GameMaker Studio Book
Tips & Tricks

Ben Tyers

ISBN-13: 978-1495947384
ISBN-10: 1495947386

DEDICATION

Many thanks to Mark Overmars, creator of the original GameMaker

CONTENTS

Contents

ACKNOWLEDGMENTS

Thanks To All These People, For Helping Me With This Book and Allowing Me To Re-Use Some Of Their Code.
Thanks also to the GMZ for answering my questions in a timely manner.

In Alphabetical Order

05GREL2
64 DIGITS
AMORBIS
CAHARKNESS
CPSGAMES
GMEXPERT
LEANDRO SACCOLETTO
ISMAVATAR
MANUEL777
NEHEMEK AMADOR
ROKAN
SIMON DONKERS
SMASH GAMES
TREVOR MADGE
T. WESTENDORP
WDALPHIN2
YELLOWAFTERLIFE
ZARNIWOOOP

Special Thanks to Alesia Buonomo For Proof Reading

Thanks to Alexander For The Technical Review

Advert System

A simple system to create and show your own adverts, great for directing players to your website or app store page.

You'll need to assign a sprite with 4 sub images for this example.

Create Event

```
alarm[0]=60;

ad_number=0;

total_adverts=4;;
```

Alarm[0] Event

```
ad_number++;
if ad_number==total_adverts
{
    ad_number=0;
}
alarm[0]=60;
```

Left Released Event

```
if ad_number=0 url_open("www.google.com");

if ad_number=1 url_open("www.bing.com");

if ad_number=2 url_open("www.gamemakerbook.com");

if ad_number=3 url_open("www.bbc.com");
```

Draw Event

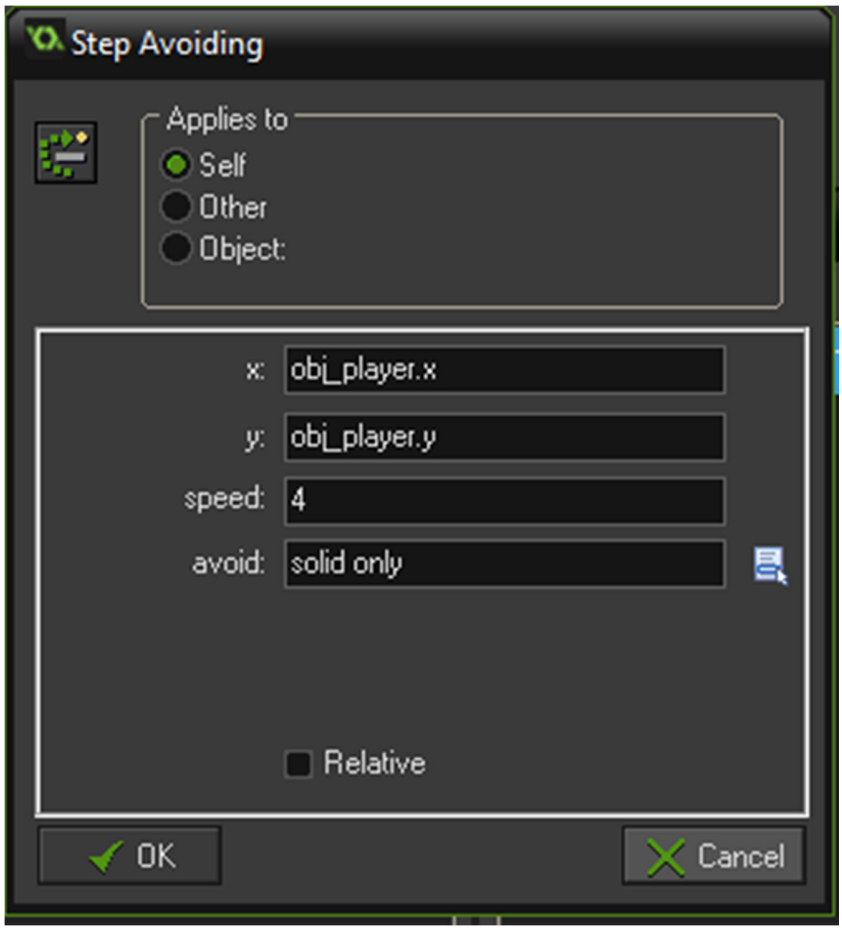
```
draw_sprite(adverts,ad_number,x,y);
```

Basic AI

A top down example of a very basic AI code.

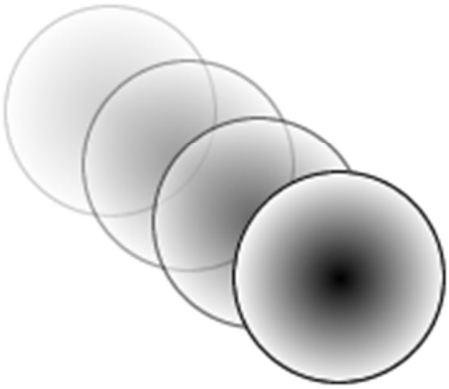
This example assumes you have a player object, obj_player and solid walls.

```
Step Event
```



Ball Bounce

This little code helps reduce likelihood of a bouncing ball becoming 'stuck'.



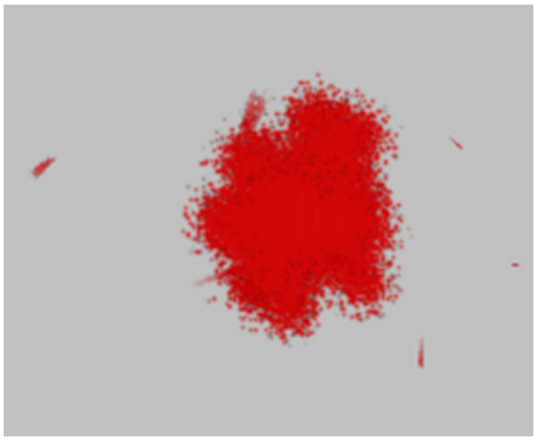
Just add the following the collision event of the ball with another object

```
Collision Event
```

```
move_bounce_all( true);

x=xprevious;

y=yprevious;
```

Blood Effect

This cool code draws a top down blood splatter effect



This assumes you have:

Object obj_blood with 3 or more small blood sub-images

Object obj_blood_2 with one larger image

obj_blood

Create Event

```
speed = random(4)+4; //Give out a random speed.
friction = 0.3; //Make the gore slow down from friction.
direction = random(360); //Random Direction
image_angle = random(360); //Random Angle
image_single = random(image_number-1); //Pick a random look.
image_xscale = random(1); //Random Size
image_yscale = image_xscale; //Restrain Proportions (aspect ratio)
fade = true; //Whether to fade the gore out or not.
```

Step Event

```
image_angle += speed/2; //Spin according to speed.
if fade=true{image_alpha -= 0.01;} //Fade out if relevant.
if image_alpha <0.1{instance_destroy();} //If completely faded, destroy.
```

Object obj_blood_2

Create Event

```
fade = true; //Whether to fade the gore out or not.;
```

Step Event

```
if fade=true{image_alpha -= 0.01;} //Fade out if relevant.
if image_alpha <0.1{instance_destroy();} //If completely faded, destroy.
```

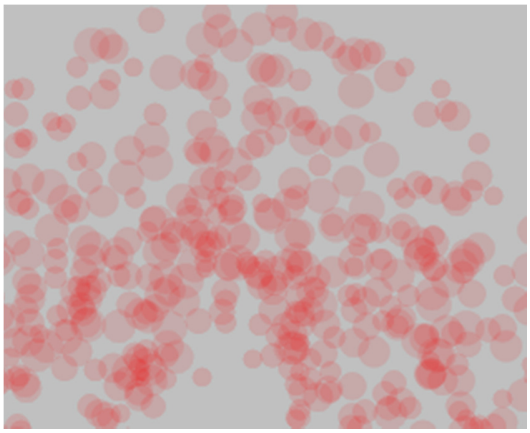
Object obj_spawn_blood

Moouse Global Left Button (or whatever trigger you are using

```
x = mouse_x

y = mouse_y

repeat (10) {instance_create(x,y,obj_blood)}

repeat (3) {instance_create(x,y,obj_blood_2)}
```

Blood Effect 2

Creates a blood explosion with blood on screen effect using particles



Create Event

```
globalvar flow;
```

Alarm [0] Event

```
 visible = true;//as needed
```

Script blood

```
  // example blood(50,50); or blood(x,y);
spotX = argument0;
spotY = argument1;

flow = part_system_create();
droplet = part_type_create();
spatter = part_type_create();
b_color = c_red;

part_type_shape(droplet,pt_shape_pixel)
part_type_size(droplet,0.10,0.10,0.05,0)
part_type_color1(droplet,b_color)
part_type_alpha2(droplet,1,0)
part_type_speed(droplet,1,6,0,0)
part_type_direction(droplet,-30,210,0,10)
part_type_gravity(droplet,0.2,270)
part_type_orientation(droplet,90,90,0,0,1)
part_type_life(droplet,30,40)
```

```
part_type_death(droplet, 1, spatter)

part_type_shape(spatter,pt_shape_disk)
part_type_size(spatter,0.3,0.3,.05,0)
part_type_scale(spatter,.35,.35)
part_type_color2(spatter,make_color_rgb(255, 0, 0), make_color_rgb(192, 0,
0))
part_type_alpha2(spatter,.15,0)
part_type_speed(spatter,0,0,0,0)
part_type_direction(spatter,0,0,0,0)
part_type_gravity(spatter,0,270)
part_type_life(spatter,50,60)

wound = part_emitter_create(flow);
part_system_depth(flow, -25);

part_emitter_region(flow,wound,spotX+10,spotX+22,spotY+10,spotY+22,ps_shape_
line ,ps_distr_gaussian);
effect_create_above(ef_explosion, spotX +16, spotY + 16, .5, c_red);
effect_create_above(ef_flare, spotX +16, spotY + 16, 25, c_orange);
obj_blood.visible=false;
part_emitter_burst(flow,wound,droplet,500);
obj_blood.alarm[0] = 120;
```

Example usage:

blood(x,y);

Change Size

You'll need two additional objects for this to work, obj_grow and obj_shrink

Step Event

```
if (instance_place(x,y,obj_grow))
{
     if (image_xscale < 2) //or whatever size you want
     {
          image_xscale += 0.5;
          image_yscale += 0.5;
     }
}
else if (instance_place(x,y,obj_shrink))
{
    if (image_xscale >0.5)//or whatever size you want
        {
            image_xscale -= 0.5;
            image_yscale -= 0.5;
        }
}
To destroy obj_grow / obj_shrink, but the following in their create event

if (instance_place(x,y,obj_player)) instance_destroy();
```

Cheat System

Allows user to enter a cheat code using the keyboard. Suitable for Windows Exports only.

Create Event

```
keyallowed=true;
```

Alarm[10] Event

```
keyallowed=true;
```

Step Event

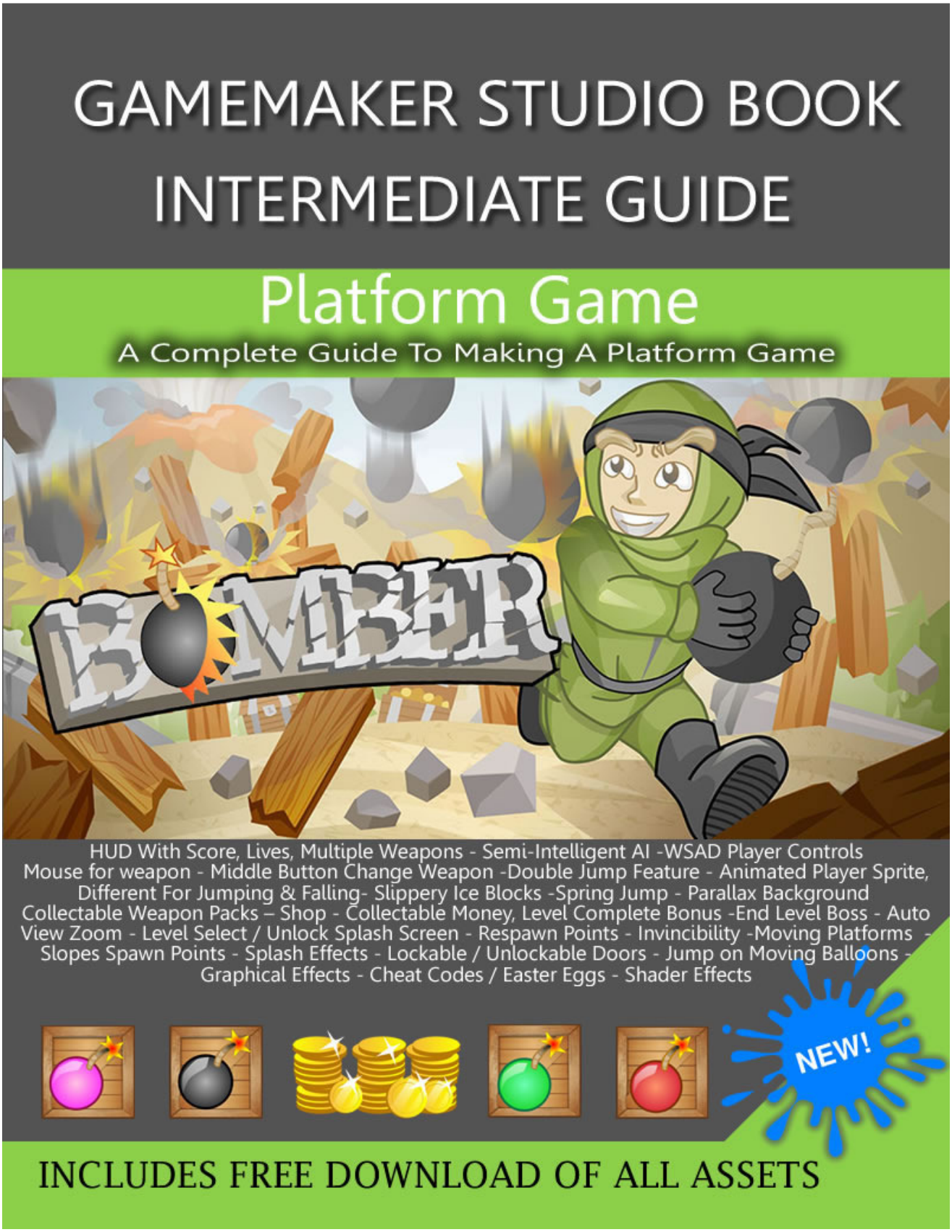
```
if keyallowed=true

{

if      keyboard_check(ord('A'))      {cheatstring+="A";      alarm[10]=5;
keyallowed=false;}

if      keyboard_check(ord('B'))      {cheatstring+="B";      alarm[10]=5;
keyallowed=false;}

if      keyboard_check(ord('C'))      {cheatstring+="C";      alarm[10]=5;
keyallowed=false;}

if      keyboard_check(ord('D'))      {cheatstring+="D";      alarm[10]=5;
keyallowed=false;}
```

```
if     keyboard_check(ord('E'))          {cheatstring+="E";     alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('F'))       {cheatstring+="F";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('G'))       {cheatstring+="G";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('H'))       {cheatstring+="H";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('I'))       {cheatstring+="I";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('J'))       {cheatstring+="J";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('K'))       {cheatstring+="K";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('L'))       {cheatstring+="L";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('M'))       {cheatstring+="M";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('N'))       {cheatstring+="N";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('O'))       {cheatstring+="O";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('P'))       {cheatstring+="P";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('Q'))       {cheatstring+="Q";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('R'))       {cheatstring+="R";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('S'))       {cheatstring+="S";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('T'))       {cheatstring+="T";       alarm[10]=5;
keyallowed=false;}

if       keyboard_check(ord('U'))       {cheatstring+="U";       alarm[10]=5;
keyallowed=false;}
```

```
if      keyboard_check(ord('V'))      {cheatstring+="V";      alarm[10]=5;
keyallowed=false;}

if      keyboard_check(ord('W'))      {cheatstring+="W";      alarm[10]=5;
keyallowed=false;}

if      keyboard_check(ord('X'))      {cheatstring+="X";      alarm[10]=5;
keyallowed=false;}

if      keyboard_check(ord('Y'))      {cheatstring+="Y";      alarm[10]=5;
keyallowed=false;}

if keyboard_check(ord('Z')) {cheatstring=""; alarm[10]=5; keyallowed=false;}

}

if cheatstring="SPEED"

{

    speedcheat=true;

    cheatstring="";

}
```

Draw Event [as needed]

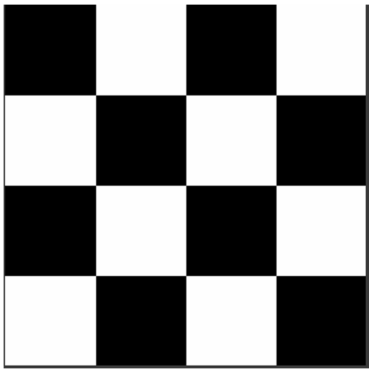
```
draw_text(x,y,cheatstring);

if speedcheat=true
{
     draw_text(100,100,"speed cheat active");
}
```



This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Check Point System

This simple code remembers the location of the last checkpoint the player touched and spawns there when dead.



Assumes you have an object obj_checkpoint with sprite set (like image above for example) . Also assumes sprite origin as center for both checkpoint and player object

Player Object

Create Event Example

```
lives=5;
checkpoint_x=50;
checkpoint_y=50;
x=checkpoint_x;
y=checkpoint_y;
```

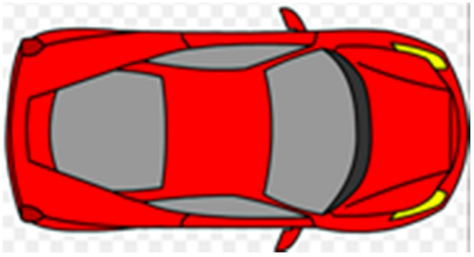
Step Event

```
x+=4*(keyboard_check(vk_right)-keyboard_check(vk_left));
y+=4*(keyboard_check(vk_down)-keyboard_check(vk_up));
if keyboard_check(ord('D'))// or your own trigger/death code
    {
        lives-=1;
        x=checkpoint_x;
        y=checkpoint_y;
    }
```

Collision with Object obj_checkpoint

```
checkpoint_x=other.x;
checkpoint_y=other.y;
```

Car Drift



Creates car control with drift effect.

