestaña control


Acción	Opciones	Código GML
 Check Empty	x: número y: número objects: Only Solid All	Objetos sólidos <code>if (place_free(x, y))</code> Todos los objetos <code>if (place_empty(x, y))</code>


Acción	Opciones	Código GML
 Check Collision	x: número y: número objects: Only Solid All	Objetos sólidos <code>if (place_free(x, y))</code> Todos los objetos <code>if (place_empty(x, y))</code>


Acción	Opciones	Código GML
 Check Object	object: objeto x: número y: número	<pre>if (place_meeting(x, y, nombre_objeto))</pre>


Acción	Opciones	Código GML
 Test Instance Count	object: objeto number: número operation: equal to <small>smaller than</small> <small>larger than</small>	<p>Igual a</p> <pre>if (instance_number(nombre_objeto) == numero)</pre> <p>Menor que</p> <pre>if (instance_number(nombre_objeto) < numero)</pre> <p>Mayor que</p> <pre>if (instance_number(nombre_objeto) > numero)</pre>


Acción	Opciones	Código GML
 Test Chance	side: número	<pre>if(floor(random(numero)) == 0)</pre>


Acción	Opciones	Código GML
 Check Question	question: cadena	<pre>if (show_question(cadena))</pre>


Acción	Opciones	Código GML
 Test Expression	expression: expresion	if (expresion)


Acción	Opciones	Código GML
 Check Mouse	button: no left middle right	if(mouse_check_button(button))


Acción	Opciones	Código GML
 Check Grid	Snap Hor: número Snap Vert: número	if (place_snapped(snap_hor, snap_vert))


Acción	Opciones	Código GML
 Start Block		{


Acción	Opciones	Código GML
 End Block		}


Acción	Opciones	Código GML
 Else		else


Acción	Opciones	Código GML
 Exit Event		<code>exit;</code>


Acción	Opciones	Código GML
 Repeat	times: número	<code>repeat(numero)</code>


Acción	Opciones	Código GML
 Call Parent Event		<code>event_inherited();</code>


Acción	Opciones	Código GML
 Execute Code		<i>No aplica</i>

Acción	Opciones	Código GML
 Execute Script	script: Script argument0: parámetro argument1: parámetro argument2: parámetro argument3: parámetro argument4: parámetro	Se llama al nombre del script con sus posibles parámetros <code>script();</code>


Acción	Opciones	Código GML
 Comment	comment: cadena	Una línea //cadena Un bloque de líneas /* cadena */


Acción	Opciones	Código GML
 Set Variable	variable: nombre variable value: valor	nombre_variable = valor;


Acción	Opciones	Código GML
 Test Variable	variable: nombre variable value: valor operation: equal to less than greater than less than or equal to greater than or equal to	Igual a if (nombre_variable == valor) Menor que if (nombre_variable < valor) Mayor que if (nombre_variable > valor) Menor o igual que if (nombre_variable <= valor) Mayor o igual que if (nombre_variable >= valor)


Acción	Opciones	Código GML
 Draw Variable	variable: nombre_variable x: número y: número	<code>draw_text(x, y, nombre_variable);</code>


La pestaña score


Acción	Opciones	Código GML
 Set Score	new score: número	<code>score = numero;</code>


Acción	Opciones	Código GML
 Test Score	value: número operation: equal to <small>smaller than</small> <small>larger than</small>	Igual que <code>if (score == numero)</code> Menor que <code>if (score < numero)</code> Mayor que <code>if (score > numero)</code>


Acción	Opciones	Código GML
 Draw Score	x: número y: número caption: cadena	<code>draw_text(x, y, "cadena" + string(score));</code>


Acción	Opciones	Código GML
 Clear highscore		<code>highscore_clear();</code>


Acción	Opciones	Código GML
 Set Lives	new lives: número	<code>lives = numero;</code>


Acción	Opciones	Código GML
 Test Lives	value: número operation: equal to smaller than larger than	Igual que <code>if (lives == numero)</code> Menor que <code>if (lives < numero)</code> Mayor que <code>if (lives > numero)</code>

Acción	Opciones	Código GML
 Draw Lives	x: número y: número caption: cadena	<code>draw_text(x, y, "cadena" + string(lives));</code>

Acción	Opciones	Código GML
 Draw Life Images	x: número y: número image: nombre sprite	<pre>for (i = 0; i < lives; i++) { ancho = i * sprite_get_width(nombre_sprite); draw_sprite(nombre_sprite, 0, x + ancho, y); }</pre>

