

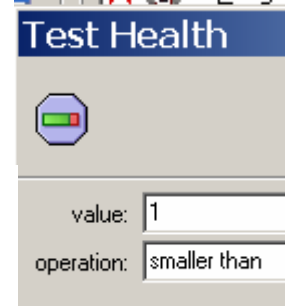
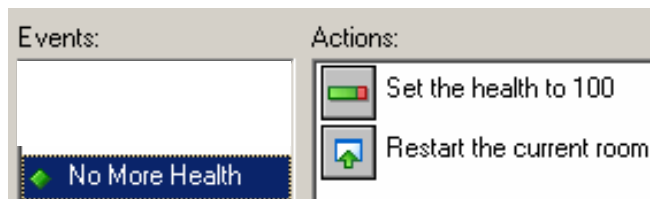
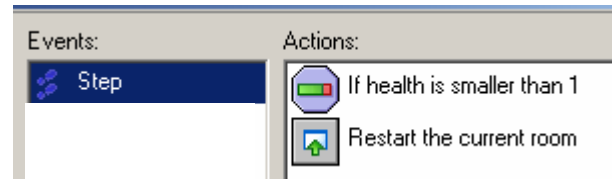
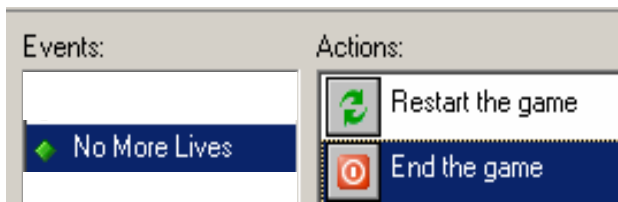
For **Actions:** in the Main2 tab select **Show the Game Information.**



For this Event to be relevant you must firstly have written the game information. This usually contains the “How to Play “ information

Paths

You can create specific paths for the objects to follow but first you need to create them first.



Game Loading Image

A more professional looking game :

Show your own image when loading the game:

Select **“Global Game Settings”**
 “Loading” Tab, select Show your own image when loading



Showing the Game Information

In the Controller Object

Select **Event** -> **Other**

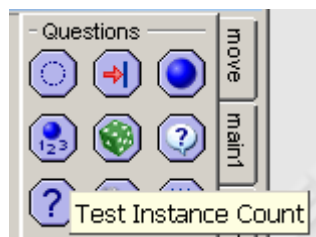
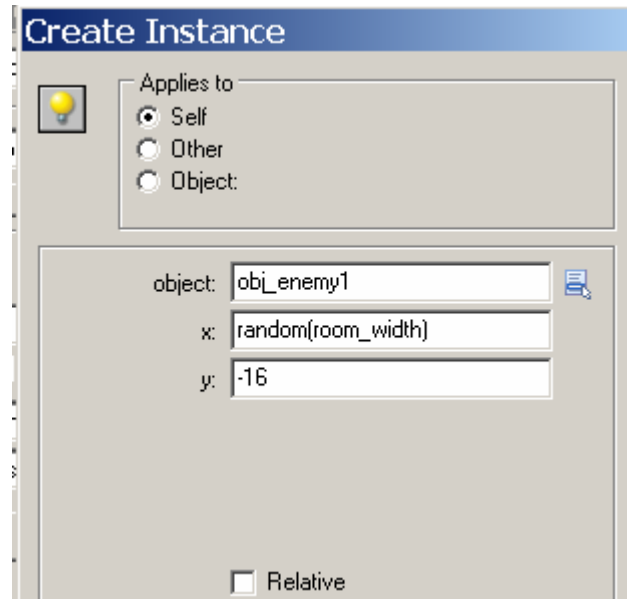


