Drag and Drop Icons and Their GML Equivalents for Version 7.0

Coding, Sprites, Logo © Mark Overmars, http://www.yoyogames.com/
Prepared by D. Eugene Perry, http://www.blackratstudios.com/

Table of Contents

Notes

Reference Colors

These are the colors that are used on the following pages and what they mean…

Green - Comments
Purple – User input
Blue – Built in Variables or statements.

Assigning statements to other objects

Drag and drop icons allow you to assign statements to other objects. You can do this in code with the following...

```
//to assign to another object...
with (object) {
//actions here
}


//To assign to the other object in a collision...
with (other) {
//actions here
}

//To assign the statement to that object...
with (self) {
//actions here
}
```

Using 'ID' as opposed to 'id'

Please note that throughout the document I show examples of assigning code to other objects from a separate object. In order to make it easier I use the capitalization, 'ID' which Game Maker recognizes as a local variable. This is not to be confused with 'id' which is an inherited variable that will give you the current object's unique id number.

Move Tab

Move section

 'Move Fixed'
 'Move Free'
motion_set(direction,speed);

 'Move Towards'
move_towards_point(x,y,speed);

 'Speed Horizontal'
hspeed=speed;

 'Speed Vertical'
vspeed=speed;

 'Set Gravity'
gravity_direction=direction;
gravity=amount;

 'Reverse Horizontal'
hspeed=-hspeed; //actual code.

 'Reverse Vertical'
vspeed=-vspeed; //actual code.

 'Set Friction'
friction=amount;

Jump Section

 'Jump to Position'
x=value;
y=value;

 'Jump to Start'
x=xstart; //actual code.
y=ystart; //actual code.

 'Jump to Random'
move_random(1,1);
//actual code. The 1 and 1 in the code are the hsnap and vsnap positions.

 'Align to grid'
move_snap(hsnap,vsnap);

 'Wrap Screen'
move_wrap(hort,vert,margin);
// This code should be placed in the outside room event. Set hort (horizontally) and vert (vertically) to either 1 for true or 0 for false. Set margin to how far outside the room the instance must be before the action happens.

 'Move to Contact'
move_contact_solid(dir,maxdist) //for solid objects

move_contact_all(dir,maxdist)
//for non solid objects, dir=direction, maxdist=maximum distance.

 'Bounce'
move_bounce_solid(advanced);
//for solid objects, advanced=advance bounce(0 or 1).

move_bounce_all(advanced);
//for all objects, advanced=advance bounce(0 or 1).

Paths Section

 'Set Path'
path_start(path,speed,endaction,absolute);

 'End Path'
path_end();

 'Path Position'
path_position=value; //Must lie between 0 and 1.

 'Path Speed'
path_speed=value; //pixels per step.

Steps Section

 'Step Towards'
mp_linear_step(x,y,stepsize,checkall);
// stepsize is in pixels. Checkall can be either 1 for stopping when hitting any object, or 0 for only solid objects.

 'Step Avoiding'
mp_potential_step(x,y,stepsize,checkall);

Main1 Tab

Objects Section

 'Create Instance'
instance_create(x,y,object0); //use x and y variables for relative.

 'Create Moving'
//No equivalent, but you can use the following code.
ID = instance_create(x,y,object1);
with (ID) motion_set(direction,speed);

 'Create Random'
instance_create(x,y,choose(object0,object1,object2,object3));
//object0, etc are the object names. Drag and Drop allows only four slots, but the 'choose' code allows up to sixteen.

 'Change Instance'
instance_change(obj,perf);
//perf(1 or 0)is whether or not to perform create and destroy events.

 'Destroy Instance'
instance_destroy();

 'Destroy at Position'
position_destroy(x,y);

Sprite Section

 'Change Sprite'
sprite_index=sprite0;

 'Transform Sprite'
image_xscale=value; //horizontal scaling of sprite.
image_yscale=value; //vertical scaling if the sprite.
image_angle=value; //angle the sprite.
image_xscale=-1; //flip the sprite horizontally, actual code.
image_yscale=-1; //flip the sprite vertically, actual code.