Acción	Opciones	Código GML
 Set Health	value (0-100): número	<pre>health = numero;</pre>


Acción	Opciones	Código GML
 Test Health	value: número operations: equal to smaller than larger than	Igual a <pre>if (health == numero)</pre> Menor que <pre>if (health < numero)</pre> Mayor que <pre>if (health > numero)</pre>

Acción	Opciones	Código GML
 Draw Health	x1: número y1: número x2: número y2: número back color: color bar color: color	<pre>draw_healthbar(x1, y1, x2, y2, health, back_color, bar_color, 0, true, true);</pre>


<http://www.aprendegamemaker.com/>


La pestaña extra


Todas las acciones de partículas se detallarán próximamente.


Acción	Opciones	Código GML
 Set Cursor	sprite: sprite cursor: don't cursor show	<pre>cursor_sprite = nombre_sprite;</pre> <p>Sin cursor</p> <pre>windows_set_cursor(cr_none);</pre> <p>Monstrando cursor</p> <pre>windows_set_cursor(cr_default);</pre>


La pestaña draw


Acción	Opciones	Código GML
 Draw Self		<pre>draw_self();</pre>


Acción	Opciones	Código GML
 Draw Sprite	sprite: sprite x: número y: número subimage: número	<pre>draw_sprite(nombre_sprite, subimage, x, y);</pre>


Acción	Opciones	Código GML
 Draw Background	background: background x: número y: número tiled: booleano	Si tiled = false <code>draw_background(nombre_background, x, y);</code> Si tiled = true <code>draw_background_tiled(nombre_background, x, y);</code>


Acción	Opciones	Código GML
 Draw Text	text: cadena x: número y: número	<code>draw_text(x, y, cadena);</code>


Acción	Opciones	Código GML
 Draw Scaled Text	text: cadena x: número y: número xscale: número yscale: número angle: número	<code>draw_text_transformed(x, y, cadena, xscale, yscale, angle);</code>


Acción	Opciones	Código GML
 Draw Rectangle	x1: número y1: número x2: número y2: número filled: booleano	<code>draw_rectangle(x1, y1, x2, y2, filled);</code>


Acción	Opciones	Código GML
 Horizontal Gradient	x1: número y1: número x2: número y2: número color1: color color2: color	<code>draw_rectangle_color(x1, y1, x2, y2, color1, color2, color2, color1, false);</code>


Acción	Opciones	Código GML
 Vertical Gradient	x1: número y1: número x2: número y2: número color1: color color2: color	<code>draw_rectangle_color(x1, y1, x2, y2, color1, color1, color2, color2, false);</code>


Acción	Opciones	Código GML
 Draw Ellipse	x1: número y1: número x2: número y2: número filled: booleano	<code>draw_ellipse(x1, y1, x2, y2, filled);</code>


Acción	Opciones	Código GML
 Gradient Ellipse	x1: número y1: número x2: número y2: número color1: color color2: color	<code>draw_ellipse_color(x1, y1, x2, y2, color1, color2, false);</code>

Acción	Opciones	Código GML
 Draw Line	x1: número y1: número x2: número y2: número	<code>draw_line(x1, y1, x2, y2);</code>


Acción	Opciones	Código GML
 Draw Arrow	x1: número y1: número x2: número y2: número tip size: número	<code>draw_arrow(x1, y1, x2, y2, tip_size);</code>

Acción	Opciones	Código GML
 Set Color	color: color	<code>draw_set_color(color);</code>

Acción	Opciones	Código GML
 Set Font	font: font align: left center right	<code>draw_set_font(font);</code> <code>draw_set_halign(align);</code>

Acción	Opciones	Código GML
 Set Full Screen	action switch window fullscreen Modo ventana Pantalla completa	<pre>window_set_full_screen(!window_get_fullscreen());</pre> <pre>window_set_full_screen(false);</pre> <pre>window_set_full_screen(true);</pre>

Acción	Opciones	Código GML
 Take Snapshot	filename: nombre archivo	<pre>screen_save(nombre_archivo);</pre>

Acción	Opciones	Código GML
 Create Effect	type: effect x: número y: número size: small medium large color: color where: below objects above objects	<pre>effect_create_below(type, x, y, size, color);</pre> <pre>effect_create_above(type, x, y, size, color);</pre>