You'll need a car sprite pointing right for this example.

Create Event

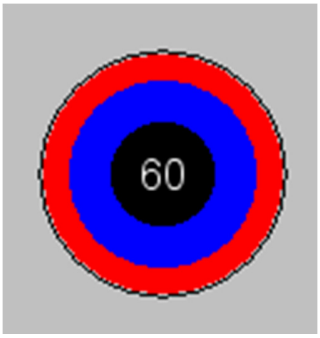
```
car_speed=0
direction=0
tire_rot=0
friction=1
drift=0
```

Step Event

```
if keyboard_check(ord("W")){if car_speed<7{car_speed+=.3;}}
if keyboard_check(ord("S")) {if car_speed>0{car_speed-=.5;}
else if car_speed>-3{car_speed-=.05}}
if ! keyboard_check(ord("S")) && ! keyboard_check(ord("W"))
{if car_speed<-.5 car_speed+=.5; else if car_speed>.5
car_speed-=.2;
else car_speed=0;
}
if keyboard_check(vk_space) {if abs(car_speed)>0.7
{
car_speed-=sign(car_speed)*0.7
}
Else car_speed = 0;
```

```
}
if keyboard_check(ord("D")){if tire_rot>-40 {tire_rot-=5;}}
if keyboard_check(ord("A"))
{if tire_rot<40
{tire_rot+=5}}
if ! keyboard_check(ord("A")) && ! keyboard_check(ord("D"))
    {if tire_rot<-5 {tire_rot+=5;}
else if tire_rot>5{tire_rot-=5;}
    else tire_rot=0}


if car_speed>0 image_angle+=car_speed/30*tire_rot;
else
if car_speed<0 image_angle+=car_speed/20*tire_rot;
if image_angle>360 image_angle=0;
if image_angle<0 image_angle=360;
friction=speed/10;
motion_add(image_angle,car_speed/6);

if tire_rot>30 && speed>6.5 drift=1
if tire_rot<-30 && speed>6.5 drift=1
if abs(((((image_angle-direction) mod 360)+540) mod 360)-180)<5
{
   drift=0
}
if speed<3 drift=0
if drift=0
{
   if car_speed>0 direction=image_angle;
}
```

Circular Health 2



This example assumes a health value between 0 and 100 inclusive.

Script: draw_circle_health2

```
//x,y,radius,value,colour1,colour2,outline)
//draw_circle_part_color(200,200,health,c_red,c_red);
//                        0   1  2        3     4
//draw background in colour argument4
draw_set_color(argument4);
draw_circle(argument0,argument1,185/4,0);
//draw propotion in colour argument 3
draw_set_color(argument3);
draw_circle(argument0,argument1,(85+argument2)/4,0);

//draw outlines and health value
draw_set_color(c_black);
draw_circle(argument0,argument1,(185/4)+1,1);
draw_circle(argument0,argument1,(85/4)-1,0);
draw_set_color(c_white);
draw_set_font(font_health);
draw_set_halign(fa_center);
draw_set_valign(fa_center);
draw_text(argument0,argument1,health);
```

Example Usage (in Draw Event):

```
//x,y,radius,value,colour1,colour2,outline)
//draw_circle_part_color(200,200,health,c_red,c_red);
//                        0   1  2        3     4
draw_circle_health2(200,200,health,c_blue,c_red);
```

Circular Text

Draws text in a moving circle.

Script: draw_text_circular

```
/*
draw_text_circular(x,y,string,xscale,yscale,speed,rad,fir angle,sec angle)
-------------------------------------------------------------------------------
----------------------------------------------
x - X position
y - Y position
string - The text you want to show
xscale - The width of the text
yscale - The height of the text
speed - The speed in which the text rotates. This must be a varible declared
in the step event
rad - The radius of the circle.
fir angle - The direction of the first character in the text.
sec angle - The direction of the angle relative to the first. Moves
clockwise. Set to 360 to make a full circle
-------------------------------------------------------------------------------
----------------------------------------------
other info:
This function sets the halign to fa_center and the valign to fa_top.
The basic idea is to seperate every character and give it a x/y position and
an angle value, then draw it
V.1.0
```

```
*/

var xx,yy,text,xs,ys,dd,length,dir,a,b,bb,tt,dir;
xx=argument0;
yy=argument1;
text=argument2;
argument2 = text;
xs=argument3;
ys=argument4;
dd=argument5
length=argument6;
dir=min(argument7,argument8);
dir2=max(argument7,argument8);
tt=string_length(text)
a=+1;
b=0;
bb=b;
repeat(tt) {pos[b]=string_char_at(text,a); a+=1; b+=1;}
b=0;
bb=b;
draw_set_valign(fa_top);
draw_set_halign(fa_center);
repeat(tt) {draw_text_transformed(xx+lengthdir_x(length,dir+dd),
yy+lengthdir_y(length,dir+dd),pos[bb],xs,ys,dir-90+dd); bb+=1; dir-=dir2/tt;
}
```

Example Usage:

Object obj_text

```
Create Event
```

```
ss=0;
```

```
Step Event
```

```
ss+=0.25;
```

```
Draw Event
```

```
draw_text_circular(200,200,"Example Circular Text",1,1,ss,90,90,180);
```

Clickable Text

Script Code draw_text_click

```
/// draw_text_click(x,y,string,margin,hover_color,hover_transparency);
var c;
c = draw_get_color();

var sx,sy,ex,ey;
sx = argument0-argument3;
sy = argument1-argument3;
ex = argument0+argument3+string_width(argument2);
ey = argument1+argument3+string_height(argument2);

var h;
h = (mouse_x>=sx && mouse_y>=sy && mouse_x<=ex && mouse_y<=ey);

draw_set_alpha(argument5);
draw_set_color(argument4);

if(h)
{
    draw_rectangle(sx,sy,ex,ey,false);
}

draw_set_alpha(1);
draw_set_color(c);
draw_text(argument0,argument1,argument2);

// Returns 1 if the text is currently pressed - change to
mouse_check_pressed for different affect.
return(h && mouse_check_button(mb_left));
```

Example Usage

```
if draw_text_click(16,16,"You Can Click Here!",4,c_red,0.5)=1
{
draw_text(50,50,"Clicked");
//or do something else
}
```

Countdown Bar

This example creates a counting down bar, great for visualizing temporary power ups.

You'll need two sprites for this spr_countdown_bar1 610x30 pixels (background) and spr_countdown_bar2 590x20 pixels.

Create Event

```
maxcount=600;

count=maxcount;
```

Step Event

```
count-=1;

 if(count<=0){

   //any code triggered by the ending of counting

   instance_destroy();

}
```

Draw Event

```
draw_sprite(spr_countdown_bar1,0,x,y);

draw_sprite_ext(spr_countdown_bar2,0,x+5,y+5,count/maxcount,1,0,c_white,1);
```

This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Countdown Clock

Create Event

```
// Set Your Time Start Time
seconds = 20; //Set The Starting Seconds
minutes = 1; //Set The Starting minutes
hours = 0; //Set The Starting hours
alarm[0]=room_speed; //Set alarm for one second (room_speed)
```

Alarm Event

```
seconds-=1;

if seconds < 0// checks if seconds is less than 0

{

    seconds = 59;//resest seconds

    minutes -=1;// take one off minutes

}

if minutes < 0 // check if minutes less than 0

{

    minutes = 59;// resiest minutes

    hours -=1;//take one of hours

}

alarm[0]=room_speed;//reset alarm

if seconds==0 && minutes==0 && hours==0 // check if countdown has reached
00:00:00
```

```
{
    // Do Something, for example@
    room_goto(room_game_over);


}
```

Draw Event

```
show_hours=string_repeat("0", 2-
string_length(string(hours)))+string(hours);//if single digit add a leading
0
show_minutes=string_repeat("0", 2-
string_length(string(minutes)))+string(minutes);//if single digit add a
leading 0
show_seconds=string_repeat("0", 2-
string_length(string(seconds)))+string(seconds);//if single digit add a
leading 0
draw_text(view_xview+5,view_yview+5,"Time: " + show_hours + ":" +
show_minutes + ":" + show_seconds);//draw variables
```

Day Night Engine

Changes background depending on time of day. Assumes you have a background set.

Create Event

```
global.daytime = 0;
hour = date_get_hour(date_current_datetime());
minu = date_get_minute(date_current_datetime());
sec = date_get_second(date_current_datetime());
fase = 0;
to = 1;
color = c_black;
col = 0;
red = 0;
alarm[0] = room_speed;
```

Step Event

```
sec +=30;//More than 30 to set the time too fast, or directly replace "sec"
for "minu"
if sec > 60
 {
  sec = 0;
  minu += 1;
 }
if minu > 59
 {
  minu = 0;
  hour += 1;
 }
if hour > 23
  hour = 0;
hh = (hour)+(minu/60);
to = -1;
if hour > 12
 {
  hh = (hour-12)+(minu/60);
  to = 1;
 }
fase = ((12-hh)/12)
if to = 1
  fase = 1-((12-hh)/12);
if to = -1
  fase*=1.6;

if hour = 18
  red = ((minu)/60)*255;
if hour = 19
  red = 255-(((minu)/60)*255);
if fase < 0.5           //Day
  global.daytime = 0;
else if fase < 0.64     //Morning
  global.daytime = 0.5;
else                    //Night
  global.daytime = 1;
```

Draw Event

```
if fase < 0.5
  col = make_color_rgb(255,255,255);
else if fase < 0.6
 {
  ff = (fase-0.5)*10;
  col = make_color_rgb(255-(ff*31),255-(ff*79),255-(ff*111));
 }
else if fase < 0.7
 {
  ff = (fase-0.6)*10;
  col = make_color_rgb(224-(ff*65),176-(ff*17),144+(ff*94));
 }
else
  col = make_color_rgb(159,159,238);
draw_set_color(c_white);


show_hours=string_repeat("0", 2-
string_length(string(hour)))+string(hour);//if single digit add a
show_minutes=string_repeat("0", 2-
string_length(string(minu)))+string(minu);//if single digit
show_seconds=string_repeat("0", 2-
string_length(string(sec)))+string(sec);//if single digit
draw_text(view_xview+5,view_yview+5,"Time: " + show_hours + ":" +
show_minutes + ":" + show_seconds);//draw variables

draw_set_color(col);
background_blend = col;
```

Disappearing Platform

A simple disappearing platform.

You'll need to assign a sprite to your object.

Create Event

```
image_alpha=1;
disappear_speed=0.01;//how quicly to disappear
```

Step Event

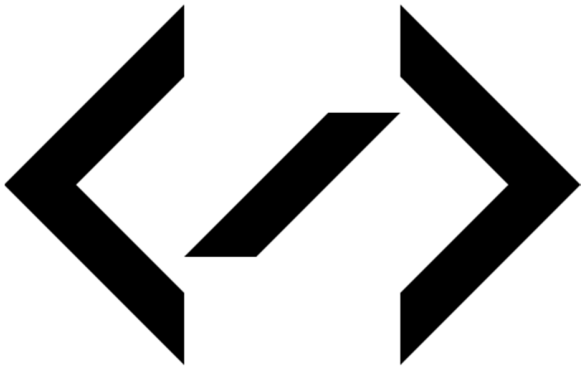
```
if image_alpha<0 instance_destroy();
```

Collision With Object Player Event

```
image_alpha-=dissapear_speed;
```

Set `disappear_speed=0.01;` to `disappear_speed=1`; so object disappears after one collision.

Do-until



Basic explanation of the structure “do-until”.

That is the basic structure of “do-until”:

```
do{
  code 1;
}until( statement 1);
```

First, the code 1 is executed. After the statement 1 is checked. If it is true, the loop ends. But if it is false, the code 1 is executed again and this process continue until the statement 1. The code 1 is always executed at least once.

First example:

```
do{
  y+=1;
}until( not place_free(x,y) or y>200);
```

In this example, a object is shifted by 1 in the y-direction until it collides with a solid instance or it’s position is higher than 200.

Second example:

```
do{

  image_alpha-=0.05;

}until(image_alpha<=0);
```

This is a simple fade effect for an object, remember to destroy it when alpha == 0.

Dragging Object

Allows click and drag of objects

Create

```
dragging=true;
```

Step Event

```
if(dragging==true)
{
    with(self)
    {
        x+=mouse_x-x;
        y+=mouse_y-y;
    }
}
```

Left Pressed Event

```
instance=instance_position(mouse_x,mouse_y,self);
dragging=true;
```

Global Left Released Button

```
dragging=false;
```

Draw Array Contents in Boxes

Useful if you wish to check values for debugging, or if required to draw them in game.

0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9
0	2	4	6	8	10	12	14	16	18
0	3	6	9	12	15	18	21	24	27
0	4	8	12	16	20	24	28	32	36
0	5	10	15	20	25	30	35	40	45
0	6	12	18	24	30	36	42	48	54
0	7	14	21	28	35	42	49	56	63
0	8	16	24	32	40	48	56	64	72
0	9	18	27	36	45	54	63	72	81

Showing contents of array drawn on the screen.

```
 Create Event (if needed, provided here for example purposes).
```

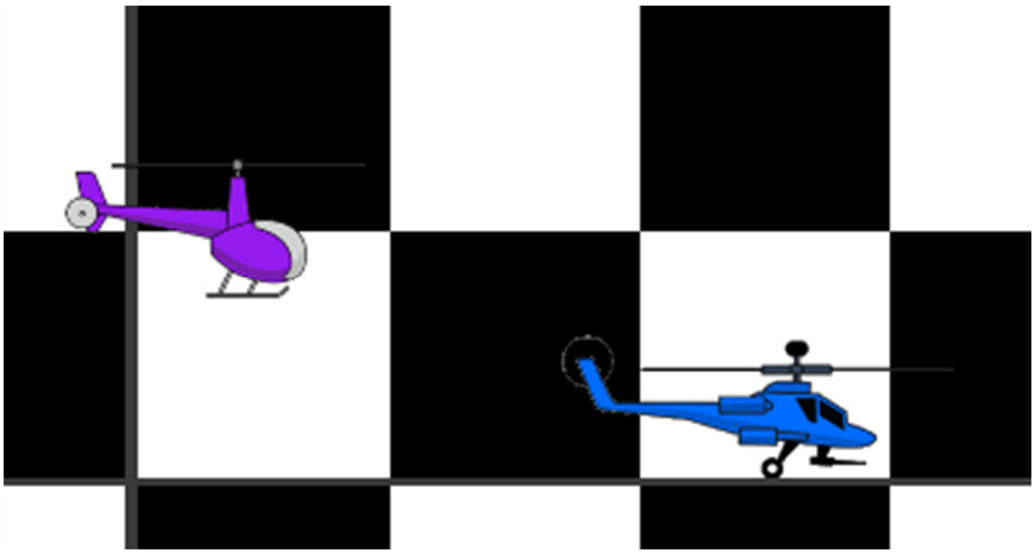
```
var i,j;
for (i = 0; i < 10; i++)
{
    for (j = 0;j < 10; j++)
    {
        array[i,j] = i*j;
    }
}
```

```
 Draw Event
```

```
var i,j;
for i= 0; I < 10; i++)
{
    for (j= 0; j < 10; j++)
    {
           cellsize=32
           xpos=i*cellsize
           ypos=j*cellsize
           border=8
           draw_rectangle(border+xpos,border+ypos,
           border+xpos+cellsize, border+ypos+cellsize,2);
           draw_text(border+xpos+5,border+ypos+12,array[i,j]);
    }
}
```

Dual View

Allows you to keep two objects in the view by zooming in and out.