 'Color Sprite'
image_blend=color;
image_alpha=value; //from 0 to 1, 1 being opaque.

Sounds Section

 'Play Sound'
sound_play(sound); //plays sound once.
sound_loop(sound); //loops sound.

 'Stop Sound'
sound_stop(index);
//Stops the indicates sound. If there are multiple sounds with this index playing simultaneously, all will be stopped.

 'Check Sound'
if sound_isplaying(sound)=true {
// actions here.
}

Rooms Section

//If you wish to use transitions with the following statements call this statement first. Transition types can be found in Appendix A.

transition kind=value;

 'Previous Room'
room_goto_previous();

 'Next Room'
room_goto_next();

 'Restart Room'
room_restart();

 'Different Room'
room_goto(room);

 'Check Previous'
if room_previous(room)<>-1 then {
// actions here
}
//'room' is constant variable for current room. Actual code.

 'Check Next'
```
if room_next(room)<>-1 then {
// actions here.
}
//'room' is constant variable for current room. Actual code.
```

Main 2 Tab

Timing Section

 'Set Alarm'
```
alarm[0]=value; //set 0 from 0 to 11 for alarm.
```

 'Sleep'
```
sleep(numb); //'numb' is in milliseconds.
```

 'Set Timeline'
```
timeline_index=timeline;
```

 'Set Timeline Position'
```
timeline_position=value;
```

Info Section

 'Display Message'
```
show_message('Hello');
```

 'Show Info'
```
show_info();
```

 'Show Video'
```
show_video(fname,full,loop); //full and loop are either 1 or 0 for yes or no.
```

Game Section

 'Restart Game'
game_restart();

 'End Game'
game_end();

 'Save Game'
game_save(fname);
//fname is the name of the save file. Place it in quotes.

 'Load Game'
game_load(fname);
//fname is the name of the save file to load. Place it in quotes.

Resources Section

 'Replace Sprite'
sprite_replace(ind,fname,imgnumb,precise,transparent,smooth,preload,xorig,yorig);
// precise, transparent,smooth,preload are all 1 or 0 for yes or no.

 'Replace Sound'
sound_replace(index,fname,kind,loadonuse);
//loadonuse is 1 or 0 for yes or no.
//Kind is one of the following...
0-normal
1-background
2-3d
3-mmplayer

 'Replace Background'
background_replace(ind,fname,transparent,smooth,preload);
// transparent,smooth and preload are all 1 or 0 for yes or no.

Control tab

Questions Section

 'Check Empty'
```
if place_free(x,y) {
//actions here.
}
//for solid

if !place_empty(x,y)
//actions here.
}
//for all
```

 'Check Collision'
```
if !place_empty(x,y) // for All
if place_meeting(x,y,all) // for All

if !place_free(x,y) // for Solid only

// As well, there are several advanced codes that allow you greater control over
checking collisions. Please see the manual for explanations of each...

if collision_point(x,y,obj,prec,notme) {
//actions here.
if collision_rectangle(x1,y1,x2,y2,obj,prec,notme) {
//actions here.
}
if collision_circle(xc,yc,radius,obj,prec,notme) {
//actions here.
}
if collision_ellipse(x1,y1,x2,y2,obj,prec,notme) {
//actions here.
}
if collision_line(x1,y1,x2,y2,obj,prec,notme) {
//actions here.
}
```

'Check Object'
```
if place_meeting(x,y,object0) {
//actions here.
}
```

'Test Instance Count'
```
if instance_number(obj)=value {
//actions here.
}
```

'Test chance'
```
if floor(random(value))=0 {
//actions here.
}
```

'Check Question'
```
if show_question('Do you want to do this?') {
// actions here.
}
```

'Test Expression'
```
if (the expression) {
// actions here.
}
//examples for expression would be x=5, y>10, global.item='Apple'.
```

'Check Mouse'
```
if mouse_check_button(numb) {
//actions here.
}
// numb can be mb_none,mb_left, mb_middle,mb_right.
```

'Check Grid'
```
if place_snapped(value,value) {
//actions here.
}
```

Other Section

 'Start Block'