Select – **Game Start**

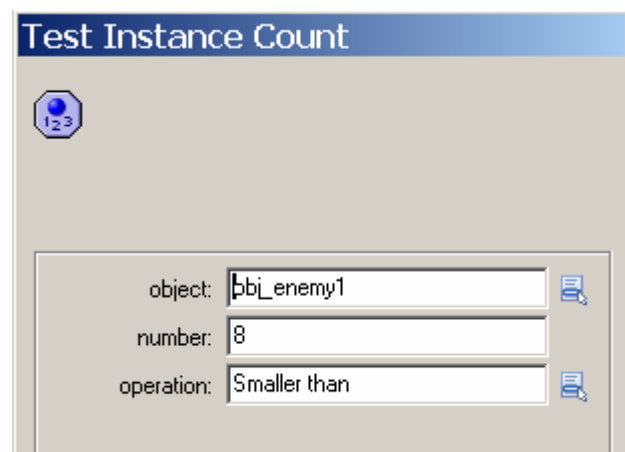


Create an instance at **random(room_width)**.

This will cause the object to appear anywhere across the room at y -16



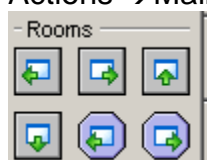
Ask a question Action -
Check the number of objects In this example are there less than 8 enemies?



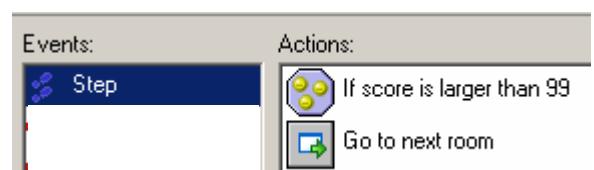
Go to next room

You will have to decide what Object -> Event will cause this Action. Usually it's connected to collisions, points , lives . As these are inter-dependant you will have to set up how points/lives etc are awarded .

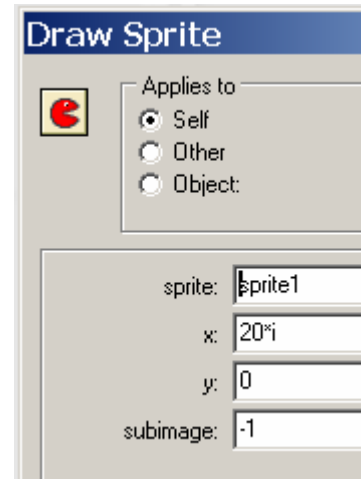
The rooms transition Actions can be found in the Actions ->Main 1 tab



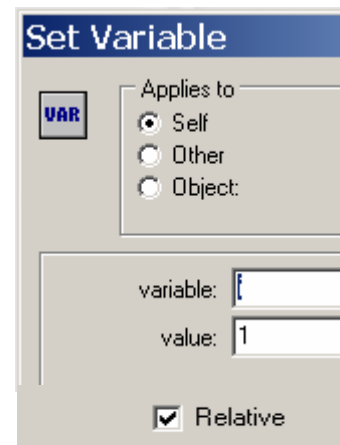
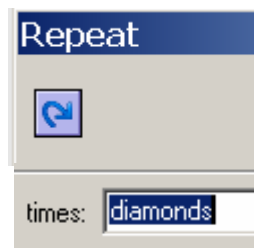
Here are some examples:



set variable i to 0
 repeat next action (block) diamonds times
 at position (20*i,0) draw image -1 of sprite sprite1



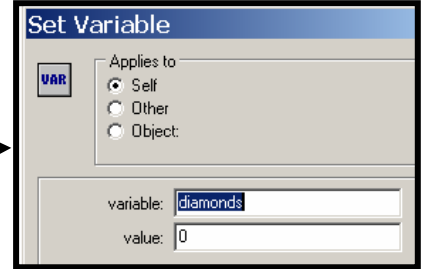
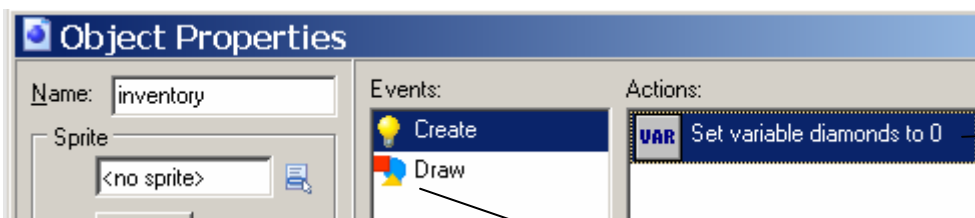
set variable i relative to 1
 The sprite is drawn "diamonds" times
 i is increased each time it is drawn, it is drawn further to the right
 each time as i increases