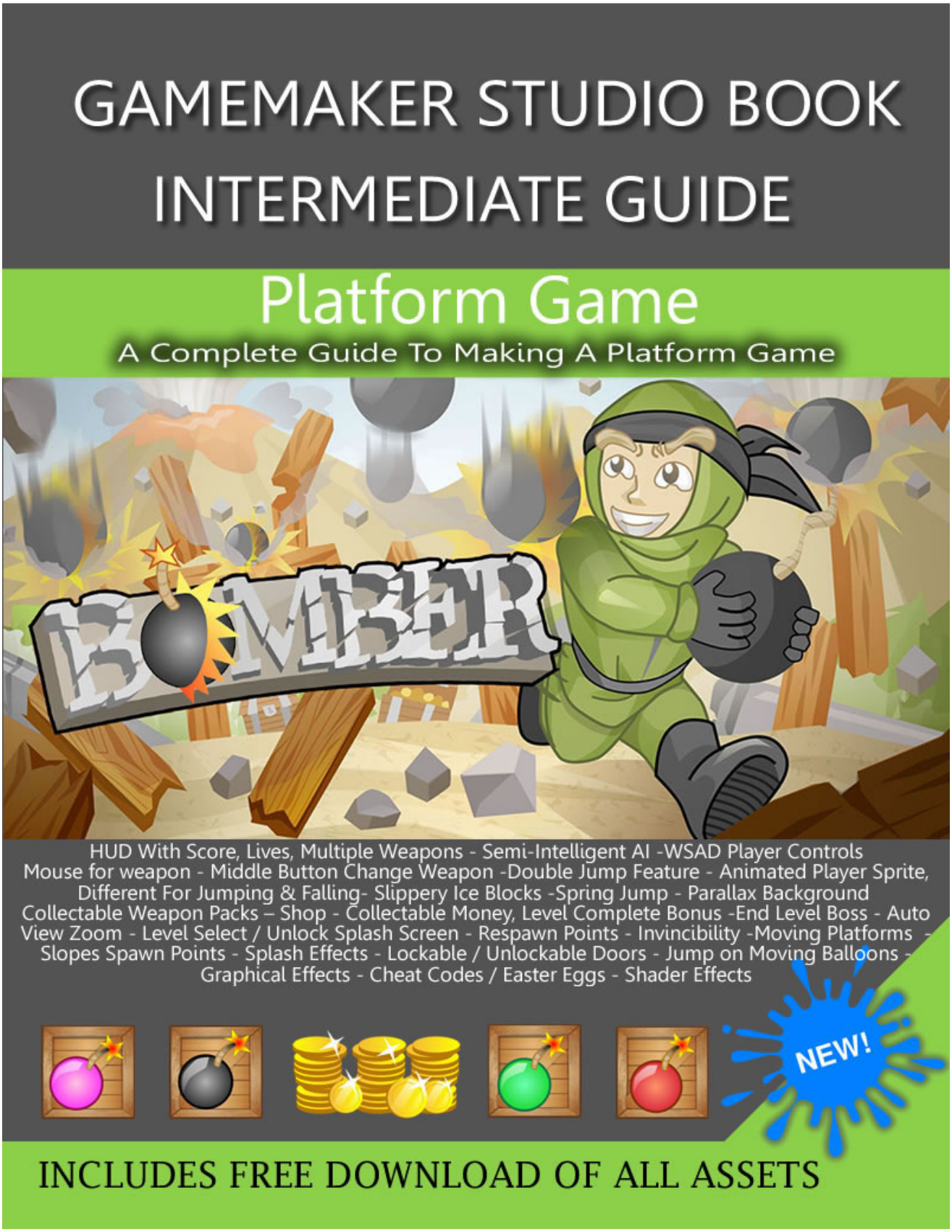
This example assumes you have two objects, obj_player1 and obj_player2. You will also need to create a view in your room, for example view[0] width and height of 400 and check the enable views checkbox.

```
Script: view_control
```

```
var o1, o2, x1, x2, y1, y2, vw, vh, vb, vscale;
o1 = argument0; x1 = o1.x; y1 = o1.y
o2 = argument1; x2 = o2.x; y2 = o2.y
vb = argument2; vw = view_wport; vh = view_hport;
vscale = max(1, abs(x2 - x1) / (vw - vb * 2), abs(y2 - y1) / (vh - vb * 2))
view_wview = vscale * vw
view_hview = vscale * vh
view_xview = (x1 + x2 - view_wview) / 2
view_yview = (y1 + y2 - view_hview) / 2
```

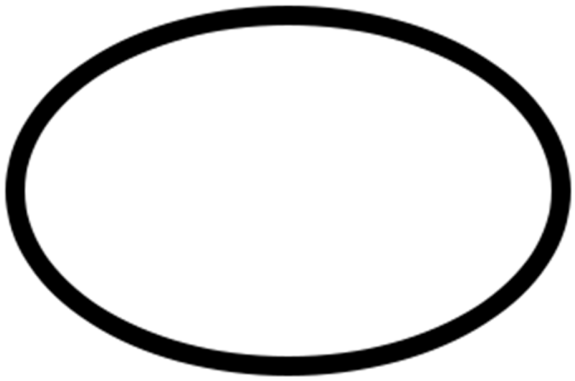
```
Example usage:
```

```
Place the following in the step event of a control object:
view_control(obj_player1, obj_player2, 150);// where 150 is room border from
object
```



This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Ellipse Movement



This code shows how to make a elliptical pattern .

This code just a basic concept of Analytic geometry to make a object moves counterclockwise in a ellipse with parameters rx and ry, centered in the point (xc,yc), with angular speed angle_speed. In the case r1=r2, the case reduces to a circle.

```
Create Event
```

```
angle=0;
rx=16;
ry=24;
angle_speed=0.05*pi;
xc=xstart;
yc=ystart;
```

```
End Step Event
```

```
x=xc+rx*cos(angle);
y=yc-ry*sin(angle);
angle+=angle_speed;
if(angle>2*pi) angle-=2*pi;
else if(angle<0) angle+=2*pi;
```

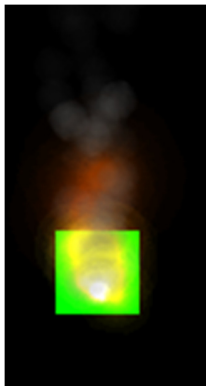
If you use a negative value of angle_speed, the motion will be clockwise. If yo want a object move in ellipse around a object 2 named “target”, just just replace xc and yc in the end step event for target.x and target.y.

Plus, you can make the elippse move to another direction if you move a elipse’s center. For example if you want move it to right by 2 pixels every step, just add it:

```
End Step Event
```

```
xc+=2;
```

Fire Effect



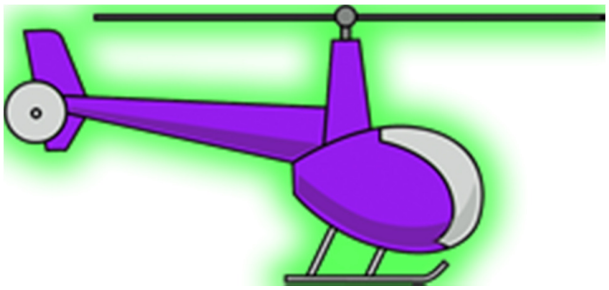
You'll need to assign a sprite to your object.

```
Script fire
 //example fire(x,y);
spotX = argument0;
spotY = argument1;

flow = part_system_create();
flame = part_type_create();
smoke = part_type_create();

part_type_shape(flame,pt_shape_flare) // the fire
part_type_size(flame,0.5,0.8,0.30,0)
part_type_scale(flame,0.10,0.10)
part_type_color3(flame,c_white,c_yellow,c_red)

part_type_alpha3(flame,1,0.70,.30)
part_type_speed(flame,0.20,.50,0,0)
part_type_direction(flame,0,359,0,20)
part_type_gravity(flame,0.10,90)
part_type_orientation(flame,0,180,0,0,1)
```

```
part_type_blend(flame,1)
part_type_life(flame,1,40)

part_type_shape(smoke,pt_shape_smoke) // where there's fire, there's smoke
part_type_size(smoke,1,1,0,0)
part_type_scale(smoke,0.25,0.25)
part_type_color2(smoke,c_gray, c_black)
part_type_alpha3(smoke,0.30,0.2,0.1)
part_type_speed(smoke,0.25,0.25,0,0)
part_type_direction(smoke,0,359,0,0)
part_type_gravity(smoke,0.10,90)
part_type_orientation(smoke,0,359,0,1,1)

part_type_blend(smoke,0)
part_type_life(smoke,60,60)

part_type_death(flame,1,smoke)

wound = part_emitter_create(flow);
part_system_depth(flow, -25);

part_emitter_region(flow,wound,spotX+8,spotX+24,spotY+16,spotY+32,ps_shape
_line ,ps_distr_gaussian);
part_emitter_stream(flow,wound,flame,1);
```

Flashing Sprite

A simple method of making a flashing sprite.



You'll need to assign a sprite to your object.

Create Event

```
flashon=false;

flashing=false;

flashingspeed=5;

flashinglength=100;
```

Step Event

```
if flashing==true
{
    flashingspeed++;
    if flashingspeed mod 5=0
        {
            flashon=true;
        }
    else
        {
            flashon=false;
        }
else
{
    flashon=false;
}
```

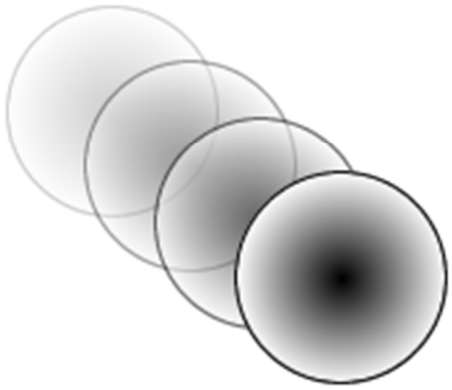
```
if flashingspeed==flashinglength
{
    flashing=false;
    flashon=false;
    flashingspeed=0;
}
```

Draw Event

```
if flashon==false

{

image_blend=c_white

}

else

{

image_blend=c_yellow;

}

draw_self();
```

This will make the sprite flash when flashing=true for the time of flashinglength at the speed flashing speed.

Following Object



This example assumes you have two objects with sprites assigned, obj_player and obj_follow.

Object obj_player

Step Event

```
x+=4*(keyboard_check(vk_right)-keyboard_check(vk_left));
y+=4*(keyboard_check(vk_down)-keyboard_check(vk_up));
```

Object obj_follow

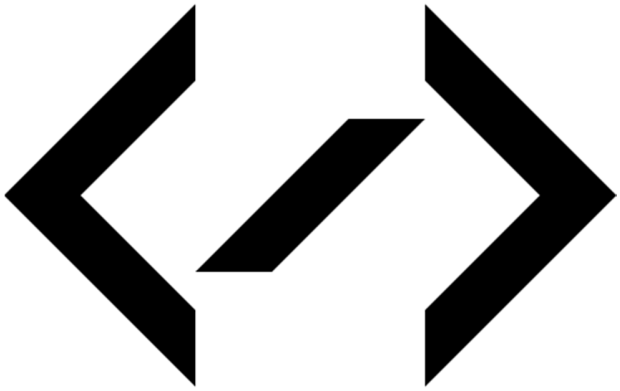
Create Event

```
point=obj_player;
```

Step Event

```
dir = point_direction( x, y, point.x, point.y );
x = point.x - lengthdir_x( 32, dir );
y = point.y - lengthdir_y( 32, dir );
```

For



That is the basic structure of “for”:

```
for(statement 1; condition ; statement 2)
{
  code 1;
}
```

First is executed the statement 1 (normaly initialize a variable) and is check the condition, if the condition is false, the loop ends. But if it is true, the code 1 is executed. After that, the stament 3 (normaly is a variable increment or anything that can turn the loop main condition false).

Let’s show a simple example:

```
var i;
for(i=0;i<9;i+=1)
{
  instance_create(12+24*i,36,obj_ball);
}
```

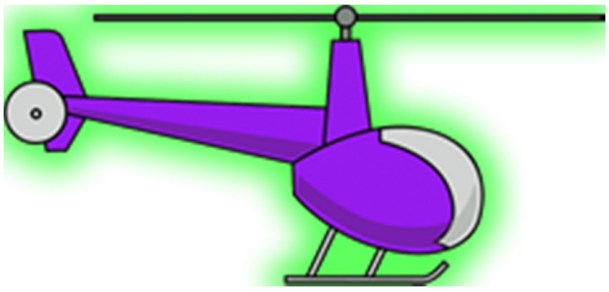
In this example we created 10 obj_ball at y=36 , with a distance of 24 pixels. You can notice that in this way is more easy than create use the function instance_create 10 times.

Another example:

```
for(times=0;x<9;x=irandom(12))
{
  times+=1;
}
```

This example counts how many tries took to obtain a number x equal or higher than 9 from 0 to 12.

Glowing Sprite



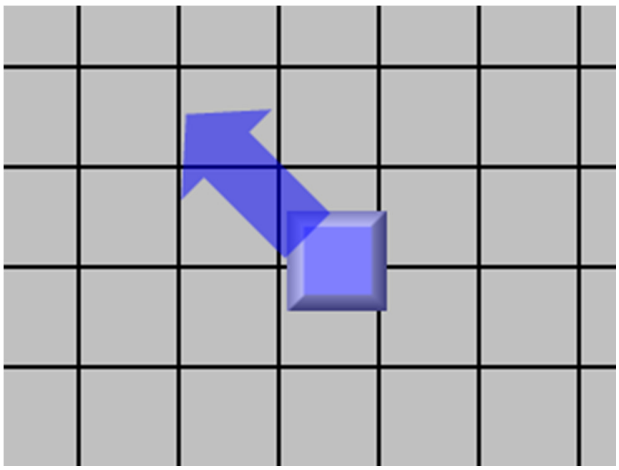
This example creates a glowing border around a transparent backgrounded sprite.

Script: draw_glow

```
//draw_glow(border,sprite,subimage,xpos,ypos,color);

if glow=true
{
    border=argument0;
    draw_set_blend_mode(bm_add);
    draw_sprite_ext(argument1,argument2,argument3-border,argument4-
border,1,1,0,argument5,0.8);
    draw_sprite_ext(argument1,argument2,argument3-
border,argument4+border,1,1,0,argument5,0.8);
    draw_sprite_ext(argument1,argument2,argument3+border,argument4-
border,1,1,0,argument5,0.8);

draw_sprite_ext(argument1,argument2,argument3+border,argument4+border,1,1,0,
argument5,0.8);
    draw_set_blend_mode(bm_normal);
}
```

This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Grid Slowly Move



This code snaps an object to a grid

Script: move_slowly

```
/*argument0=x pos, argument1=y pos,argument2=targetx
argument3=targety, argument4=distance, argument5=speed*/
if point_distance(argument0, argument1, argument2, argument3) >
argument4 //if the mouse is outside a certain reach/5 of the object
{
move_towards_point(argument2, argument3, argument5);
//move towards mouse_x, mouse_y in the speed of 5
}
else
{
speed=0;
move_snap(grid_size,grid_size);
}
```

Create Event

```
grid_size=50;//required
```

Step Event

```
//example to make move
target_x=floor(mouse_x/grid_size)*grid_size;
target_y=floor(mouse_y/grid_size)*grid_size;
snap_distance=3;
move_speed=5;
Eample
        slowly_move(x,y,target_x,target_y,snap_distance,move_speed);
```

Health As Bars

This little bit of code will draw your health as bars.

Health As Bars

Create Event

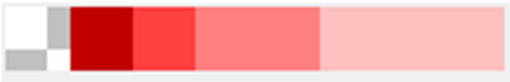
```
///Draw Health as Bars

var number_of_bars, healthbar, width, height, xpos,ypos, gap;

healthbar=health div 10;//change 10 to value you wish each bar to represent

width=20;//width of each bar

height=20;//height of each bar

xpos=100;//change this to change draw location

ypos=100;//change this to change draw location

gap=5;
```

Draw Event

```
///Draw Health as Bars

draw_set_color(c_red);

for (number_of_bars = 0; number_of_bars < healthbar; number_of_bars +=1;)

{

     draw_set_color(c_red);

     draw_rectangle(xpos+width*number_of_bars+(number_of_bars*gap),  0+ypos,
     xpos+width*number_of_bars+width+(number_of_bars*gap),           height+
     ypos,false)//false fills the rectangle

}
```

Heat Seeking Missile / Bullet Example

This example assumes have a bullet sprite, 16x2 pixels, example below:



Also assumes you have an enemy parent object, obj_enemy.

obj_bullet

Create Event

```
target = instance_nearest(x, y, obj_enemy);

speed=5;

alarm[0]=50;

alarm[0] = 60; // Life of missile targetting

friction = -0.1;

range=600;
```

Alarm 0 Event

```
target = noone;
```

Step Event

```
target = noone;

if (instance_exists(target)) {

    diff = angle_diff(point_direction(x, y, target.x, target.y),
direction);

    direction += sign(diff) * min(abs(diff), 4);

}

if (distance_to_object(obj_enemy)>range)

{instance_destroy();}
```

Jet Pack

A simple jet pack engine. Add this to your player object, with sprite.

Create Event

```
gravity_direction=270;
gravity=0.1;
```

Step Event

```
if keyboard_check(ord('W'))
{
    effect_create_above(ef_spark, x, y+16, 1, c_green);
    vspeed-=0.3;
}
if keyboard_check(ord('A'))
{
    hspeed-=0.3;
}
    if keyboard_check(ord('D'))
{
    hspeed+=0.3;
}
    if vspeed<0
{
    effect_create_above(ef_spark, x, y+16, 0.5, c_green);
}
```

Keep Object In Room

This bit of code keeps the object in the room, stopping it from leaving by the sides or top and bottom.

This code assumes sprite origin set as center. You can use this code if you views present.

End Step Event

```
if x<(0+sprite_width/2) x=sprite_width/2;
if y<(0+sprite_height/2) y=sprite_height/2;
if x>(room_width-sprite_width/2) x=room_width-sprite_width/2;
if y>(room_height-sprite_height/2) y=room_height-sprite_height/2;
```

Key and Door

This example assumes you have two objects with sprites assigned, obj_red_door and obj_red_key.

Object Player

Create Event

```
red_key=false;
```

Step Event

```
x+=4*(keyboard_check(vk_right)-keyboard_check(vk_left));
y+=4*(keyboard_check(vk_down)-keyboard_check(vk_up));
```

Collision With obj_red_door

```
if red_key=false
{
    x=xprevious;
    y=yprevious;
}
if red_key=true
{
    with(other){instance_destroy();}
}
```

Collision With obj_red_key

```
red_key=true;
with(other){instance_destroy();}
```

Keyboard On Screen

This example creates a room with a keyboard that can be used to enter text and numbers

This example required sprites, each with 4 sub images as shown below:

spr_button:

spr_enter:

spr_space:

Object Button

Create Event

```
activated = true;
letter = "A";
image_speed = 0;
image_index = activated;
global.word="";
draw_sprite_ext(argument1,argument2,argument3+border,argument4+border,1,1,0,
argument5,0.8);
    draw_set_blend_mode(bm_normal);
}
```

Step Event

```
if(activated){
    //if the mouse hovers above the button
    if(mouse_x>=x && mouse_x<x+sprite_width && mouse_y>=y &&
mouse_y<y+sprite_height){
        //if the mouse left button is down
        if(mouse_check_button(mb_left)){
            image_index = 3;
        }else{
            image_index = 2;
        }
        //if the button is released
        if(mouse_check_button_released(mb_left)){

            global.word+=letter;
                   }
                   }
                   }
```

Draw Event

```
draw_self();
draw_set_font(fnt_button);
draw_set_halign(fa_center);
draw_text(x+16, y+4, letter);
draw_set_halign(fa_left);
draw_text(10,10,global.word);
```

Object obj_enter

Create Event

```
activated = true;

image_speed = 0;

image_index = activated;
```

Step Event

```
if(activated){

    //if the mouse hovers above the button

    if(mouse_x>=x && mouse_x<x+sprite_width && mouse_y>=y &&
mouse_y<y+sprite_height){

        //if the mouse left button is down

        if(mouse_check_button(mb_left)){

            image_index = 3;

        }else{

            image_index = 2;

        }

        //if the button is released

        if(mouse_check_button_released(mb_left)){

            activated = false;//deactivate button

            room_goto(room_start);//Or whatever you want

                }

                }

                }
```

Object obj_space

Create Event

```
activated = true;

image_speed = 0;

image_index = activated;
```

Step Event

```
if(activated){

    //if the mouse hovers above the button

    if(mouse_x>=x && mouse_x<x+sprite_width && mouse_y>=y &&
mouse_y<y+sprite_height){

        //if the mouse left button is down

        if(mouse_check_button(mb_left)){

            image_index = 3;

        }else{

            image_index = 2;

        }

        //if the button is released

        if(mouse_check_button_released(mb_left)){

            activated = false;//deactivate button

            global.word+=" ";

                    }

                    }

                    }
```

Object obj_set_up_keyboard

Create Event

```
//creating all the buttons
//numbers
var alphabet = "1234567890";
for(i = 0; i<10; i++){
    var bt = instance_create(56+i*32+(8*i), 40, obj_button);
    bt.letter = string_char_at(alphabet, i+1);
}
//top row
var alphabet = "QWERTYUIOP";
for(i = 0; i<10; i++){
    var bt = instance_create(64+i*32+(8*i), 80, obj_button);
    bt.letter = string_char_at(alphabet, i+1);
}
//middle row
var alphabet = "ASDFGHJKL";
for(i = 0; i<9; i++){
    var bt = instance_create(72+i*32+(8*i), 120, obj_button);
    bt.letter = string_char_at(alphabet, i+1);
}
//BOTTOM row
var alphabet = "ZXCVBNM";
for(i = 0; i<7; i++){
```

```
    var bt = instance_create(88+i*32+(8*i), 160, obj_button);
    bt.letter = string_char_at(alphabet, i+1);
}
//Special Characters
var alphabet = "@-_";
for(i = 0; i<3; i++){
    var bt = instance_create(120+i*32+(8*i), 200, obj_button);
    bt.letter = string_char_at(alphabet, i+1);
}
instance_create(240,200,obj_space);
instance_create(432,120,obj_enter);
```

This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Keyboard Simple Movement Code – Multi-Directional

Detects key presses and makes an object move. This code will allow diagonal movement.