 'End Block'

 'Else'

```
//All above are part of if, else statements example...
if x=50 {
hspeed=2;
vspeed=-2;
}
else {
motion_set(90,1);
}
```

 'Exit Event'

```
exit;
```

 'Repeat'

```
repeat (value) <statement>;
//example: repeat (10) instance_create(x,y,object0);
```

 'Call Parent Event'

```
event_inherited();
```

Code Section

 'Execute Code'

```
//This is the icon that all coding is placed in.
```

 'Execute Script'

```
script_execute(ind,arguments);
// or call a script in code by the script name and the arguments in () beside it…
```

 'Comment'

```
//Enter a comment in code by putting '//' followed by the comment.
```

Variables Section

'Set Variable'
```
// Set a variable by either using a built-in variable or by using your own.
example...
health=50;
lives=3;
name='Gordon';
//Use 'global.' for your own variables that are to be used by more than one
object...
global.name='Gordon';
//You do not need to use global for built-in variables like 'lives', or 'score'.
```

'Test Variable'
```
//Use an if statement to check this. Example...
if lives=0 {
//actions here
}
```

'Draw Variable'
```
draw_text(x,y,global.name);
draw_text(x,y,lives);
```

Score Tab

Score Section

'Set Score'
```
score=value;
```

'Test Score'
```
if score=value {
//actions here.
}
```

'Draw Score'
```
draw_text(x,y,'Score: ' + string(score));
```

 'Show Highscore'
highscore_set_background(back); //set with background image.
highscore_set_border(show); //1 or 0 for yes or no.
highscore_set_colors(back,new,other); //set colors for background,new entry, other entries.
highscore_set_font(name,size,style); //set style to 0=normal, 1=bold, 2=italic, 3=bold italic.
highscore_show(numb); //This actually shows the table, with numb being the new score to add if it is high enough.
//Note: there are many other controls for the high score table. These are just the ones used in the drag and drop.

 'Clear Highscore'
highscore_clear();

Lives Section

 'Set Lives;
lives=value;

 'Test Lives'
if lives=value {
//actions here.
}
 'Draw Lives'
draw_text(x,y,'Lives: ' + string(lives));

 'Draw Life Images'
// no equivalent but you can use the following code in the draw event. sprite0 is the sprite image. Set 'a' in the 5th line to however far apart you wish the images to be on the screen in pixels...
var a;
a=0;
repeat(lives){
draw_sprite(sprite0,0,view_xview+a,view_yview);
a+=16;
}

Health Section

 'Set Health'
health=value;

 'Test Health'
if health=value {
//actions here.
}

 'Draw Health'
draw_healthbar(x1,y1,x2,y2,amount,backcol,mincol,maxcol,direction,showback,showborder);

 'Score Caption'
show_score=value; //set to 1 for yes, 0 for no.
caption_score=string;
show_lives=value; //set to 1 for yes, to 0 for no.
caption_lives=string;
show_health=value; //set to 1 for yes, to 0 for no.
caption_health=string;

Extra Tab

Particles Section

 'Create Part System'
index=part_system_create(); //Assign to an index (variable). Must be used in other functions.

 'Destroy Part System'
part_system_destroy(index);

 'Clear Part system'
part_system_clear(index);

 'Create Particle'
index=part_type_create(); //Assign to an index.
part_type_shape(index,shape); //see manual for shape types.
part_type_size(index,size_min,size_max,size_incr,size_rand);
part_type_color(index,color_start,color_middle,color_end);
//there are other functions for particles; these just cover the Drag and Drop.

 'Particle Color'
part_type_color1(ind,color1)
//Indicates a single color to be used for the particle.
part_type_color2(ind,color1,color2)
//Specifies two colors between which the color is interpolated.
part_type_color3(ind,color1,color2,color3)
//Similar but this time the color is interpolated between three colors that represent the color at the start, half-way, and at the end.
part_type_color_mix(ind,color1,color2)
//With this function you indicate that the particle should get a color that is a random mixture of the two indicated colors. This color will remain fixed over the lifetime of the particle.
part_type_color_rgb(ind,rmin,rmax,gmin,gmax,bmin,bmax)
//Can be used to indicate that each particle must have a fixed color but chosen from a range. You specify a range in the red, green, and blue component of the color (each between 0 and 255).
part_type_color_hsv(ind,hmin,hmax,smin,smax,vmin,vmax)
//Can be used to indicate that each particle must have a fixed color but chosen from a range. You specify a range in the hue saturation and value component of the color (each between 0 and 255).