Collision Event with object:

set variable inventory. diamonds relative to 1
 for other object: destroy the instance
 diamonds lives inside the inventory object, when referred to inside inventory, you can just call it diamonds, when referred from another object, you must use the full name inventory. diamonds



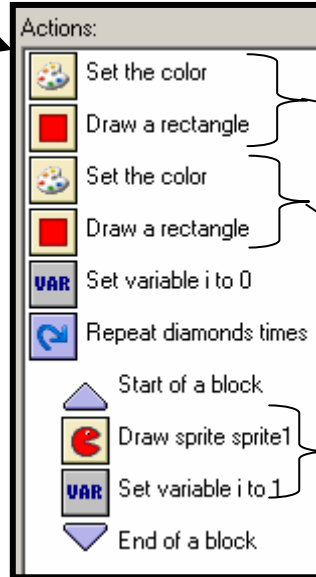


[See the Inventory Game example](#)

This OBJECT : Inventory

Causes a drawing EVENT that draws a white square and places diamonds (sprite - object) in the square when a point is scored

[Inventory eg game sample](#)



These two actions cause a white filled box to be drawn at your set X-y co ordinates

These two actions cause a black line to be drawn around the white box.

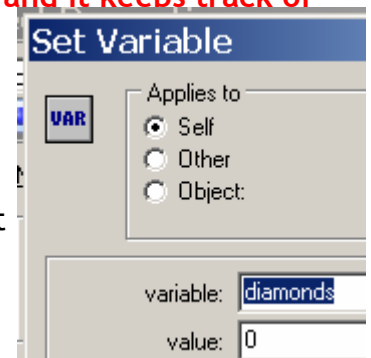
These icons draw the diamonds icons inside the white bounded box according to the number of diamonds you have collided with

The object "inventory" is used to keep track of diamonds, the variable "diamonds" is the number of diamonds collected.

A Variable is just another way of keeping count of things, it's a little like a score but you can nominate what it keeps count of. Before you can use a Variable you need to create it and give it a name. In this example the name of the variable is "diamonds" and it keeps track of the diamond objects that the bear collides with.

Create Event:

set variable diamonds to 0

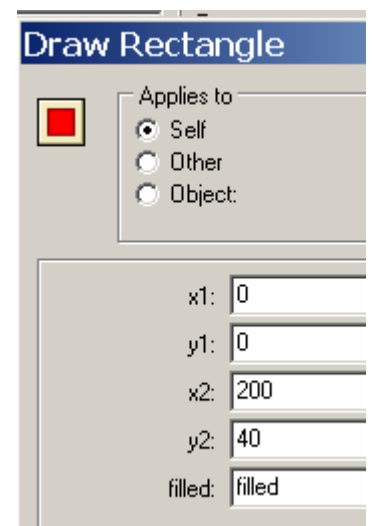
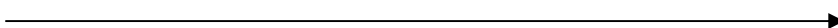


When you collide with the diamond it is destroyed and the inventory count increases by 1 and is drawn inside the white bounded box