Code

```
if keyboard_check(ord('A'))
   {
   hspeed = -5;
   }
if keyboard_check(ord('D'))
    {
    hspeed = 5;
    }
if keyboard_check(ord('W'))
    {
    vspeed = -5;
    }
if keyboard_check(ord('S'))
    {
    vspeed = 5;
    }
if keyboard_check(vk_nokey)
   {
    vspeed = 0;
    hspeed = 0;
    }
```

You could also do this using:

```
if keyboard_check(vk_left). . .
if keyboard_check(vk_right). . .
if keyboard_check(vk_up). . .
if keyboard_check(vk_down). . .
```

If you want just up/down/left/right and no diagonal movement use this:

```
x+=4*(keyboard_check(vk_right)-keyboard_check(vk_left));
y+=4*(keyboard_check(vk_down)-keyboard_check(vk_up));
```

Laser Effect

Script draw_laser

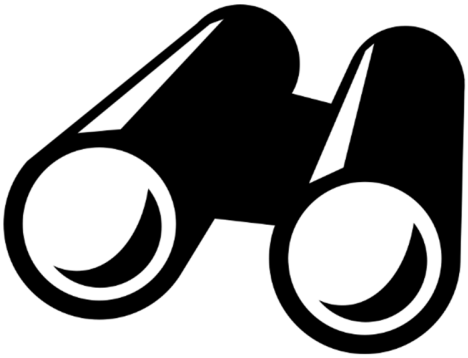
```
 draw_set_color(make_color_rgb(irandom(255),irandom(255),irandom(255)));
draw_line_width(argument0, argument1, argument2, argument3, 5);
draw_set_color(c_lime);
draw_line(argument0+1,argument1+1,argument2,argument3);
draw_line(argument0+1,argument1-1,argument2,argument3);
draw_line(argument0-1,argument1+1,argument2,argument3);
draw_line(argument0-1,argument1-1,argument2,argument3);
draw_line(argument0,argument1,argument2,argument3);
effect_create_above(ef_spark, argument2, argument3, 1,
choose(c_red,c_orange));
```

Draw Event Example

```
draw_laser(x,y,mouse_x,mouse_y);
```

Line of Sight

This nifty little code checks wether an object is hidden from view or not.



This assumes you have three objects obj_player, obj_wall, obj_enemy. This example looks for a line of site from enemy to player. This example assumes sprite origins as center.

Code for object obj_enemy

Create Event

```
hidding=false;
```

Step Event

```
if collision_line(x,y,obj_player.x,obj_player.y,obj_wall,true,true) {
    draw_line(x,y,obj_player.x,obj_player.y);
    hidding=true;
}
else
{
     hidding=false;
}
```

Draw Event

```
draw_self();

if hidding

{

    draw_text(x,y,"Hidden");

}

else

{

    draw_text(x,y,"Can See");

    draw_line(x,y,obj_player.x,obj_player.y);//shows line - great for
testing
```

Load / Save System

This code allows you to save / load variables.

To save:

```
ini_open( "savedata.ini" );
ini_write_real( "level_section", "level", global.level );
ini_write_real( "distance_section", "distance", global.distance );
ini_close();
```

This will store the current values of global.level and global.distance. If the values were 3 and 287, then the created ini file would look like:

[level_section]

level=3

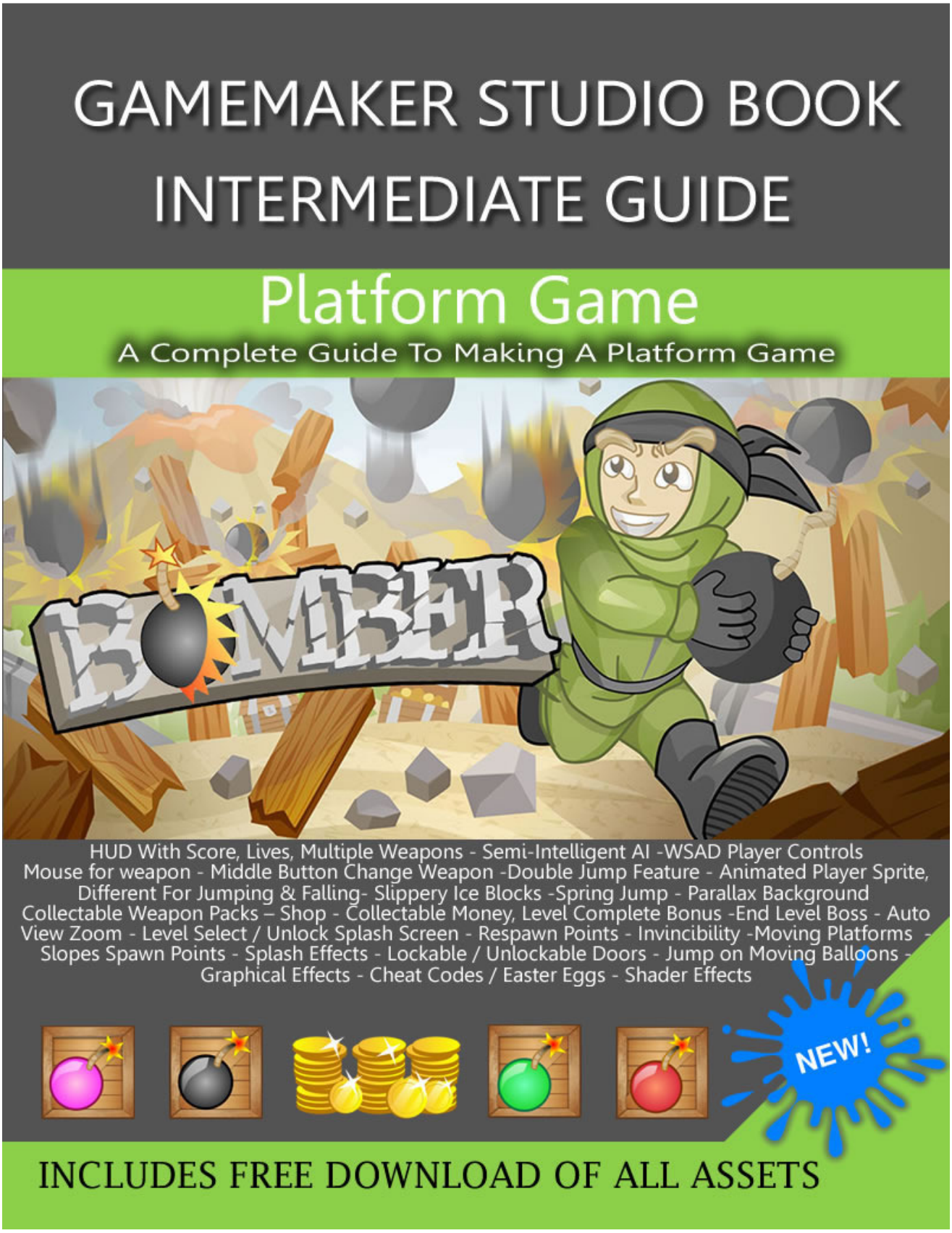
[distance_section]

distance=287

To load:

```
ini_open("savedata.ini");
global.level = ini_read_real("level_section", "level", 1);
global.distance = ini_read_real("distance_section", "distance", 0);
ini_close();
```

This would load the values stored in the ini file. If no file exists it will use the default values provided (in this case the numbers 1 and 0 after the last comma).



This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Local Time - Real Time Clock Example

Create Event

```
// use code below, or load data from saved file
seconds = 0;
minutes = 0;
hours = 0;
alarm[0]=room_speed;
```

Alarm Event

```
seconds++;
if seconds > 59
{
    seconds = 0;
    minutes +=1  ;
}
if minutes > 59
{
    minutes = 0;
    hours +=1;
}
alarm[0]=room_speed;
```

Draw Event

```
show_hours=string_repeat("0", 2-string_length(string(hours)))
+string(hours);
show_minutes=string_repeat("0", 2-string_length(string(minutes)))
+string(minutes);
show_seconds=string_repeat("0", 2-string_length(string(seconds)))
+string(seconds);
```

```
draw_text(view_xview+5,view_yview+5,"Time: " + show_hours + ":" + show_minutes + ":" + show_seconds);
```

Marquee Effect

This example creates scrolling text.

Example of text with border

Create Event

```
var position,width,text,last,first;
position = 1;
width = 20;
alarm = 25;
text = "Scrolling Text Example  ";
```

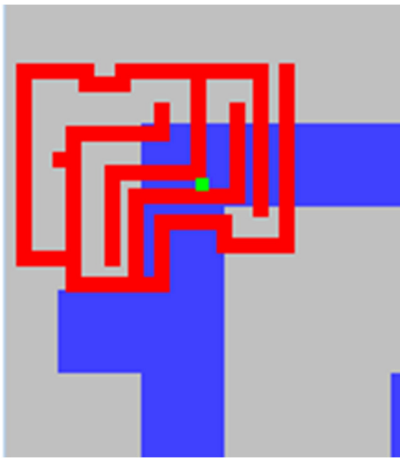
Alarm[0] Event

```
position += 1;
if position > string_length(text) { position = 1 };
alarm = 25;
```

Draw Event

```
last = min(position+width,string_length(text)+1);
first = max(0,position+width-last);
draw_text(x,y,string_copy(text,position,last-
position)+string_copy(text,0,first));
```

Map



```
Assumes you have a sprite, spr_map, 6x6 pixels in solid red. And a wall parent
object obj_wall_parent(no sprite required).

Draw Event (control object)
```

```
var d,a,xx,yy;
xx = Player_Obj.x;
yy = Player_Obj.y;
with(obj_wall_parent)
{
    d = point_distance(xx,yy,x,y);
   if(d <= 1000)
    {
        d = d/500*75;
        a = point_direction(xx,yy,x,y)
        draw_sprite(spr_map, 0, view_xview[0]+75 + lengthdir_x(d,a),
view_yview[0]+75 + lengthdir_y(d,a));
    }
}
draw_set_color(c_lime);draw_rectangle(view_xview[0]+75-2,
view_yview[0]+75-2,view_xview[0]+75+2,view_yview[0]+75+2,0);
```

Mist Effect

Creates a darkening red colour over the screen based on the players health.

Assumes you have a sprite, spr_blood, 20x20 pixels coloured red.

Example Usage

Create Event (control object)

```
health=100;
alpha=0;
```

Step Event

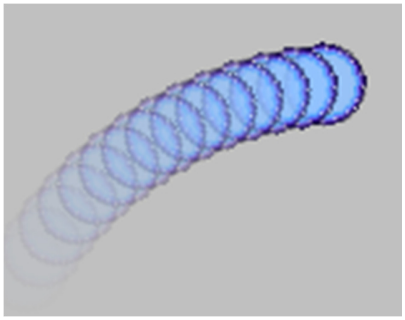
```
if keyboard_check(ord('A')){health-=1;}
if keyboard_check(ord('D')){health+=1;}
alpha=1-(0.5+(health/200));
```

Draw Event

```
draw_sprite_ext( spr_blood, 0, 1, 1, room_width/20, room_height/20, 0,
c_red, alpha);
draw_text(100,100,health);
```

Motion Trail

Create a motion trail effect



You'll need a sprite a transparent circle inside.

Create Event

```
t = 0;
tm = 16;
speed = 10 + random(5);
_direction = -5 + random(10);
```

Draw Event

```
t = min(t + 1, tm);
for (i = t; i > 1; i -= 1)
{
    tx[i] = tx[i-1];
    ty[i] = ty[i-1];
    ta[i] = ta[i-1];
    ti[i] = ti[i-1];
}
tx[1] = x;
ty[1] = y;
ta[1] = direction;
ti[i] = image_index;
direction += _direction;

for (i = 1; i <= t; i += 1)
draw_sprite_ext(sprite_index,ti[i],tx[i],ty[i],image_xscale,image_yscale,ta[
i],image_blend,sqr(1-i/t));
```

Mouse Over Button Effect

Changes sprite if mouse over or pressed on object.

This example uses a sprite with 3 different submiages,0=no mouse, 1=mouse over and 2=mouse pressed

```
Step Event
if(mouse_x>=x && mouse_x<x+sprite_width && mouse_y>=y &&
mouse_y<y+sprite_height)
{
    if(mouse_check_button(mb_left))
    {
        image_index = 2;
    }
    else
    {
        image_index = 1;
    }
}
```

Mouse Over Button Effect 2

Changes sprite if mouse over or pressed on object.

This example assumes sprite origin at top left.

Create Event

```
Mouse_over=false;
```

Step Event

```
if(mouse_x>=x && mouse_x<x+sprite_width && mouse_y>=y &&
mouse_y<y+sprite_height)
{
   mouse_over=true
}
else
{
     mouse_over=false
}
```

Draw Event

```
draw_self();
if mouse_over draw_text(100,100,"Mouse is Over");
```

Mouse Pointer Point Direction

Draws a mouse cursor pointing in direction of movement.

This Example assumes sprite pointing right with origin as center.

Step Event

```
move_towards_point(mouse_x,mouse_y,point_distance (x,y,mouse_x,mouse_y)/6);
```

Draw Event

```
window_set_cursor(cr_none);//hides default

draw_sprite_ext(sprite_index,0,x,y,1,1,point_direction
(xprevious,yprevious,x,y),c_white,1);

//Note: use window_set_cursor(cr_default); to allow drawing default cursor
again.
```

Moving Background

Set a background for your room and make it move with the following code.

Example Usage

Step Event

```
 if keyboard_check(vk_left)
{
    background_hspeed = -2;
}
if keyboard_check(vk_right)
{
    background_hspeed = 2;
}
```

Moving Cloud Effect

Creates a cool effect of moving clouds.

This example uses two objects, the cloud object and a spawning control object.

Object obj_cloud (with cloud sprite assigned).

Create Event

```
motion_set(90,0.5+random(2));
```

Step Event

```
if y<-100 instance_destroy();
```

Object obj_spawn_clouds

Create Event

```
spawn_speed=10;//smaller number means quicker spawning

alarm[0]=spawn_speed;
```

Alarm Event

```
instance_create(random(room_width),room_height+100, obj_cloud);
alarm[0]=spawn_speed;
```

Moving Platform 2

Another way of creating a moving platform.

This example uses two objects, obj_block and obj_platform. With obj_block assign a 32x32 pixel sprite and uncheck the 'visible' button in the object properties. For this example obj_platform is 96x32 pixels.

Platform Object obj_platform

```
Create Event
```

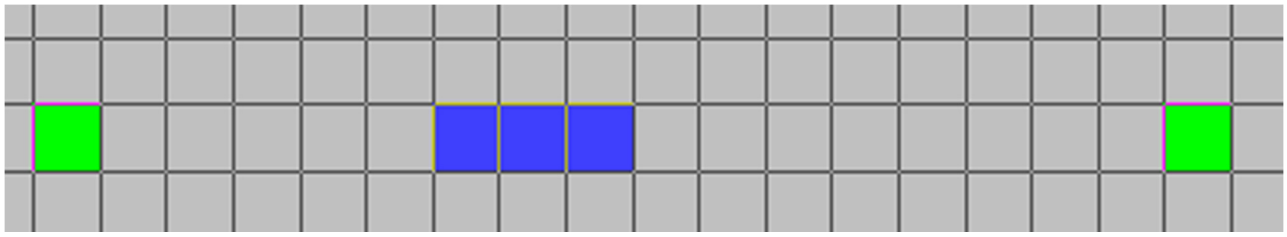
```
motion_set(0,3);
```

```
Collision with obj_block
```

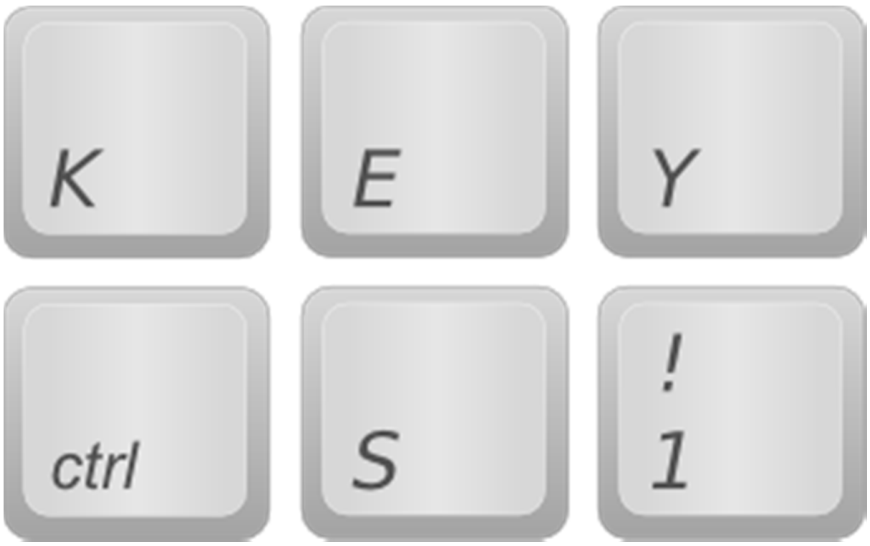
```
hspeed *= -1;
```

Just place obj_platform between 2 obj_blocks, and the platform will happily bounce back and forth.