 'Particle Life'
part_type_life(index,life_min,life_max);

 'Particle Speed'
part_type_speed(index,speed_min,speed_max,speed_incr,speed_rand);
part_type_direction(index,dir_min,dir_max,dir_incr,dir_rand);

 'Particle Gravity'
part_type_gravity(index,grav_amount,grav_dir);

 'Particle Secondary'
part_type_death(index,death_number,death_type);

 'Create Emitter'
index=part_emitter_create(ps);
// ps is the index of the particle system. You must assign this to an index.
part_emitter_region(ps,index,xmin,xmax,ymin,ymax,shape,distribution);
//ps is the index of the particle system. Index is the index of the emitter.

 'Destroy Emitter'
part_emitter_destroy_all(ps) //ps is the index of the emitter.

 'Burst from Emitter
part_emitter_burst(ps,index,parttype,number) ;
// ps is the index of the particle system. Index is the index of the emitter. Parttype is the index of the particle.

 'Stream from Emitter'
part_emitter_stream(ps,index,parttype,number);

CD Section

//You must call the function cd_init(); before calling other Cd functions.

 'Play CD'
cd_play(first,last);

 'Stop CD'
cd_stop();

 'Pause CD'
cd_pause();

 'Resume CD'
cd_resume();

 'Check CD'
if cd_present()=true {
//actions here.
}

 'Check CD playing'
if cd_playing()=true {
//actions here.
}

Other Section

 'Set Cursor';
window_set_cursor(curs);
//this will set the cursor to a default setting (see end of document for types)
to have a custom sprite as a cursor use the following statement…
cursor_sprite=sprite0;
//Change cursor image to sprite index.

 'Open a Web Page'
execute_shell('http//www.somepage.com',0);

Drawing tab

Drawing Section

 'Draw Sprite'
draw_sprite(sprite,subimage,x,y);

 'Draw Background'
draw_background(back,x,y) //single image.
draw_background_tiled(back,x,y); //tiled image.

 'Draw Text'
draw_text(x,y,string);

 'Draw Scaled Text'
draw_text_transformed(x,y,string,xscale,yscale,angle);

 'Draw Rectangle'
draw_rectangle(x1,y1,x2,y2,outline); //Outline is 1 or 0 for yes or no.

 'Horizontal Gradient'
draw_rectangle_color(x1,y1,x2,y2,col1,col2,col3,col4,outline);
//Set col1 and col4 to the left color. Set col2 and col3 to the second color.

 'Vertical Gradient'
draw_rectangle_color(x1,y1,x2,y2,col1,col2,col3,col4,outline);
//Set col1 and col2 to the top color. Set col3 and col4 to the bottom color.

 'Draw Ellipse'
draw_ellipse(x1,y1,x2,y2,outline); //outline is 1 or 0 for yes or no.

 'Gradient Ellipse'
draw_ellipse_color(x1,y1,x2,y2,col1,col2,outline);
//col1 is the color in the middle. col2 is the color at the boundary.

 'Draw Line'
draw_line(x1,y1,x2,y2);

 'Draw Arrow'
draw_arrow(x1,y1,x2,y2,size); //size is in pixels.

Settings Section

 'Set Color'
draw_set_color(col);
//See the manual for colors.
 'Set Font'
draw_set_font(font);
draw_set_halign(halign); //Can be set to fa_left, fa_center, fa_right.
draw_set_valign(valign); //Can be set to fa_top, fa_middle, fa_bottom.

 'Set Full Screen'
window_set_fullscreen(full); //Set to 0 for window, to 1 for full screen.