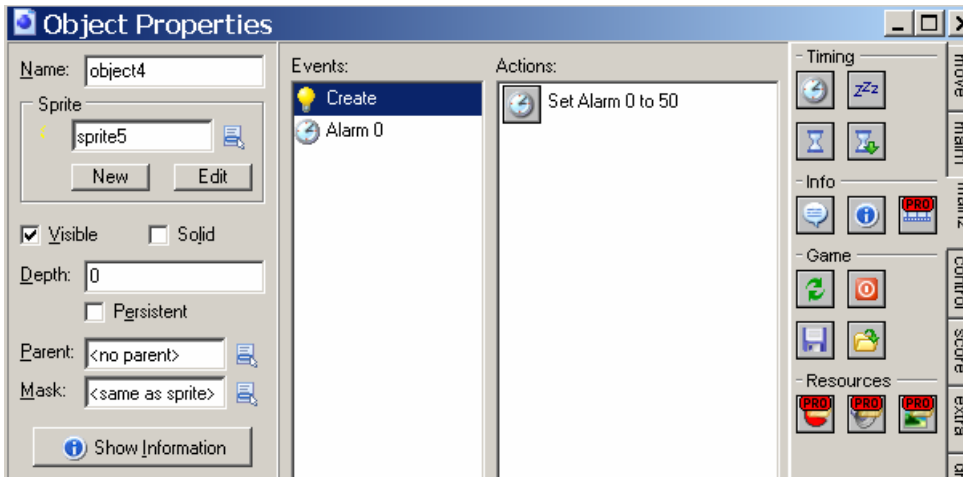
In the draw event for inventory, a box and the collected diamonds are drawn

Draw Event:

set the fill color to 16777215 and line color to 0
draw rectangle with x-y co-ordinates (x1=0,y1= 0) and (x2=200,y2= 40)
A white box is drawn



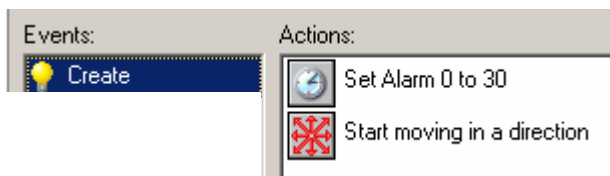
Set Alarm



Setting an alarm then referencing it in the Events allows Actions to happen at regular time intervals. There are 12 Alarm clocks, match the Alarms to their pairs ie: Alarm 0 to Alarm 0.

You must Set the alarm before you can have an Alarm Event

If you use an alarm you need to set it first in Actions. Then you need it to “ring” in Events. You will also need to re set it if you want it to keep happening say every 10 seconds.



Create the alarm and set it to 30
Note Alarm 0 is paired with Alarm 0 in the Events.



Then set the Alarm to ring

Object - Inventory

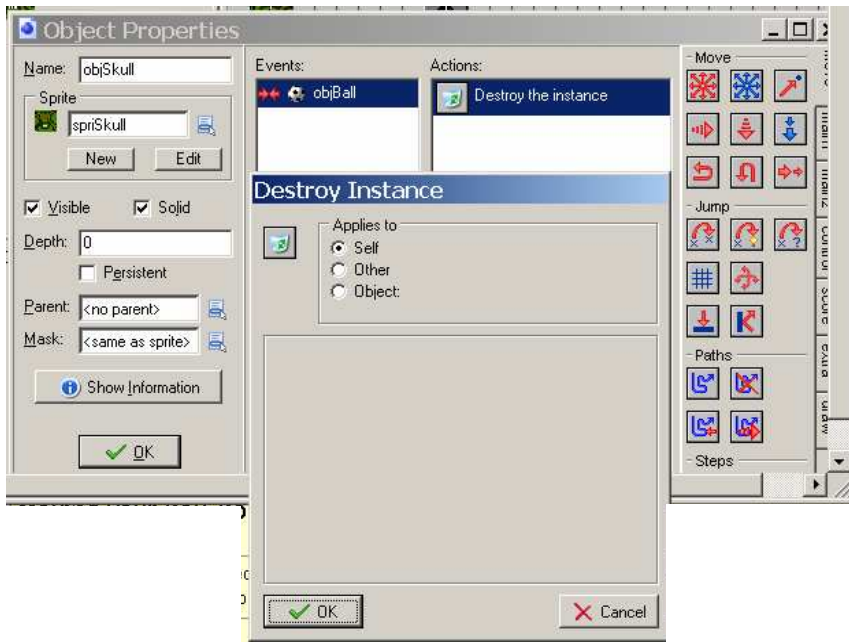
Sometimes, depending on the complexity of your game you will have a Controller object and an Inventory object. When you wish to score objects like treasure you use an Inventory object to draw the box on the screen for the objects and cause them to be collected.