Example:



Multi Key Press



This little script detects two key presses and returns true or false

Script: both_pressed

```
if keyboard_check(ord(argument0)) && keyboard_check(ord(argument1))
{
    return(true);
}
else
{
    return(false);
}
```

Example Usage (in Step Event):

```
if both_pressed('Z','X')
{
     //Do Something
}
else
{
     //Do Something Else
}
```

This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Move and Rotate

Moves in direction pointing.

Assumes you have a right-facing sprite assigned to the object, with origin as center.

Step Event

```
image_angle=direction;
if speed>10
{
speed=10;
}

if speed<-8
{
speed=-8;
}
if keyboard_check(ord('A'))
   {
   direction+=6;
   }
if keyboard_check(ord('D'))
    {
   direction-=6;
    }
if keyboard_check(ord('W'))
    {
speed+=2;
    }
if keyboard_check(ord('S'))
    {
    speed-=2;
    }
if speed>0 speed-=0.1;
if speed<0 speed+=0.1;
```

Moving Platform

A simple moving platform.

You'll need to assign a sprite to your object.

Create Event

```
distance=200;
moving="right";
start_x=x;
```

Step Event

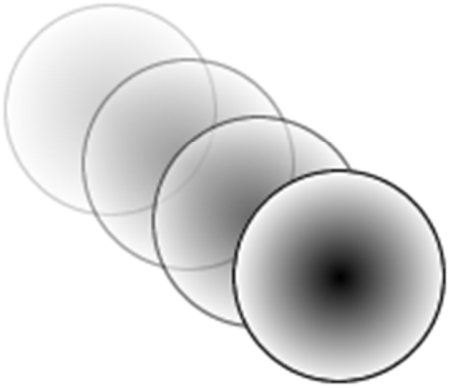
```
if moving="right" x+=4;
if moving="left" x-=4;
if x=start_x+distance moving="left";
if x=start_x moving="right";
```

Can also be used as up/down movement, just change x to y, and right and left to down and up.

If you have multiple platforms that move different distances, just set the above as a parent object and in the create event of the child objects set distance as required.

Multiple Collision

Makes sure multiple items are destroyed upon collision.



```
Collision Event (Put in a ball's collision event with obj_target_parent).
```

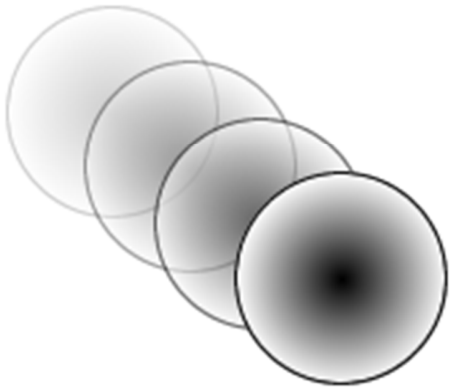
```
with(obj_target_parent){
 if(instance_place(x,y,other.id){
   instance_destroy();
 }
```

You'll to create an object obj_target_parent , no sprite or code is needed.

Just set obj_target_parent as parent of your objects that can be destroyed. For example if a ball hits more than one object on a collision, all objects it touches at that time with parent set will be destroyed.

Multiple Press (Devices)

Some mobile devices allow detection of you touching five different places on screen simultaneously. This example allows you to visualize these places.



Assumes you have a sprite assigned for object obj_finger_mark.

```
 Step Event in Control Object obj_control
```

```
for (i = 0; i < 5; i += 1; )
{
  if device_mouse_check_button(i, mb_lef;
  {
      instance_create(device_mouse_raw_x(i), device_mouse_raw_y(i),
      obj_finger_mark);
  }
}
```

Just set obj_target_parent as parent of your objects, so for example if a ball hits more than one object on a collision, all will be destroyed.

Nested if-else

How to know and use correctly a nested if-else.

Normaly when we code a action with several cases of if to check, the usual options are using:

```
if(condition1){action1;}
else{
  if(condition2){action2;}
  else{
    if(condition3){action3;}
    else {action4;}
  }
}
Or using the switch statement.
```

The second option only work with conditions that are base in == (like x==2 , action=='stand') and can make comparation with one single variable. And the first one can be a chore to work.

So we can use a structure that looks a better case of switch and more compact and easy to implement and edit.Let's look in the following example.

```
if(x<2){y=0;}
else if(x>6){y=1;}
else if(x==3 or x==4){y=2;}
else {y=3;}
  }
}
```

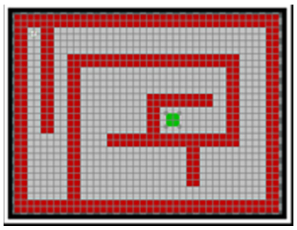
Using the way is easier to work with and it is more flexible to use condition.Plus, if one of the condition is fulfilled, the next conditions are skipped.

Path Finding

Path finding is when you you calculate a path between two point in a room or map.

Usually you'll want to find a path that avoids things like walls or trees.

This example shows how to do simple path finding by avoiding walls.



Above: Example Room For Testing

This example assumes you have three objects with sprites assigned:

obj_wall 32x32 pixels

obj_finish 32x32 pixels

obj_player with sprite named spr_player – 16x16 face with 16x16 border ie:

Object obj_player

Create Event

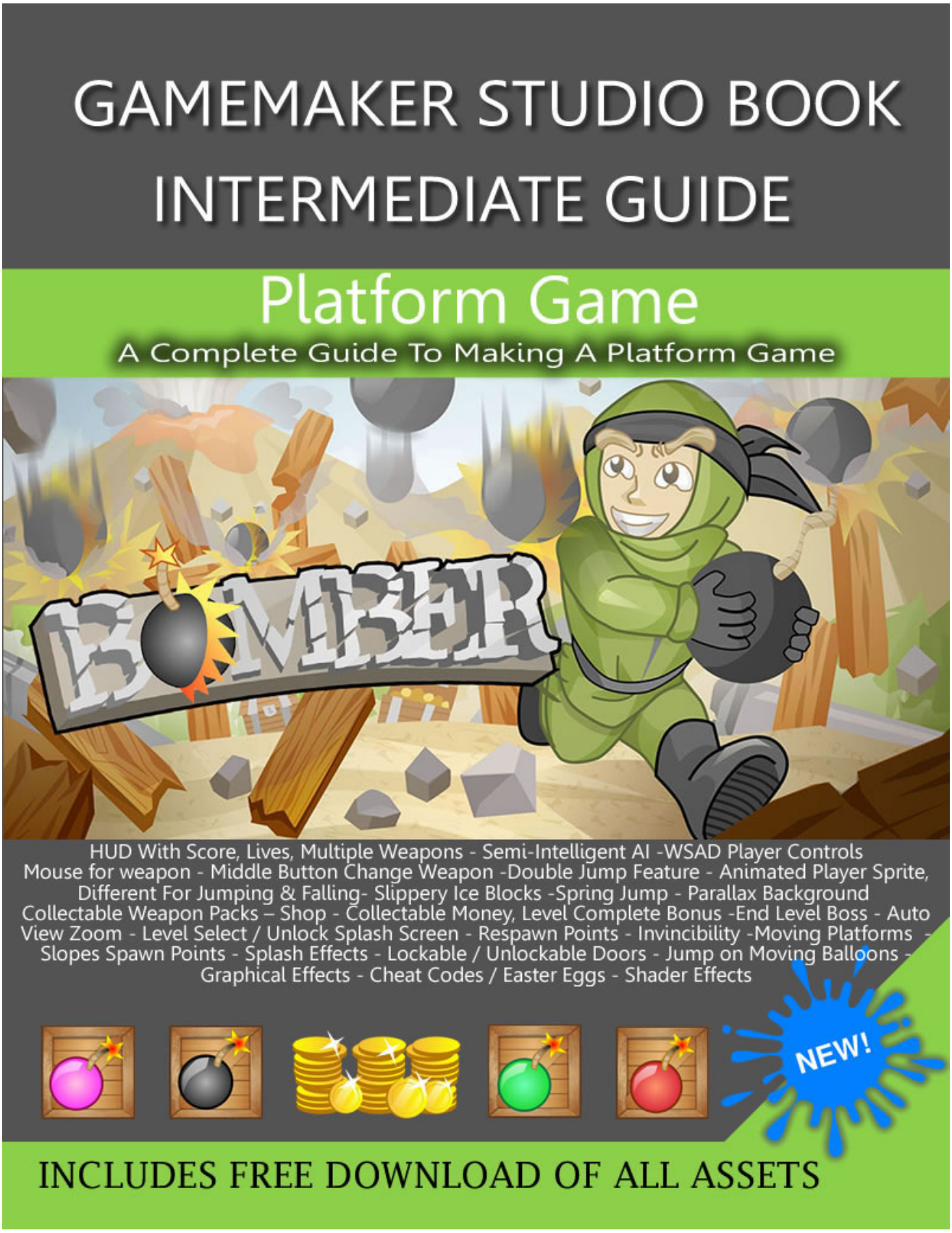
```
dir=direction;
```

Begin Step Event

```
 dir=direction
grid=mp_grid_create(0,0,room_width/32,room_height/32,32,32);
path=path_add();
mp_grid_add_instances(grid,obj_wall,1)
mp_grid_path(grid,path,x,y,obj_finish.x,obj_finish.y,1);
path_start(path,2,'',1);
```

Draw Event

```
 draw_sprite_ext(spr_player,0,x,y,1,1,dir,c_white,1);
```



This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Platform & Change Direction

A quick way to make an object change direction.

Code

```
hspeed*=-1;
```

Can also be used with vspeed. Just put it in the event you want, like mouse button or collision event.

This can also be used to make a moving platform change direction. Just put an invisible object to the left and right where you want the range of the platform to be.

Platform Jumping

This code allows a player to jump once or twice.

```
Create Event
gravity_direction=270;
gravity=0.5;
friction=0.2;
jump=1;
max_jumps=2;// change this to 1 for single jump
Step Event
if keyboard_check_pressed(vk_space)and max_jumps >0
{
  vspeed -= 10
  max_jumps -= 1
}
Collison with obj_floor
max_jumps=2;
if vspeed > 0 && !place_free(x,y+vspeed)
{
     move_contact(270);
     vspeed = 0;
}
```

Player Lock

In some cases, you want that the user don't have control over player, when he is hurt, in a cutscene , a pause screen or in other situations. To have a better control of when you can move the player or not. We are going using a flag variable control_lock that has 2 values: False if you the user has full control over the player, or true if the player's control is locked. Consider an example that the obj_player, hit by a moving ball obj_ball, it loses control for 60 steps. Below is an example:

```
Create Event with obj_ball
```

```
control_lock=false;
```

```
Collision Event with obj_ball
```

```
If(control_lock==false){
  control_lock=true;
  alarm[0]=60;
}
```

```
Step Event
```

```
If(control_lock==true)exit;
if(keyboard_check(vk_right))x+=4;
if(keyboard_check(vk_left))x-=4;
```

```
Alarm[0] event
```

```
control_lock=false;
```

So in the Step event, if control lock is true, the exit statement is executed, skipping all the code after this.

Point To Nearest

Point to nearest object.

This example assumes have a sprite pointing right and a target object obj_target.

Create Event

```
turn_speed = 0;
sight_dist = 200; // Length of the field of view
sight_angle = 15; // Width of the field of view
```

Step Event

```
  var target, diff;

target = instance_nearest(x, y, obj_target);
if (instance_exists(target)) {
    // Make the turret slowly turn to the nearest enemy
    turn_speed += sign(angle_diff(point_direction(x, y, target.x, target.y),
image_angle)) * 0.2;
    if (abs(turn_speed) > 3) turn_speed = sign(turn_speed) * 3;
}
else {
    // Make it stop when there is no enemy
    turn_speed -= sign(turn_speed) * 0.2;
}
image_angle += turn_speed;
```

Point Direction Movement

This little nugget of a code will make a sprite look in the direction it's moving.

Create Event

```
mirror=false;
```

Step Event

```
if keyboard_check(vk_left)
{
    mirror=true;
    //Plus any movement code
}


if keyboard_check(vk_right)
{
    mirror=false;
    //Plus any movement code
}
```

Draw Event

```
if mirror=true
{
image_xscale=-1;
}
else
{
image_xscale=1;
}
Draw_self();//as required
```

Power Bar

Upon key press power bar goes up and down then gives value to global.powerreturn. Sets global.powerreturn as -1 when not in use (in your code check that not -1 and after using in your code set back to -1 when done using).

Create Event

```
mult = 1;

powerbar = 0;

global.powerreturn=-1;
```

Step Event

```
if keyboard_check(ord('Z'))
{
     powerbar += 5*mult;
     if powerbar >= 100 || powerbar <= 0
     {
          powerbar = clamp(powerbar,0,100);
          mult *= -1;
     }
}
```

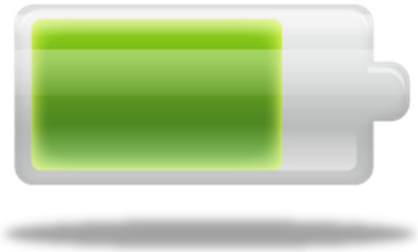
```
if keyboard_check_released(ord('Z'))
{
     global.powerreturn=powerbar;
     show_message(global.powerreturn);
}
```

Draw Event

```
draw_text(6,120,powerbar);
draw_set_color(c_red);
draw_rectangle(0,100-powerbar,25,100,0)
draw_set_color(c_black);
draw_rectangle(0,0,25,100,1);
```

Power Up

Creates a temporary power up.



Create Event

```
power_up=false;
```

Alarm Event

```
power_up=false;
```

Step Event

```
if keyboard_check(ord('P')) && power_up=false
{
    power_up=true;
    alarm[0]=4*room_speed;
}
// sets power_up when active
```

Draw Event

```
if power_up=true
{
    draw_text(50,50,"POWER UP");
}
else
{
    draw_text(50,50,"NO POWER UP");
}
```

Pushing Block

A simple method for pushing a block by a player.

This example assumes you have a player sprite and block sprite.

```
Player Step Event (provided for example, replace with your own movement code as
required
```

```
if keyboard_check(vk_left)x-=5;
if keyboard_check(vk_right)x+=5;
```

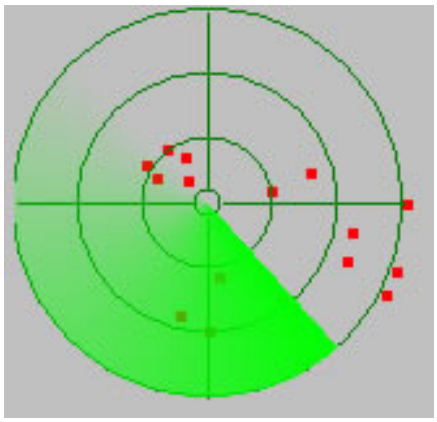
```
Block Step Event
```

```
if place_meeting(x+5,y,obj_player)if place_free(x-5,y){if
keyboard_check(vk_left){x-=5}}
if place_meeting (x-5,y,obj_player) if place_free(x+5,y){if keyboard_check
(vk_right) {x+=5}}
```

Can also be used on y axis, just copy and paste and change the xs to ys.

Radar

This cool code draws a radar and position of enemies in relation to the player.



This assumes you have:

Object Player_Obj with sprite assigned

Object Enemy_Parent with no sprite

Object Obj_Enemy with sprite and parent set as Enemy_Parent

Sprite RadarBlip (2x2 pixels in red)

Code for object Player_Obj

```
Step Event
```

```
//replace with your movement code as needed
x+=4*(keyboard_check(vk_right)-keyboard_check(vk_left));
y+=4*(keyboard_check(vk_down)-keyboard_check(vk_up));
```

Radar Object Radar_Obj

Create Event

```
//Initialize Variables

indexCount = 0;             //iterates the index for the enemyIndex
radarX = 0;                 //the corrosponding X coord of the playerShip
radarY = 0;                 //the corrosponding Y coord of the playerShip
maxEnemies = 0;
radar_sweep_angle=2;
```

Draw Event

```
//stop here if no player
if (!instance_exists(Player_Obj))
{
    exit;
}
//local vars for easy access in the with() and to prevent creating variables
in the instances
var d,a,xx,yy;
xx = Player_Obj.x;
yy = Player_Obj.y;

with(Enemy_Parent)
{
    //how far
    d = point_distance(xx,yy,x,y);
    //in range
    if( d < 3000 && d > 1500)  // This will set the blips to the outside
edge, creating a lingering effect
    {
        //convert radar range to radar display radius
        d = 75;
        //angle to target
        a = point_direction(xx,yy,x,y)
        //draw relative to center of radar using simplified lengthdir
function
        draw_sprite(RadarBlip, 0, view_xview[0]+75 + lengthdir_x(d,a),
view_yview[0]+75 + lengthdir_y(d,a));
    }
    else if(d <= 1500)  // This is the standard distance conversion on the
radaar screen.
    {
```

```
        d = d/1500*75;
        a = point_direction(xx,yy,x,y)
        draw_sprite(RadarBlip, 0, view_xview[0]+75 + lengthdir_x(d,a),
view_yview[0]+75 + lengthdir_y(d,a));
    }
}
```

Random Numbers and Algorithms

The random algorithms are normaly used to add diversity and extra difficult for the game to avoid it turn linear and predictable. The most simple case is the function choose function, that pick randomly one of his arguments. If, for example, you has a boss with 3 attack pattern set by the variable attack, a way to use is put in the boss's event (that can be in Alarm Event, for example)

```
Alarm[0] Event
```

```
attack=choose(1,1,1,2,2,3);
```

With repeat numbers, we raise the probability of this specific attack. In this case, attack pattern 1 is more common and attack pattern 3 is more rare. Note that this function accepts upto 16 options. In this case 1 has 3/6 (50%) chance, 2 a 2/6 (33%) chance and 3 a 1/6 (17%) chance.