Other Section

 'Take Snapshot'
screen_save(filename);

 'Create Effect'
effect_create_below(kind,x,y,size,color) //Creates an effect of the given kind (see below) at the indicated position. size give the size as follows: 0 = small, 1 = medium, 2 = large. color indicates the color to be used. The effect is created below the instances, that is, at a depth of 100000.
effect_create_above(kind,x,y,size,color) //Similar to the previous function but this time the effect is created on top of the instances, that is, at a depth of -100000.

//The following are the effect kinds to use in the above statements...

ef_explosion	ef_star
ef_ring	ef_spark
ef_ellipse	ef_flare
ef_firework	ef_cloud
ef_smoke	ef_rain
ef_smokeup	ef_snow

'Action' Functions

Game Maker has built-in functions called 'Action' functions that can be used to reduce coding length. These functions are the direct coding for the Drag and Drop icons and use the same number and type of arguments as their drag and drop counterparts (for the most part). However, these codes are not in the Game Maker manual, nor do they show up in the reference window at the bottom of the coding box, so they do not have much support. They are placed here separately to avoid confusion for beginners and should only be used when they are fully understood.

Understand that many of these will make coding shorter and easier, while others will not; 'action_end_game();' is really not much better than 'game_end();', whereas, 'action_draw_life_images' is incredibly useful as there is no direct code for it.

Here is a complete list of all action codes, their arguments and some limited explanation of functionality. Codes that contain no arguments in between their brackets means that there are none to apply to the actions...

Move Tab

Move section

 'Move Fixed'
 'Move Free'
action_move(direction,speed);

 'Move Towards'
action_move_point(x,y,speed);

 'Speed Horizontal'
action_set_hspeed(value);

 'Speed Vertical'
action_set_vspeed(value);

 'Set Gravity'
action_set_gravity(direction,gravity);

 'Reverse Horizontal'
action_reverse_xdir();

 'Reverse Vertical'
action_reverse_ydir();

 'Set Friction'
action_set_friction(value);

Jump Section

 'Jump to Position'
action_move_to(x,y);

 'Jump to Start'
action_move_start();

 'Jump to Random'
action_move_random(hsnap,ysnap);

 'Align to grid'
action_snap(hsnap,vsnap);

 'Wrap Screen'
action_wrap(type); *//0=horizontal, 1=vertical, 2=both*

 'Move to Contact'
action_move_contact(direction,max,solid);
// for solid, 0=only solid, 1=all instances;

 'Bounce'
action_bounce(precise,against);

Paths Section

 'Set Path'
action_path(path0,speed,at end, relative);
//for at end 0=stop, 1=continue from start, 2=continue from here, 3=reverse
//for relative 0=relative, 1=absolute

 'End Path'
action_path_end();

 'Path Position'
action_path_position(position);

 'Path Speed'
action_path_speed(value);

Steps Section

 'Step Towards'
action_linear_step(x,y,speed,solid); *// for solid, 0=solid, 1=all instances*

 'Step Avoiding'
action_potential_step(x,y,stepsize,checkall);

Main1 Tab

Objects Section

 'Create Instance'
action_create_object(object,x,y);

 'Create Moving'
action_create_object_motion(object,x,y,speed,direction);

 'Create Random'
action_create_object_random(object0,object1,object2,object3,x,y);
//needs all four object slots to work properly

 'Change Instance'
action_change_object(object,perform events); *// for perform events 0=no, 1=yes*

 'Destroy Instance'
action_kill_object();

 'Destroy at Position'
action_kill_position(x,y);

Sprite Section

 'Change Sprite'
action_sprite_set(sprite,subimage,speed);

 'Transform Sprite'
action_sprite_transform(xscale,yscale,angle,mirror);
// for mirror, 0=no mirror, 1=mirror horizontal, 2=flip vertical, 3=mirror and flip

 'Color Sprite'
action_sprite_color(color,alpha);

Sounds Section

 'Play Sound'
action_sound(sound,loop); *// for loop, 1=true, 0=false*

 'Stop Sound'
action_end_sound(sound);

 'Check Sound'
action_if_sound(sound);