Most objects have sprites attached to them so that you can see them in the game. There are some exceptions to this , these are invisible (no sprite) objects, the Inventory object is an example of this

[For full instructions on how to create a Controller Object for showing scores, health and lives click here](#)

Inventory - Used for keeping track of items you have won , a type of score

Some Handy Information about Coding Using Game Maker



Events & Actions

Events cause Actions to happen. eg

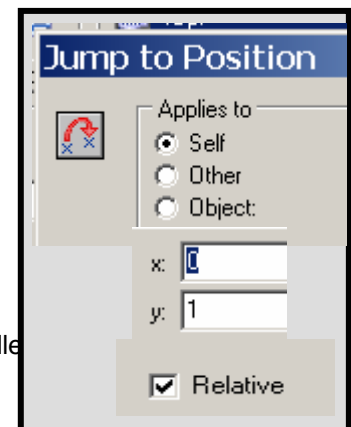
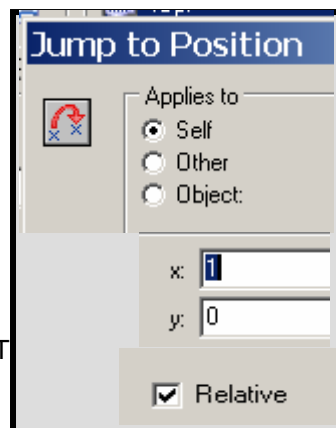
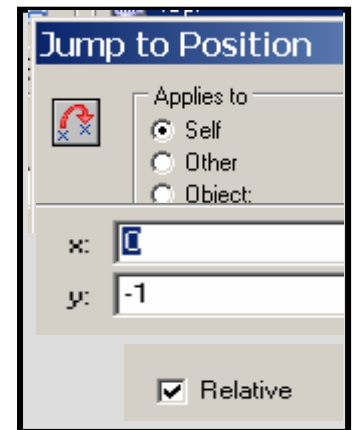
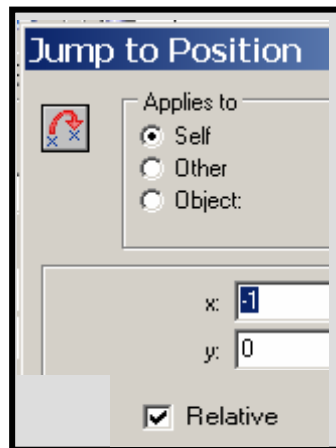
Destroy an object

Event : Collision of the skull object with the ball object

Action: Destroy an object The skull is destroyed by the ball



Move the object left , right , up down with keyboard arrows: For the Object that you wish to move select Event Keyboard, then->



The player scores lives by	<input type="text"/>
The player loosed lives by	<input type="text"/>

The player scores health by	<input type="text"/>
The player loses health by	<input type="text"/>

The player scores inventory by	<input type="text"/>
The player loses inventory by	<input type="text"/>

The game ends when the	<input type="text"/>
The room re-starts when	<input type="text"/>

The room (Level) moves up to the next room (Level) by :

You must write your game instructions first .

Your game should also include :

Showing your own images as your Game Maker game loads.

An opening room

An ending room

My Game

General outline of my game, the action that happens and it's setting/s (rooms/Levels).

My game could be used to teach players to :

Characters in my game: (these will be your sprites-→Objects) Cross out when not applicable

This Character represents the player	<input type="text"/>
And character throws this missile	<input type="text"/>

This Character represents the target	<input type="text"/>
This character throws this missile	<input type="text"/>

The Levels:

The first room (Level) setting is :

Other actions and objects (and sprites) in my game are :

Scoring

The player scores points by	<input type="text"/>
The player loses points by	<input type="text"/>

9 **scripts**: small pieces of code that can be used to extend the possibilities of your game.