We can use function like random(x) (that returns randomly a real number (with decimals) between 0 and x, not including x), random_range(x1,x2) (that returns randomly a real number between x1 and x2, not including x2), and theirs respectives versions that return only integer numbers(whole numbers) (irandom(x) and irandom_range(x1,x2)).
For example let's place an object in a random placee in the room:

```
Create Event
```

```
x=floor(random(room_width));
y=floor(random(room_height));
```

This will make the object start at a random x / y position. floor rounds the number down to a whole number. You could also use ceiling which would round up or round to round to nearest whole number.

For the purposes of making games easier to create and test GameMaker will always generate the same random numbers each time a game is played. For example if in your game you rolled a dice 5 times and got 5 3 1 6 3 2, the next time you played you'd also get 5 3 1 6 3 2. Depending on your game you may want different numbers each time. To do this use:

```
randomize();
```

There may times, for example you have randomly generated level, that you want to be the same each time, you can use:

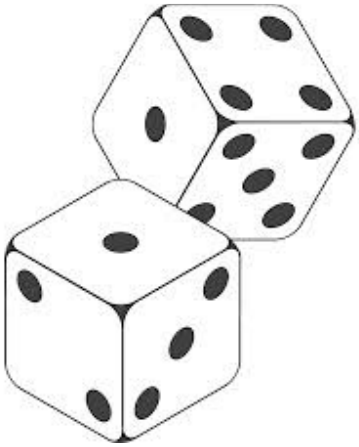
```
random_set_seed(number);
```

You could set a new seed for each level, for example: random_set_seed(level), where level is the level number as an integer.

This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Random Word From A Text File

This example shows how to select a random word from a text file.



Script

```
var file;
if(file_exists(working_directory + "dictionary.txt")){
    //open the dictionary file
    file = file_text_open_read(working_directory + "dictionary.txt");
    if(file == -1){
        //if loading the file failed return -1
        return -1; //will end the script
    }
    var wordList, wordNumber = 0;
    //make a list containing all words of the dictionary
    while(!file_text_eof(file)){
        wordList[wordNumber] = file_text_read_string(file);
        file_text_readln(file);
        wordNumber++;
```

```
    }
    file_text_close(file);
    return wordList[irandom(wordNumber-1)]; //return a random word
}
```

To get a word use:

```
word = newWord();
```

This uses an included file, dictionary.txt , an example of the contents is below:

CHEESE
BACON
MELON
POTATO
MUSHROOM

Random Word From Array

Chooses a random word from an array.

```
Script random_word
```

```
var array_word;
array_word[0]="CHEESE";
array_word[1]="BACON";
array_word[2]="BREAD";
array_word[3]="MUSTARD";
array_word[4]="CAKE";
array_word[5]="CHOCOLATE";
array_word[6]="PIE";
array_word[7]="ORANAGE";
array_word[8]="BANANA";
array_word[9]="COFFEE";
size=array_length_1d(array_word)-1;
return array_word[irandom(size)];Change as required
Example Usage:
Word=random_word();
```

Real Time Clock Example

Create Event

```
minute=0;

second=0;

hour=0;
```

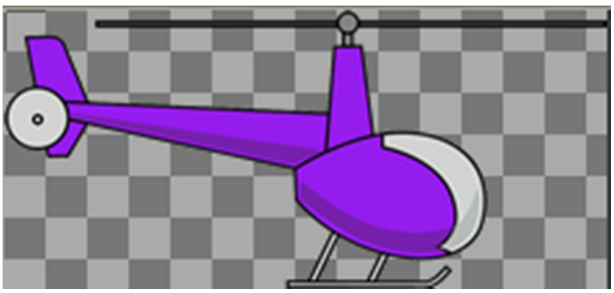
Step Event

```
minute=current_minute;

hour=current_hour;

second=current_second;
```

Draw Event

```
draw_text(x,y,string(hour)+string(":")+string(minute)+string(":")+string(second));
```

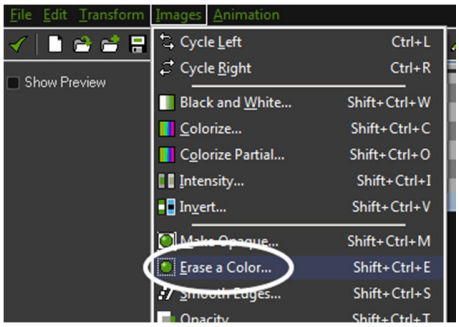
Remove Sprite Background



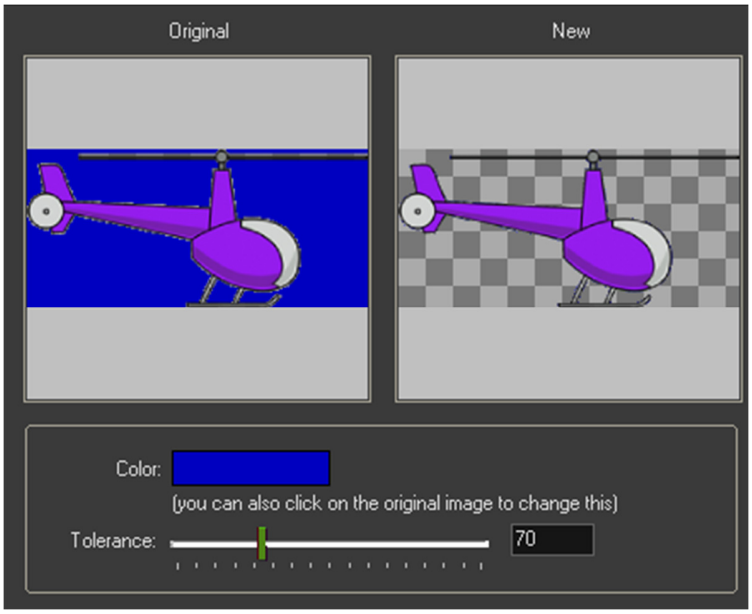
Sprite with background removed

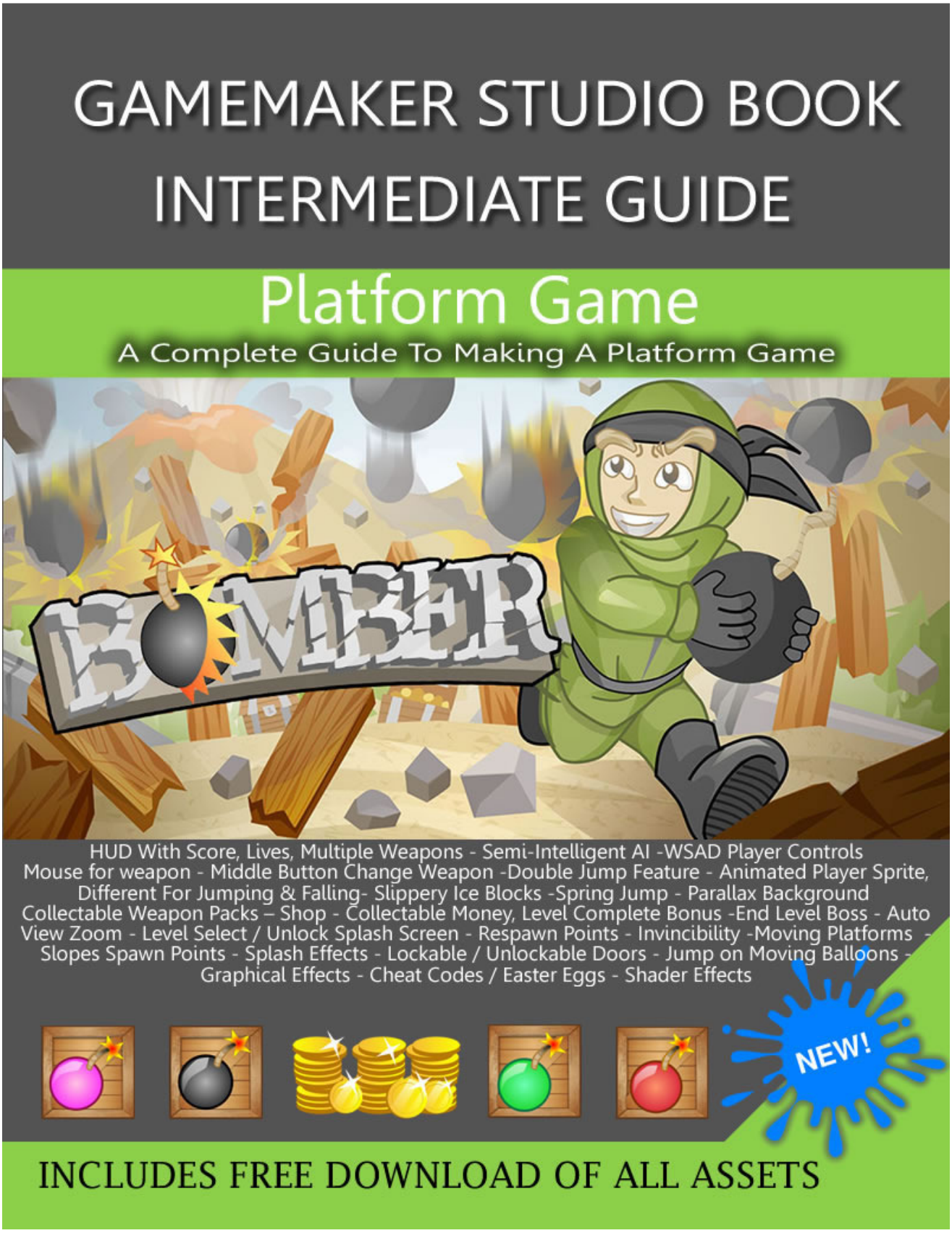
This is a quick and easy way to remove a background.

Open the sprite editor and select Erase a Color



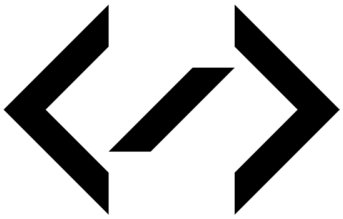
Then erase the background colour





This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Repeat



That is the basic structure of “repeat”:

```
repeat(n)
{
  code 1;
}
```

In this way, the code is repeated n times. With this form, it can group the same code that would be typed again in n-1 times. It can be used easily to created n instances randomly placed, using in arrays or data structures.

First example:

```
repeat(8)
  with(instance_create(x,y,obj_snow_splatter))
{
    speed=1;
    direction=irandom_range(1,360);
  }
```

With this one, is created 8 particles that moves in randomly directions with speed 1.

Second example:

```
Var n;
n=choose(1,2,3);
repeat(n){
  x+=10;
}
With his one the code x+=10; is repeated in n time, that is set randomly between 1
and 3.
```

Room Transition

An easy to way to fade in and out of rooms.

For this code you'll need a sprite 20x20 pixels coloured black, spr_fade.

Create Event

```
image_alpha=0;
```

Step Event

```
image_alpha+=0.01;

if image_alpha>1
{
   // do something (go to a room for example)
   instance_destroy();
}
```

Draw Event

```
draw_sprite_ext( spr_fade, 0, 1, 1, room_width/20, room_height/20, 0, 0,
image_alpha );
```

This should also work if you're using views, but may need some adjustment.

To fade in change start the alpha as 1 and reduce in the step event, and change the conditional to check for <0.

Rotating Background

A demonstration of a rotating background.

You'll need a background assigned to the room to see this effect. You'll also need to views.

```
Create Event
```

```
angle=0;
```

```
Step Event
```

```
angle+=1;
if angle>359 angle=0;
```

```
Draw Event
```

```
view_angle[0]=angle;
```

Score Example

This examples assumes you have an object obj_coin with sprite assigned.

Object obj_player

Create Event

```
score=0;
```

Step Event

```
x+=4*(keyboard_check(vk_right)-keyboard_check(vk_left));
y+=4*(keyboard_check(vk_down)-keyboard_check(vk_up));
```

Collision with object obj_coin

```
with (other) instance_destroy();
score+=1;
```

Draw Event

```
draw_text(50,50,"Score:="+string(score));
draw_self();
```

Score With Leading Zeros

Draw Event

```
str = string(score);
```

```
draw_text(x, y, string_repeat("0", 6-string_length(str))+str);
```

Scrolling Credits

Allows you to swap object positions thorugh clicking

Create Event

```
text[0]="This Text Floats Up";
text[1]="And Fade Outs at Top Of Screen";
text[2]="Great For Providing Inoformation";
text[3]="Or Display Game Credits";
text[4]="When It's Done";
text[5]="Program It To Do Something";
total_lines=array_length_1d(text);
i=room_height;
p=0;
h=1;
title=room_first;
```

Draw Event

```
draw_set_color(c_black);
draw_set_halign(fa_center);
draw_set_valign(fa_middle);
draw_set_font(font0);///set your font here
h=i/100;
draw_set_alpha(h);
draw_text_ext(room_width/2,i,text[p],20,room_width-20);
if keyboard_check(vk_enter) i-=5; else i-=2
if i<0
{
   i=(room_height+(string_height(text[p])*2)); p+=1
}
draw_set_alpha(1);
if p>total_lines then
{//do something, go to room/restart for example}
```

Shooting Alarm

This code sets an alarm to limit how often a player can shoot.

Create Event

```
can_shoot=true;
```

Alarm[0] Event

```
can_shoot=true;
```

Step Event

```
if can_shoot=true
{ //do something}
else
{//do something else}
```

Global Left Button

```
if can_shoot=true

{

alarm[0]=room_speed*3;

can_shoot=false;

}
```

Draw Event

```
//Provided for testing purposes/example

if can_shoot=true

{ draw_text(50,50,"Can Shoot"); }
else
{draw_text(50,50,"Can't Shoot");}
```

Sliding Bar

Allows user to change/set values using a sliding bar with button.

This example assumes you have two sprites, spr_sliderTrack solid black 4*4 pixels, and spr_sliderMark solid red 4*8 pixels.

Object obj_slider

Create Event

```
minValue = self.x - (sprite_width / 2);
maxValue = self.x + (sprite_width / 2);
curValue = 0.5;
length = maxValue - minValue;
isSliding = false;
```

Step Event

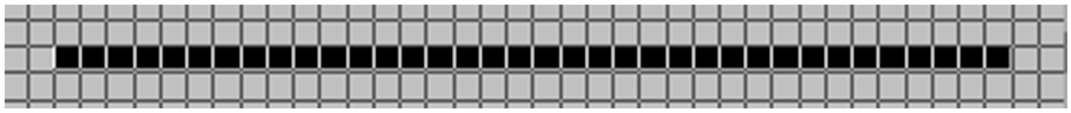
```
 if(mouse_check_button_pressed(mb_left) && position_meeting(mouse_x,
mouse_y, self))
{
    isSliding = true;
}
if(mouse_check_button_released(mb_left))
{
    isSliding = false;
}

if(isSliding) {
    curValue = (mouse_x - minValue) / length;
    if(mouse_x < minValue) {
        curValue = 0;
    }
    if(mouse_x > maxValue)
{
        curValue = 1;
```

```
    }
}

      switch(function)
      {
           case "mainVolume": sound_global_volume(curValue); break;
      }
```

Draw Event

```
draw_self();
draw_sprite(spr_sliderMark, 0, (length * curValue) + minValue, y);
draw_text(x, y+10, string(curValue));
```

When you place object obj_slider in the room click on a corner and stretch out as needed, for example:



Slowly Move

Two methods to make an object slowly move to a position

Method 1

Step Event

```
movement_speed=25;//Higher Number Makes Slower Speed
target_x=mouse_x;//or other target position
target_y=mouse_y;//or other target position
x +=( target_x-x)/ movement_speed; //target position-current position
y +=( target_y-y)/ movement_speed; //target position-current position
```

Method 2

Script slowly_move

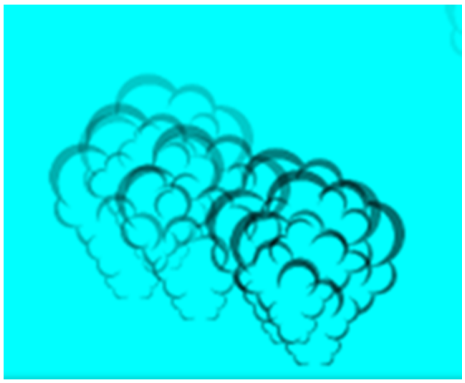
```
/*argument0=x pos, argument1=y pos,argument2=targetx
argument3=targety, argument4=distance, argument5=speed*/

if point_distance(argument0, argument1, argument2, argument3) > argument4
//if the mouse is outside a certain reach/5 of the object
{
    move_towards_point(argument2, argument3, argument5); //move towards
mouse_x, mouse_y in the speed of 5
}
    else speed = 0;
```

Example, in a step event:

```
target_x=mouse_x;
target_y=mouse_y;
slowly_move(x,y,target_x,target_y,0,5);
```

Smoke Effect

A simple code to create a falling and fading effect



Showing Smoke Effect

Create Event

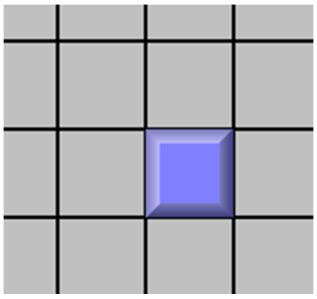
```
motion_set(90,2);
```

Step Event

```
image_alpha-=0.02;
if image_alpha<0 {instance_destroy()}
```

Assign the sprite you want to use for this effect, making sure the background is transparent.You can change the speed and direction as required, for example change the 90 to 270 and the effect will go down the screen and fade, for example for a blood splatter effect.