Rooms Section

 'Previous Room'
action_previous_room(kind); *// kind is transition kind*

 'Next Room'
action_next_room(kind); *// for kind is transition kind*

 'Restart Room'
action_current_room(kind); *// for kind is transition kind*

 'Different Room'
action_another_room(room,transition kind); *// for kind is transition kind*

 'Check Previous'
action_if_previous_room(){
//stuff here
}

 'Check Next'
action_if_next_room(){
//stuff here
}

Main 2 Tab

Timing Section

 'Set Alarm'
action_set_alarm(time,alarm);

 'Sleep'
action_sleep(milliseconds,redraw); *//for redraw, 1=true, 0=false*

 'Set Timeline'
action_set_timeline(timeline,position);

 'Set Timeline Position'
action_set_timeline_position(position);

Info Section

 'Display Message'
action_message(string);

 'Show Info'
action_show_info();

 'Show Video'
action_show_video(filename,windowed,looped);
//for windowed, 0=windowed, 1=fullscreen,
// for looped, 0=no, 1=yes

Game Section

 'Restart Game'
action_restart_game();

 'End Game'
action_end_game();

 'Save Game'
action_save_game(filename);

 'Load Game'
action_load_game(filename);

Resources Section

 'Replace Sprite'
action_replace_sprite(spite,filename,subimages);

 'Replace Sound'
action_replace_sound(sound,filename);

 'Replace Background'
action_replace_background(background,filename);

Questions Section

 'Check Empty'
action_if_empty(x,y,type){
//Stuff here
}
// for type, 0=only solid, 1=all instances

 'Check Collision'
action_if_collision(x,y,type) {
//Stuff here
}
// for type, 0=only solid, 1=all instances

 'Check Object'
action_if_object(object,x,y) {
//Stuff here
}

 'Test Instance Count'
// none available. Please use the code describe in the regular section above.

 'Test chance'
action_if_dice(sides){
//Stuff here
}

 'Check Question'
action_if_question(string); //returns 1 for yes and 0 for no

 'Test Expression'
// none available. Please use the code describe in the regular section above.

 'Check Mouse'
```
action_if_mouse(argument){
//Stuff here
}
// 0=none, 1=left, 2=right, 3=middle
```

 'Check Grid'
```
action_if_aligned(x,y){
//Stuff here
}
```

Other Section

 'Start Block'
```
{
```

 'End Block'
```
}
```
 'Else'
```
else
```

 'Exit Event'
```
exit;
```

 'Repeat'
```
repeat(argument);
```

 'Call Parent Event'
```
action_inherited();
```

Code Section

 'Execute Code'
```
// none
```

 'Execute Script'
```
action_execute_script(script,argument0,argument1,argument2,argument3,
argument4);
//needs all 5 argument slots to work properly
```

'Comment'
//

Variables Section

'Set Variable'
//none

'Test Variable'
```
action_if_variable(variable,number,operation){
//Stuff here
}
// for operation 0=equal to, 1=smaller than, 3=larger than
```

'Draw Variable'
```
action_draw_variable(variable,x,y);
```

Score Tab

Score Section

'Set Score'
```
action_set_score(value);
```

'Test Score'
```
action_if_score(score,operation){
//Stuff here
}
// for operation, 0=equal to, 1=smaller than, 3=larger than
```

'Draw Score'
```
action_draw_score(x,y,caption); //caption is a string before the score
```

'Show Highscore'
// Although a code exists in Game Maker to use 'Show Highscore' as an Action Function, it would appear to be unstable and, in tests, actually crashed the game with an unexpected error. If I had to guess, I believe it's not receiving enough information due to the font window that appears with the regular Drag and Drop,

which is not available with the Action Function. It's best to steer clear of it and use regular coding instead.