Designing Your Game

Like a screenplay your game play is what happens in the game and why it happens. You should sketch storyboards to illustrate your game idea. Your game characters are like actors and the rooms in your game are like scenes from a movie. Try to give your game a unifying theme and some purpose to the action.

You might like to think in the following terms:

- what sort of audience (young kids, teenagers, teachers) the game is for
- general background idea for the plot of the game
- the characters, their characteristics, and what they do
- the levels available, what they represent, and how to change from one to another
- How you score points, lives etc.

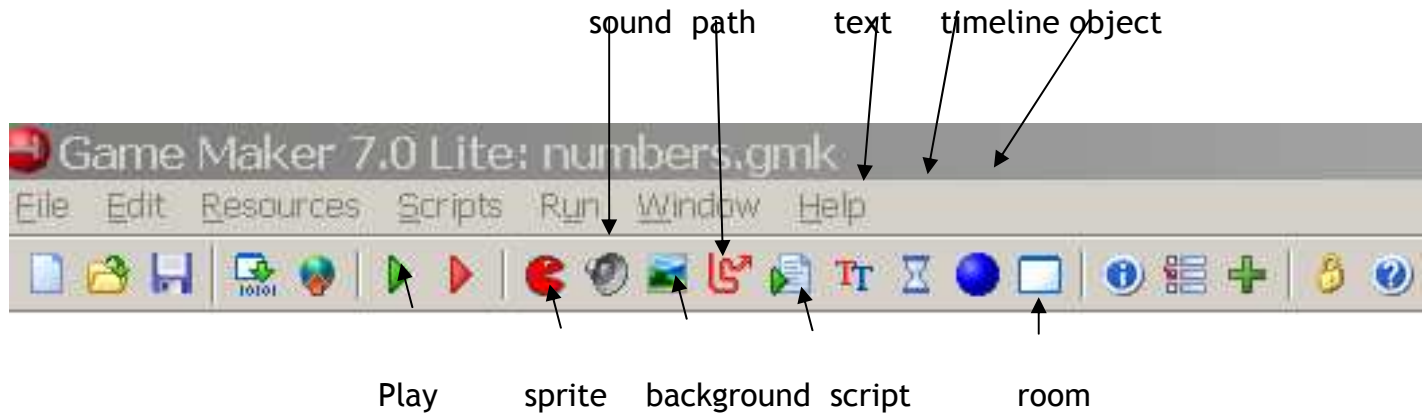
Games can be used to influence the players behavior or attitudes. Games can also be used to teach skills such as typing, math's etc. e.g. Nintendo Brain Training.

Before starting creating your game using Game Maker, complete a Game-Plan on the next two page.

It may prove helpful to read the [Mind Games instruction](#) and the [How to Show Scores](#) etc. instructions as well as this document

This document provides some handy general coding hints

Starting out with Game Maker



1 sprites: images that are used to show the objects.

2 objects: the things in the game. In your room you put **objects**. Objects are the things in your game such as balls, walls, characters etc. Some objects just sit (like walls) and other objects move around and react to things that happen. **Objects have Events and Actions** connected to them. Not all objects have a sprite, some objects such as Controller, Inventory, Timer etc are used to control the action in the game and are placed in the room and represented by a blue ball



3 backgrounds: the images used as background for the rooms. Rooms have a **background**. This could be a plain colour or an image. You can move from room to room during your game. Backgrounds can be tiled (repeating in the room) or a single background stretched to fill the room.

The objects cannot react to the background

4 rooms: the scenes (levels) in which the objects live and where you place the background

5 sounds: these can be used in games, either as background music or as effects. They must be created first before you can use them.

6 events

Things that happen are called **events**. An event could be the player pressing a key on the keyboard or it could be your game character colliding with a wall.

7 actions

When events happen, your objects can take **actions**

8 instances: one particular occurrence of an object

There can be more than one **instance** of an object in a game. For example we might have multiple instances (lots of monsters) of a monster object in a game. When we talk about the monster object we mean all the instances of the monster in the game. When we talk about the instance we mean one particular instance of the monster.