Snap To Grid



Create Event

```
grid_size=50;//required

grid_x=round(room_width div grid_size)+1;//used in example to draw grid

grid_y=round(room_height div grid_size)+1;//used in example to draw grid
```

Step Event

```
x=grid_size*(round(mouse_x div grid_size));
y=grid_size*(round(mouse_y div grid_size));
```

Draw Event

```
draw_self();//as required

///draw a grid example
 var i,j;
  draw_set_color(c_black);
 for (i = 0; i < grid_x; i += 1)
{
    {
        for (j = 0; j < grid_y; j += 1)
        {

draw_rectangle(i*grid_size,j*grid_size,i*grid_size+49,j*grid_size+49,1);
        }
    }
}
```

There is also an built function

```
move_snap(grid_size,grid_size);
```

Though you'll probably find the script better as it detects which grid to position, not an approximation.

Snow and Rain Effect

Creates a beautiful snow / rain effect.

Just create an object and put this in the step or draw event.

Code

```
effect_create_above(ef_snow,irandom(room_width), -100, random(4)+0.1,
c_white);

Just change ef_snow to ef_rain for a rain effect.
```

Sniper Effect

Blacks out the screen except small 'gun site'.

You'll need a sprite a transparent circle inside.

Draw Event

```
 draw_set_color(c_black);
draw_sprite(sprite_index,image_index,mouse_x,mouse_y);
draw_rectangle(0,0,mouse_x-sprite_xoffset,room_height,false);
draw_rectangle(mouse_x-
sprite_xoffset+sprite_width,0,room_width,room_height,false);
draw_rectangle(0,0,room_width,mouse_y-sprite_yoffset,false);
draw_rectangle(0,mouse_y-
sprite_yoffset+sprite_height,room_width,room_height,false);
```

This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Soft Ball Physics

Creates a squashy bouncy ball effect:

This example assumes you have object obj_solid_parent with a sprite assigned.

You'll also need a obj_ball with a ball sprite (128x128 pixels) with transparent background.

Object Ball

Draw Event

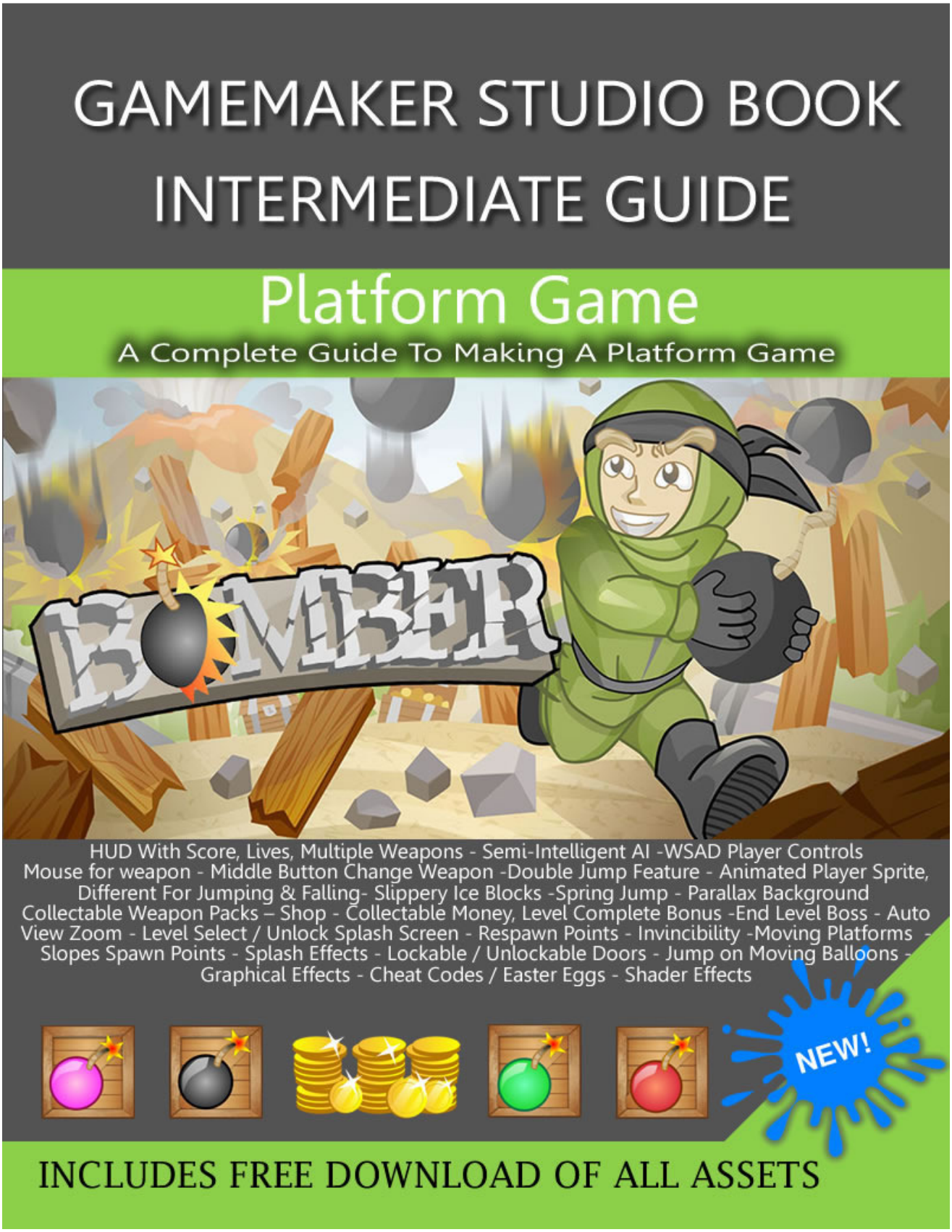
```
texture=sprite_get_texture(spr_ball,0);

radius=48;

quality=50;

bounce=1;

image_xscale=radius/80;

image_yscale=image_xscale;

gravity_direction=270;

gravity=.5;

scr_blob(texture,radius,bounce,quality,obj_solid_parent);
```

Script scr_blob

```
//scr_blob(texture,radius,bounce,quality,solid_parent)

//////////////////////////////////////////////
//////////////////////////////////////////////
//////////////////////////////////////////////
//Initialize:
blob_t=argument0                //Texture
blob_r=argument1                //Radius
    blob_b=100*argument2/argument3  //Bouncyness
    blob_q=argument3                //Quality
    blob_s=argument4                //Solid parent (what objects to colide
with)
    blob_a=0                        //Angle
    blob_rs=0                       //Rotational speed
    blob_rn=(1+20/blob_q)           //Round "corners (when colliding with
things) Don't change
    blob_db=true                    //Whether or not to draw twice as many
vertexes are there are "arms"
    for (i=0;i<=blob_q;i+=1){
        blob_l[i]=blob_r
        blob_d[i]=2*pi*i/blob_q+pi/2
        blob_dc[i]=cos(blob_d[i])
        blob_ds[i]=sin(blob_d[i])}
    u_start=blob_r/10
    u_s_add=blob_r/6
    u_to=blob_r+u_s_add
    blob_sl=floor(40/blob_q)
blob_mr=blob_r
//////////////////////////////////////////////
//////////////////////////////////////////////
//Anti-stuck
if collision_line(x,y,xprevious,yprevious,blob_s,true,true){
    if
!collision_line(xprevious,yprevious,xprevious,y,blob_s,true,true){x=xpreviou
s hspeed=0}
    else {
        if
!collision_line(xprevious,yprevious,x,yprevious,blob_s,true,true){y=ypreviou
s vspeed=0}
        else{
            speed=0
            x=xprevious
            y=yprevious}}
            if collision_circle(x,y,1,blob_s,true,true){
                for (s=4;s<blob_r;s+=4){
```

```
                        for (d=0;d<360;d+=20){
                            xx=x+lengthdir_x(s,d);
                            yy=y+lengthdir_y(s,d);
                            if !collision_circle(xx,yy,1,blob_s,true,true){
                                x=xx;
                                xprevious=xx;
                                y=yy;
                                yprevious=yy;
                                d=360;
                                s=blob_r;
                                break;}}}}}
////////////////////////////////////////////
////////////////////////////////////////////
//Physics
var Fx,Fy,F,c,s,u;
Fx=0
Fy=0
if collision_circle(x,y,blob_r,blob_s,true,true){blob_rs*=0.8}
for (i=0;i<blob_q;i+=1){
    c=blob_dc[i]
    s=blob_ds[i]
    u=blob_r
    blob_l[i]=u
    if collision_line(x,y,x+u*c,y-u*s,blob_s,1,1)>0{
        u_add=u_s_add
        test=true
        for (u=u_start;u<u_to;u+=u_add){
            if collision_line(x,y,x+u*c,y-u*s,blob_s,1,1){
                global.bug=collision_line(x,y,x+u*c,y-u*s,blob_s,1,1)
                if test=true{
                    u-=u_add
                    u_add=1
                    test=false}
                else{break;}}}
        u=min(u,blob_r)
        uu=u/blob_r
        blob_rs+=2*sin(degtorad(direction)-blob_d[i])*speed/((10+blob_q)*uu)
//Rotate
        blob_mr+=(blob_r-u)
        F=blob_b*(1-uu)
        Fx-=F*c
        Fy-=F*s
        blob_l[i]=u}}
for (i=0;i<blob_q;i+=1){
    blob_l[i]=min((blob_r*blob_sl+blob_l[i]+blob_l[(i+1) mod
blob_q]+blob_l[(i-1+blob_q) mod blob_q])/(3+blob_sl),blob_l[i])}
hspeed+=Fx
```

```
vspeed-=Fy
blob_a+=blob_rs
blob_l[blob_q]=blob_l[0]
//////////////////////////////////////////
//////////////////////////////////////////
//Drawing:
draw_set_color(c_white)
draw_primitive_begin_texture(pr_trianglefan,blob_t)
draw_vertex_texture(x,y,0.5,0.5)
for (i=0;i<blob_q;i+=1){
    d=blob_d[i]+degtorad(blob_a)
    draw_vertex_texture(x+blob_l[i]*blob_dc[i],y-
blob_l[i]*blob_ds[i],0.5*(1+cos(d)),0.5*(1-sin(d)))
    if blob_db=true{
        ii=i+1
        d1=(blob_d[i]+blob_d[ii])/2
        d=d1+degtorad(blob_a)
        l=(blob_l[i]+blob_l[ii])/2
        draw_vertex_texture(x+l*cos(d1),y-l*sin(d1),0.5*(1+cos(d)),0.5*(1-
sin(d)))}}
d=degtorad(blob_a)+pi/2
draw_vertex_texture(x,y-blob_l[0],0.5*(1+cos(d)),0.5*(1-sin(d)))
draw_primitive_end()
```



This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com

Sound Effects

Basic explanation of playing effects.

Before beginning, check in Global Game Section -> General tab, if the option “Use New Audio Engine”. This a simple of how play the sound snd_explosion a object obj_ball1 collides with object obj_ball2, and is destroyed in the process. The code of obj_ball1 will be:

```
Collision Event with obj_ball2
```

```
audio_play_sound(snd_explosion,20,0);
instance_destroy();
```

From the function audio_play_sound, it has 3 arguments. The first one is the sound’s name, the second set the channel priority of the sound (for beginners, you just need set 20 or any other value), and the third if the sound will loop (1) or not (0).

In the case that you want to loop a sound in the create event of obj_ball1, but want stop it after 60 steps, Just need follow this.

```
Create Event
```

```
audio_play_sound(snd_explosion,20,1);

alarm[0]=60;
```

```
Alarm[0] event
```

```
audio_stop_sound(snd_explosion);
```

Sprite Face Direction

This little code changes the sprite direction to match left/right movement

Step Event

```
if keyboard_check(ord('A'))
{
hspeed = -5;
image_xscale=-1;
image_angle = 0;
}
if keyboard_check(ord('D'))
{
hspeed = 5;
image_xscale=1;
image_angle = 0;
}
if keyboard_check(ord('W'))
{
image_xscale=1;
vspeed = -5;
image_angle = 90;
}
if keyboard_check(ord('S'))
{
image_xscale=1;
vspeed = 5;
image_angle =270;
}
if keyboard_check(vk_nokey)
{
vspeed = 0;
hspeed = 0;
//add these to set back to starting direction
image_xscale=1;
image_angle =0;
}
```

Sprite Rotate – Example 1

Using contants can make your code easier to write and understand. This example shows how to use constants to reference sprite subimges. This example assumes you have a sprite and four sub-images, set in order as left, down, right and up.

```
Constants (set the following as constants)
```

```
down=1
right=2
up=3
left=0
```

```
Draw Event:
```

```
draw_sprite(sprite_example,up,x,y);//change direction as needed
```

Sprite Rotate – Example 2

Using contants can make your code easier to write and understand. This example shows how to use constants to reference sprite subimges. This example assumes you have one sprite currently pointing left.

```
Constants (set the following as constants)
```

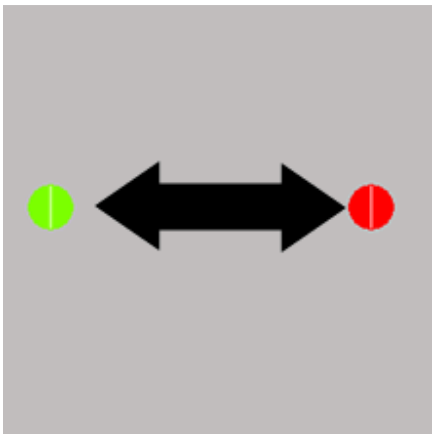
```
left=0;
down=90;
up=270;
right=180;
```

```
Draw Event:
```

```
draw_sprite_ext(sprite_example,1,x,y,1,1,left,1,1);//change direction as
needed
```

Swap Objects

Allows you to swap object positions thorugh clicking



Just create a parent object, obj_swap_parent to your objects, and use this code in object obj_swap_parent:

Create Event

```
image_index=0;
image_speed=0;
pressed=false;
selected=false;
```

Left Pressed Event

```
if(selected==false and pressed==false){
  pressed=true;
  selected=true;
  //check if had a previous object selected
  with(obj_par_ball){
    if(id!=other.id and selected==true){
    //swap the 2 objects
      var xx,yy;
      xx=other.x;
      yy=other.y;
      other.x=x;
      other.y=y;
      x=xx;
      y=yy;
```

```
      selected=false;
      other.selected=false;
    }
  }
}
```

Global Left Released

```
pressed=false;
```

Switch

Basic explanation of the structure “switch”.

That is the basic structure of “switch”:

```
switch(statement)
{
  case expression 1 : code 1; break;
  case expression 2 : code 2; break;
  case expression 3 : code 3; break;
  ...
  default : default code; break;
}
```

First, the statement is compared with the expression 1. If the result is ture, the code 1 is executed and , due the break statement, the loops ends. Otherwise, if the result is false, the stamen will be checked with the code 2 and so on. If all the cjecking turn false, the default code is executed. The default case is optional , so can be removed.

Note that the switch can be only work with comparison ==.

One simple example:

```
switch(x)
{
  case 0 :
  case 16 : y=0; break;
  case 32 : y=24; break;
  case 48 : y=48; break;
  default : y=64; break;
}
```

In this example , the y-position of an object depends of of the x-position, notice that the case x=0 and x=16 are grouped, that can be easily done if you remove the break statement in the first case.

Second example:

```
switch(vk_key){

  case vk_left : x-=2;break;
  case vk_right: x+=2; break;
  case vk_up   : y-=2; break;
  case vk_down : y+=2; break;

}

It is a simple 4-way movement.
```

Text Above Sprite

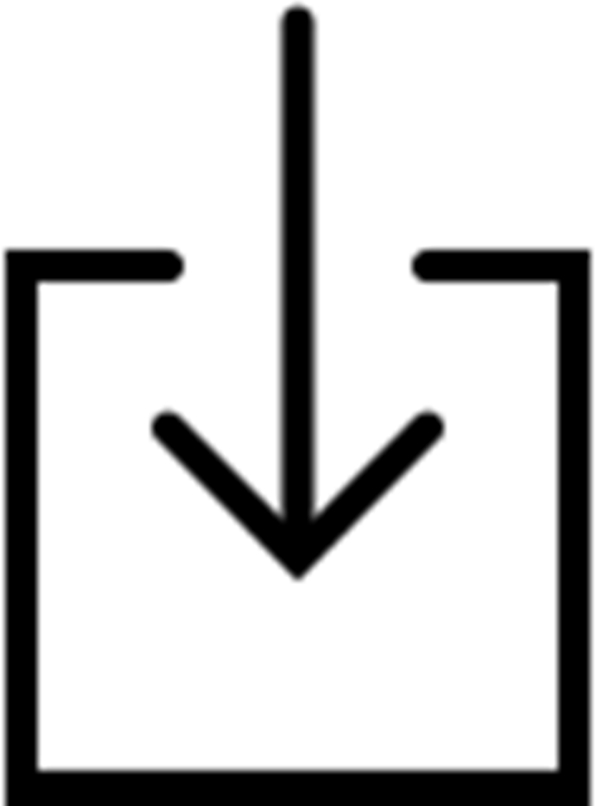
Example of text above sprite

This example assumes sprite origin as center.

Draw Event

```
font_height=16;//Put on create event if you have one
draw_self();//as required
draw_set_halign(fa_center);
draw_text(x,y-(sprite_height/2)-font_height,"Ambulance");
```

Teleport System



This examples assumes you have a player object and two teleport objects with sprites assigned, obj_tel_from and obj_tel_to

Place this in the player object's collision event with obj_tel_from

Code for Collision Event

```
x=obj_tel_to.x;
y=obj_tel_to.y;
```

Text With a Border / Shadow

This example shows how to select a random word from a text file.

Example of text with border

Script

```
var xx,yy, text;
text=argument0;  //text to draw
xx=argument1;//x position
yy=argument2;   //y position
draw_set_halign(argument3);      //font allignment
draw_set_font(argument4);    //Make sure this font is created
draw_set_color(argument5);   //border colour
draw_text(xx-1,yy-1,text);
draw_text(xx+1,yy-1,text);
draw_text(xx-1,yy+1,text);
draw_text(xx+1,yy+1,text);
draw_set_color(argument6);   //main text colour
draw_text(xx,yy,text);
//draw_bordered_text(text,xx,yy,allignment,font,color1,color2);
```

For example, place the following in a draw event:

```
draw_bordered_text("abcd",x,y+20,fa_center,fnt_word,c_red,c_black);
```

Text Menu System

A simple text based menu system.