 'Clear Highscore'
action_highscore_clear();

Lives Section

 'Set Lives;
action_set_life(value);

 'Test Lives'
action_if_life(lives,argument){
//Stuff here
}
// 0=equal to, 1=smaller than, 3=larger than

 'Draw Lives'
action_draw_life(x,y,caption); *//caption is a string*

 'Draw Life Images'
action_draw_life_images(x,y,sprite);

Health Section

 'Set Health'
action_set_health(value);

 'Test Health'
action_if_health(health,argument){
//Stuff here
}
// 0=equal to, 1=smaller than, 3=larger than

 'Draw Health'
action_draw_health(x,y,x1,y,1,back color, bar color);

 'Score Caption'
action_set_caption(show score,score caption,show lives,lives caption,show health,health caption);

//show for each is 1 for true or 0 for false. Captions are in strings

Extra Tab

Particles Section

 'Create Part System'
action_partsyst_create(depth);

 'Destroy Part System'
action_partsyst_destroy();

 'Clear Part system'
action_partsyst_clear();

 'Create Particle'
action_parttype_create(type id,shape,sprite,min size,max size,size inrease);

 'Particle Color'
action_parttype_color(type id,shape,color1,color2,start alpha,end alpha);

 'Particle Life'
action_parttype_life(type id,min,max);

 'Particle Speed'
action_parttype_speed(type id,min speed,max speed,min dir,max dir,friction);

 'Particle Gravity'
action_parttype_gravity(typeid,amount,direction);

 'Particle Secondary'
action_parttype_secondary(type id,step type,step count,death type,death count);

 'Create Emitter'
action_partemit_create(sm,shape,xmin,xmax,ymin,ymax);

 'Destroy Emitter'
action_partemit_destroy(emitter);

 'Burst from Emitter
action_partemit_burst(emitter,type,number);

 'Stream from Emitter'
action_partemit_stream(emitter,type,number);

CD Section

//You must call the function cd_init(); before calling other Cd functions.

 'Play CD'
action_cd_play(start track, final track);

 'Stop CD'
action_cd_stop();

 'Pause CD'
action_cd_pause();

 'Resume CD'
action_cd_resume();

 'Check CD'
action_cd_present();
 'Check CD playing'
action_cd_playing();

Other Section

 'Set Cursor';
action_set_cursor(sprite,show cursor); *//show cursor 1=true, 0=false*

 'Open a Web Page'
action_webpage(filename);

Drawing Tab

Drawing Section

 'Draw Sprite'
```
action_draw_sprite(sprite,x,y,subimage);
```

 'Draw Background'
```
action_draw_background(background,x,y,tiled);  // 0=no,1= yes
```

 'Draw Text'
```
action_draw_text(text,x,y);
```

 'Draw Scaled Text'
```
action_draw_text_transformed(text,x,y,xscale,yscale,angle);
```

 'Draw Rectangle'
```
action_draw_rectangle(x,y,x1,y1,filled); // 0=filled, 1=outline
```

 'Horizontal Gradient'
```
action_draw_gradient_hor(x,y,x,1,y,1,color1,color2);
```

 'Vertical Gradient'
```
action_draw_gradient_vert(x,y,x1,y1,color1,color2);
```

 'Draw Ellipse'
```
action_draw_ellipse(x,y,x1,y,1,filled); // 0=filled, 1=outline
```
 'Gradient Ellipse'
```
action_draw_ellipse_gradient(x,y,x1,y,1,color1,color2);
```

 'Draw Line'
```
action_draw_line(x,y,x1,y1);
```

 'Draw Arrow'
```
action_draw_arrow(x,y,x1,y1,tip size);
```

Settings Section

 'Set Color'
```
action_color(color);
```

 'Set Font'
action_font(font name,align); *// 0=left aligned, 1=centered, 2=right aligned*

 'Set Full Screen'
action_fullscreen(type); *// 0=yes, 1=no*

Other Section

 'Take Snapshot'
action_snapshot(file name);

 'Create Effect'
action_effect(type,x,y,size,color,where) *// for where 0=below objects, 1=above objects*