For this example you'll need two sounds, snd_beep1 and snd_beep1, and two fonts, fnt_title and fnt_menu.

Create Event

```
title = "Choose";
options[0] = "Play Game";
options[1] = "Exit";
selected = 0;
```

Begin Step

```
if(keyboard_check_pressed(vk_up) || keyboard_check_pressed(vk_down)){
    //if the player presses on up or down, move the cursor
    //this works because there are only 2 options
    selected = 1-selected;
    audio_play_sound(snd_beep1, 0, false);
}
if(keyboard_check_pressed(vk_space) || keyboard_check_pressed(vk_enter)){
    //if the player presses on enter or space ...
    audio_play_sound(snd_beep2, 0, false);
    if(selected == 0){
         /*room_goto(rm_lvl1);* change as neeeded*/
    }else{
        game_end();
    }
}

draw_set_font(fnt_menu);
var strW = string_width(options[0])
```

```
if(mouse_y>256 && mouse_y<256+string_height(options[0]) &&
    mouse_x>(room_width-strW)/2 && mouse_x<(room_width+strW)/2)
{
    //if the cursor is above the first option, select it
    if(selected != 0){
        selected = 0;
        audio_play_sound(snd_beep1, 0, false);
    }
    //if the player clicks, new game
    if(mouse_check_button_pressed(mb_left)){
        /*room_goto(rm_lvl1);* change as neeeded*/
        audio_play_sound(snd_beep2, 0, false);
    }
}

strW = string_width(options[1])
if(mouse_y>352 && mouse_y<352+string_height(options[1]) &&
    mouse_x>(room_width-strW)/2 && mouse_x<(room_width+strW)/2)
{
    //if the cursor is above the first option, select it
    if(selected != 1){
        selected = 1;
        audio_play_sound(snd_beep1, 0, false);
    }
    //if the player clicks, end game
    if(mouse_check_button_pressed(mb_left)){
        game_end();
        audio_play_sound(snd_beep2, 0, false);
    }
}
```

Turret Rotating

A different approach to rotating turret

This example assumes have a turret sprite pointing right. Obj_enemy is the parent of your enemies. obj_bullet is your bullet object (an example for this is also in this book)

Create Event

```
turn_speed = 0;

sight_dist = 200; // Length of the field of view

sight_angle = 15; // Width of the field of view

can_shoot = true;

firespeed=20;
```

Alarm 0 Event

```
can_shoot = true;
```

Step Event

```
var target, diff, bullet_id;
target = instance_nearest(x, y, obj_enemy);
if (instance_exists(target))
{
turn_speed += sign(angle_diff(point_direction(x, y, target.x, target.y),
image_angle)) * 0.2;
      if (abs(turn_speed) > 3) turn_speed = sign(turn_speed) * 3;
}
else
{
    // Make it stop when there is no enemy
    turn_speed -= sign(turn_speed) * 0.2;
}
image_angle += turn_speed;

// Shoot a bullet if an enemy (not necessarily the nearest one) is in a
sight
if (can_shoot)
{
    // Find an enemy inside the field of view
    target = noone;
    with (obj_enemy) {
    if (point_distance(other.x, other.y, x, y) < other.sight_dist
         && abs(angle_diff(point_direction(other.x, other.y, x, y),
other.image_angle)) < other.sight_angle) {
            target = id;
            break;
        }
    }
    // If there is one, shoot a bullet to it
    if (instance_exists(target))
{
bullet_id = instance_create(x + lengthdir_x(20, image_angle), y +
lengthdir_y(20,image_angle), obj_bullet);
       bullet_id.direction = image_angle;
       bullet_id.speed = 2;
       bullet_id.target = target;
       can_shoot = false;
       alarm[0] = firespeed;
    }
}
```

Turret Rotate System 2

An easy to way to a rotating turret

This examples assumes the turret sprite is pointing to the right.

Basic Version

Step Event

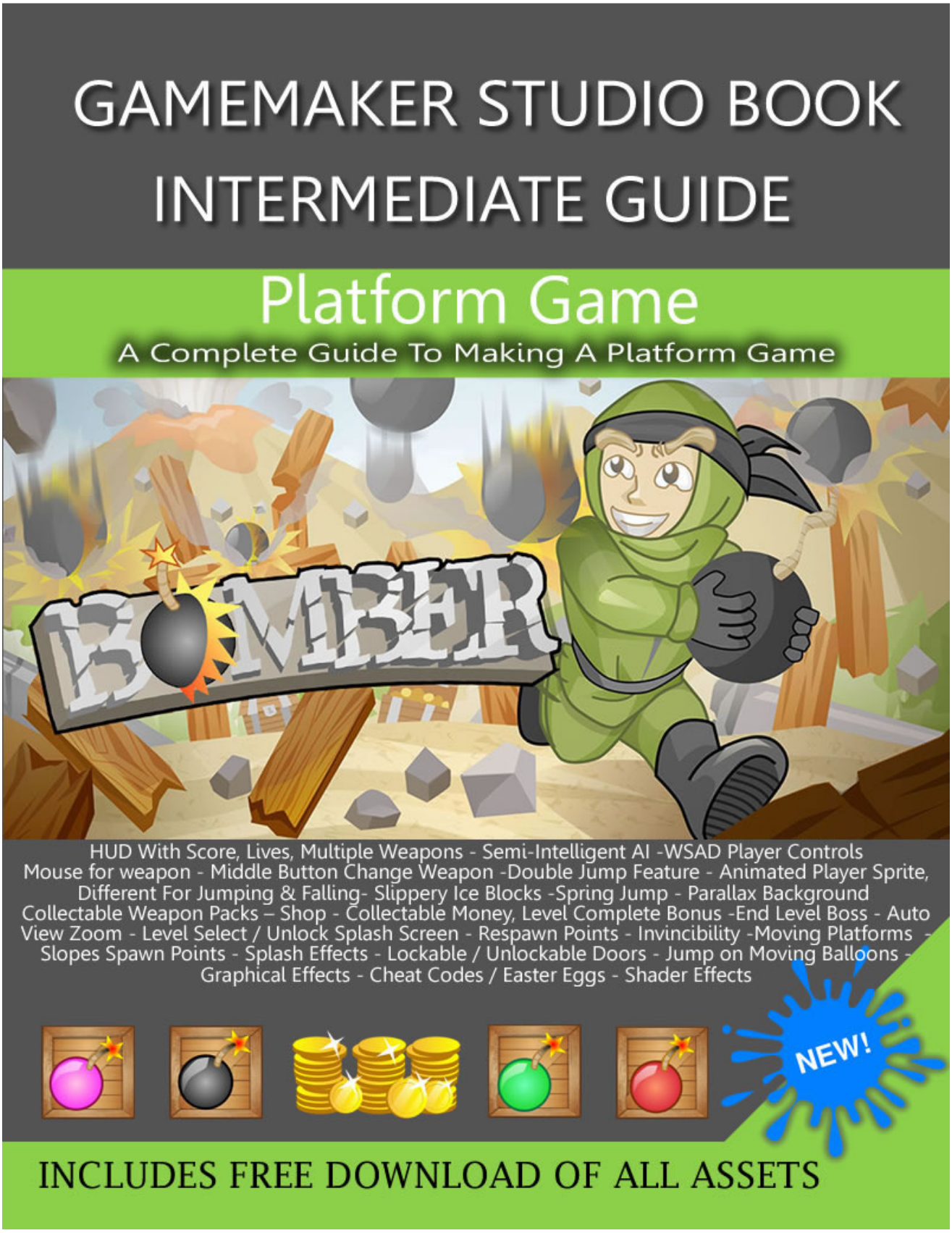
```
if distance_to_object(obj_enemy) < 100
{
     image_angle = point_direction(x,y,obj_enemy.x,obj_enemy.y);
}
```

Advanced Version

Uses two sprites, a base (circular) and a turret.

Object_base Create Event

```
turret = instance_create(x, y, obj_turret);
turret.base = id;
image_angle = random(360);//direction to point
```



Object Turret Step Event

```
if (!instance_exists(base)) {
    instance_destroy();
    exit;
}
x = base.x;
y = base.y;
// angle towards which rotation is being done:
target_angle = point_direction(x, y, mouse_x, mouse_y);
// turret base angle:
base_angle = base.image_angle;
// find difference between target and base angle, and limit it //(change
values to limit turning)
delta = max(-180, min(180, angle_difference(target_angle, base_angle)));
// rotate towards computed angle:
image_angle = angle_rotate(image_angle, base_angle + delta, 2);
```

Script: angle_difference

```
  return ((((argument0 - argument1) mod 360) + 540) mod 360) - 180;
```

Script: angle_rotate

```
return argument0 + median(-argument2, argument2, angle_difference(argument1,
argument0));
```

Typewriter Effect

This example draws text a single letter at a time.

Create Event

```
var text_to_write, typewriter_out,l;

text_to_write = "This is an example text. ##Just change to the text you
want.";

typewriter_out = "";

i = 1;

alarm[0] = 1;
```

Alarm Event

```
typewriter_out += string_copy(text_to_write, i, 1);
i += 1;
if ((i - 1) != string_length(text_to_write)) alarm[0] = 2;
```

Draw Event

```
draw_text(x, y, typewriter_out);
```

Unlock System

An easy to way to create a level unlock system.

The currently available levels is the value global.levels, ie a value of 2 would mean levels 1 and 2 unlocked. This needs to be declared before adding the following code, for example in a splash screen at the start of the game or at the end of completed level.

This example uses a sprite with 2 subimages,' image 0' unlocked, 'image 1' locked, the sprite origin is the center.

Create event

```
my_id=1;
locked=true;
subimage=1;
image_speed=0;
///check if allowed
if (global.level <= my_id-1)
{
    locked=true;
    subimage=1;
}
else
{
    locked=false;
   subimage=0;
}
```

Left Button Released Event

```
///go to level(room) if unlocked

if locked=false

{
    room_goto(room_level_1);
}
```

Draw Event

```
draw_sprite(spr_levels,subimage,x,y);

draw_set_halign(fa_center);

draw_text(x,y,string(my_id));
```

Just duplicate the objects as needed, changing the value of my_id to the level number, also change the code in the left button released event to go to the level you want.

Code

`hspeed*=-1;` Alarm Event

Can also be used with vspeed. Just put it in the event you want, like mouse button or collision event.

Code

`hspeed*=-1;` Alarm Event

Can also be used with vspeed. Just put it in the event you want, like mouse button or collision event.

Weapon Array

An example for storing info for multiple weapons. This example assumes you have the required graphics and sound effect refenced in the code below.

Create Event

```
weapon_no=1;//handgun
global.weapon_info[weapon_no,1]="Hand Gun";//Weapon Name
global.weapon_info[weapon_no,2]=100;//Starting Number
global.weapon_info[weapon_no,3]=1;//Strength
global.weapon_info[weapon_no,4]=1;//Reload Speed
global.weapon_info[weapon_no,5]=1;//Cost
global.weapon_info[weapon_no,6]=5;//Aiming Speed
global.weapon_info[weapon_no,7]=spr_handgun;//Gun Sight for Weapon
global.weapon_info[weapon_no,8]=obj_damage_handgun;//Damage Area for Weapon
global.weapon_info[weapon_no,9]=snd_handgun;//Sound When Firing
global.weapon_info[weapon_no,10]=snd_voice_handgun_selected;//snd of voice
weapon selected
global.weapon_info[weapon_no,11]=20//ammo pack size
weapon_no=2;//rifle
global.weapon_info[weapon_no,1]="Rifle";//Weapon Name
global.weapon_info[weapon_no,2]=25;//Starting Number
global.weapon_info[weapon_no,3]=2;//Strength
global.weapon_info[weapon_no,4]=2;//Reload Speed
global.weapon_info[weapon_no,5]=2;//Cost
global.weapon_info[weapon_no,6]=3;//Aiming Speed
global.weapon_info[weapon_no,7]=spr_rifle;//Gun Sight for Weapon
global.weapon_info[weapon_no,8]=obj_damage_rifle;//Damage Area for Weapon
global.weapon_info[weapon_no,9]=snd_rifle;//Sound When Firing
global.weapon_info[weapon_no,10]=snd_voice_rifle_selected;;//snd of voice
weapon selected
global.weapon_info[weapon_no,11]=15//ammo pack size
//continue as required…
```

Step Event

```
if (keyboard_check_released(ord("1")))
{
   global.weapon=1;
     sound_play(global.weapon_info[global.weapon,10]);
}

if (keyboard_check_released(ord("2")))
{
   global.weapon=2;
     sound_play(global.weapon_info[global.weapon,10]);
}
// continue as required
```

Example Code to shoot:

```
if (global.weapon_info[global.weapon,2]>0 )

{
     sound_play(global.weapon_info[global.weapon,9]);
     global.weapon_info[global.weapon,2]-=1;
     instance_create (x,y,global.weapon_info[global.weapon,8])
}

else

{
     sound_play(snd_voice_no_ammo)
}
```

Example Code For Shop

```
if (keyboard_check_released(ord("1")))
{
   global.weapon=1;
     if (score>global.weapon_info[global.weapon,5]) or
     (score==global.weapon_info[global.weapon,5])
    {
        //take money off score
        score=score-global.weapon_info[global.weapon,5];
        //increase ammo no by one

     global.weapon_info[global.weapon,2]=global.weapon_info[global.weapon,2
     ]+1;
```

```
//play sound purchase complete
        sound_play(snd_voice_purchase_complete);
    }
    else
    {
    //play not enough cash
    sound_play(snd_voice_not_enough_cash);
    }
}
```

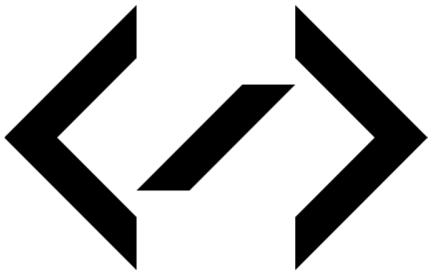
Code to Draw Weapon info:

```
draw_sprite(global.weapon_info[global.weapon,7],0,x,y);
draw_text(25,360,global.weapon_info[global.weapon,1]);
draw_text(220,360,"Ammo:");
draw_text(350,360,global.weapon_info[global.weapon,2]);
draw_text(450,360,"Damage:");
draw_text(590,360,global.weapon_info[global.weapon,3]);
```

Example code for Firing Weapon (left mouse pressed)

```
if (global.weapon_info[global.weapon,2]>0 )
{
    sound_play(global.weapon_info[global.weapon,9]);
    global.weapon_info[global.weapon,2]-=1;
    instance_create (x,y,global.weapon_info[global.weapon,8])
}
else
{
    sound_play(snd_voice_no_ammo)
}
```

While



That is the basic structure of “while”:

```
while(statement 1)
{
  code;
}
```

First, the statement 1 is checked. If the statement is true, the code is executed and then the statement is tested again. If the result is false, the loop ends. If it is true the code executed again, and so on until the statement is false.

```
var str;
str=string(score);
while(string_length(str)<9)str='0'+str;
draw_text(8,8,str);
```

In this example, we try draw a score with 9 digits, so if the score has less than 9 digits (like 10 or 125), the while loop add zeros in the left side of the string, and after finishing the looping, the score is drawn at position (8,8).

Second example:

```
while(place_meeting(x,y, obj_ball))
{
  x=irandom(100);
  y=irandom(100);
}
```

In this case, the while loop only ends when is found a position that doesn’t overalp with the obj_ball, in the 100x100 region.

Windows Shake Effect

This script makes the room shake.

Script: window_shake

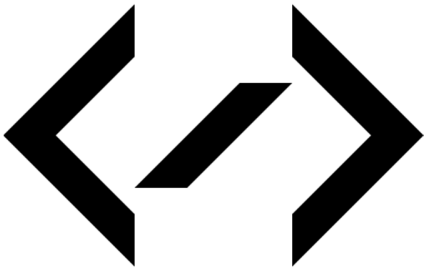
```
D = argument1; //Get The Duration
A = argument0; //The Shake Amount
curX = window_get_x(); //The Current Xpos of the window
curY = window_get_y(); //The Current Ypos of the window

if window_get_fullscreen() = true then exit;

repeat(D) //Repeat Until Time Is Over
{
   window_set_position(curX+random(A),curY+random(A));
   window_set_position(curX-random(A),curY+random(A));
   window_set_position(curX+random(A),curY-random(A));
   window_set_position(curX-random(A),curY+random(A));
}
window_set_position(curX,curY);
```

Example Usage:

Global Mouse Left Pressed

```
Window_shake(15,10);
```

With



That is the basic structure of “with”:

```
with(expression)
{
    code;
}
```

the “expression” can be a instance id, an object’s name (that will consider all the instance of this object) or the special keywords **all**, **other**, and **self**. So in each of the indicated instance in the “expression” , the code is execute. One thing to note is that inside of the structure with, the instance that contains with is referred as **other**, and the called instance as **self**. For example

One simple example:

```
with(instance_create(x,y,obj_bullet))
{
    speed=8;
    direction=other.direction;
}
```

In this example, the obj_bullet created has the direction set as direction of the calling instance.

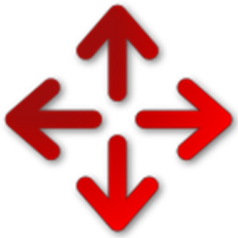
Second example, where the object obj_store_slot has the variable selected=false:

```
if(selected==false){

  with(obj_store_slot){selected=false; image_index=0;}

  selected=true;

}
```

So if this code is put in the event **Left Pressed** , all the obj_store_slot instances are selected=false and the only the one clicked has selected=true. In this one, you can reset any obj_store_slot that had selected=true before.

Wrap Around Room

This funky bit of code makes the object wrap in the room, that is, if it goes off top, bottom or sides of the screen it will come back in from the opposite side.



In the object you want to wrap, place the following in the outside room event (Other>>Outside Room)

Code for Outside Room Event

```
move_wrap(1,1,1);
```

This Book Is Available Now as Paperback or PDF at: www.GameMakerBook.com