Appendix A: Transition Kinds

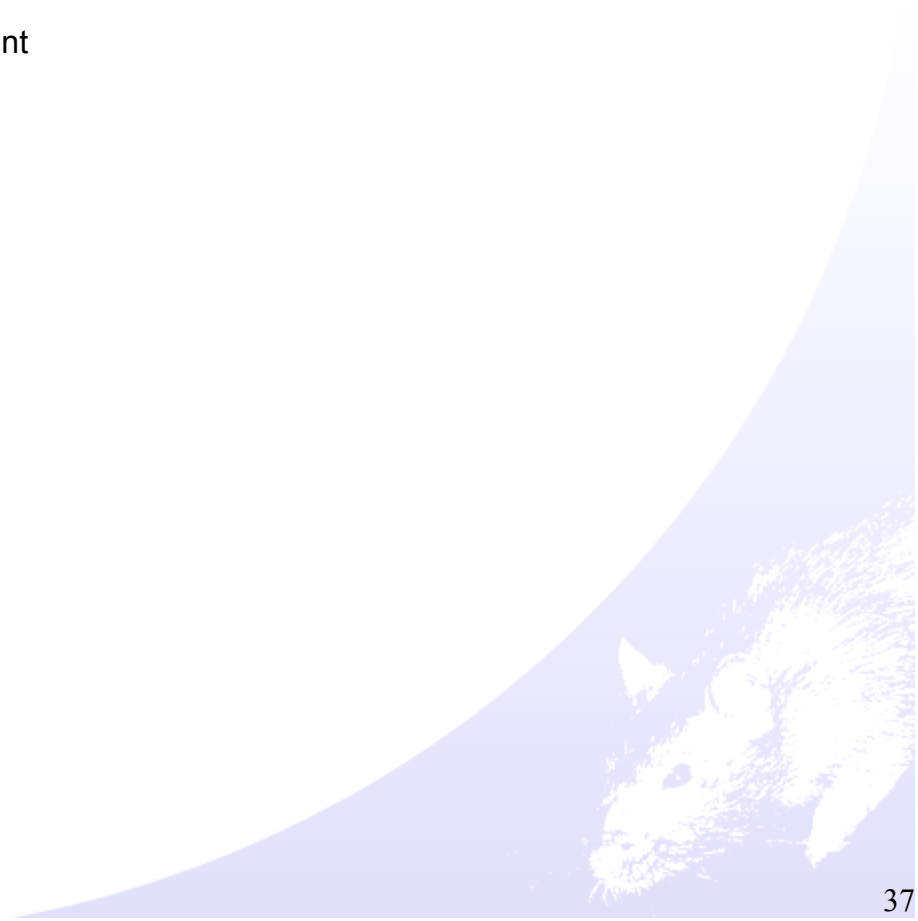
These are the values to use with room movement when it asks for Transition Kind...

0 = no effect
1 = Create from left
2 = Create from right
3 = Create from top
4 = Create from bottom
5 = Create from center
6 = Shift from left
7 = Shift from right
8 = Shift from top
9 = Shift from bottom
10 = Interlaced from left
11 = Interlaced from right
12 = Interlaced from top
13 = Interlaced from bottom
14 = Push from left
15 = Push from right
16 = Push from top
17 = Push from bottom
18 = Rotate to the left
19 = Rotate to the right
20 = Blend the rooms
21 = Fade out and in

Appendix B: Cursor Types

The following are the types of cursors to use when setting cursor

cr_default
cr_none
cr_arrow
cr_cross
cr_beam
cr_size_nesw
cr_size_ns
cr_size_nwse
cr_size_we
cr_uparrow
cr_hourglass
cr_drag
cr_nodrop
cr_hsplit
cr_vsplit
cr_multidrag
cr_sqlwait
cr_no
cr_appstart
cr_help
cr_handpoint
cr_size_all



Appendix C: Colors

The following are the default colors to be used when setting colors.

c_aqua
c_black
c_blue
c_dkgray
c_fuchsia
c_gray
c_green
c_lime
c_ltgray
c_maroon
c_navy
c_olive
c_orange
c_purple
c_red
c_silver
c_teal
c_white
c_yellow

Thanks to the Following

Mark Overmars for making Game Maker (of course)
http://www.gamemaker.nl/

New Game Studios Representative
For the original suggestion that lead to this.

And many thanks to everyone who PM'd fixing my many errors. I wish I had remembered to jot down all your names but I rarely think ahead like that (shoot). But you know who you are so….thanks!