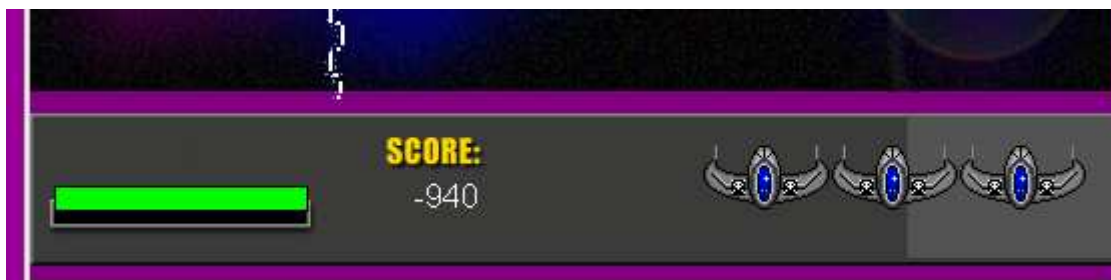


Configure the Test Score Action -

This action asks “Is the score more than 99”, if the answer is yes then the game will move to the next room, if no, you keep playing the current room



Provided that you have followed these directions your game should now have a fairly professional looking score bar along the bottom. You can download my sample “SpaceRocks” to see the completed example and check all my coding.



What do you want to happen when all lives are lost? Points get to #number, Health is zero?

Step 9

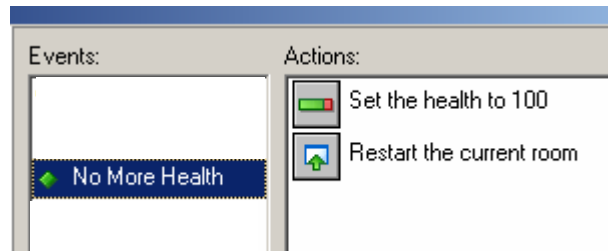
Controller Object

Restarting the room if you have no health:

Select Object : Controller



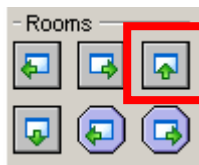
No More Health



Actions

From the **Score tab** : Set health to 100. This re sets the health to 100 again

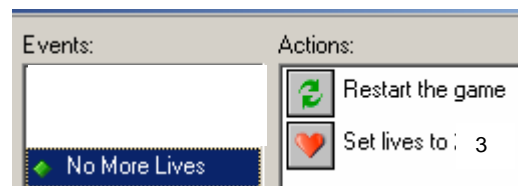
From the **Main1 Tab** : Restart the current room



And



No More Lives



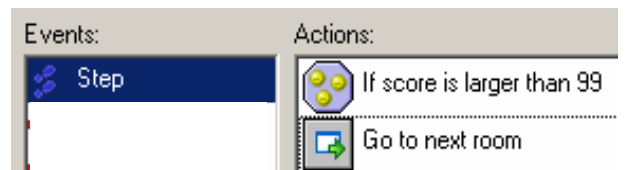
Actions

From the **Score tab** : Set lives to 3. This re sets the lives to 3 again

From the **Main2 Tab** : Restart the game

This example causes the room to move up a level when 99 points are reached.

Its created for the Missile Object and cross references the points earned for the missile colliding with the target that you have already coded.

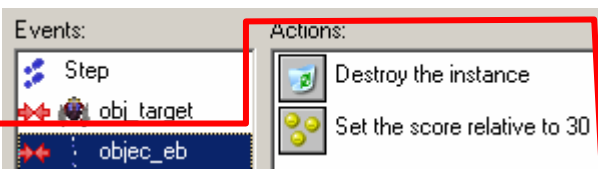


For **EVENT**, select **STEP**



Actions – From the **Score tab** select **TEST SCORE**

From the **Main1 tab**, select **Go to next room**

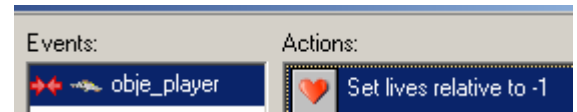


Step 8

For your OBJECTS , EVENTS : Allocate a Set Score, Set Lives, Set Points action for the

Examples:

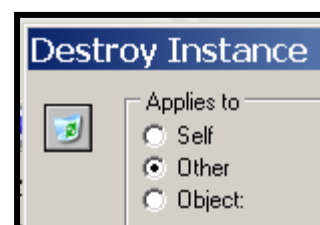
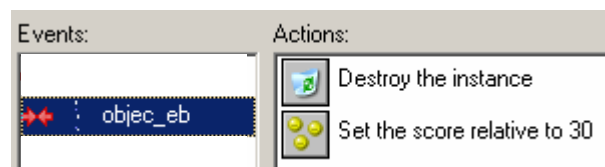
In this example when the Object enemy bullet collides with the player object you loose a life



In this example when the Object missile collides with the target you gain 10 points



In this example when the Object missile collides with the enemy bullet object you gain 30 points and destroy the enemy bullet

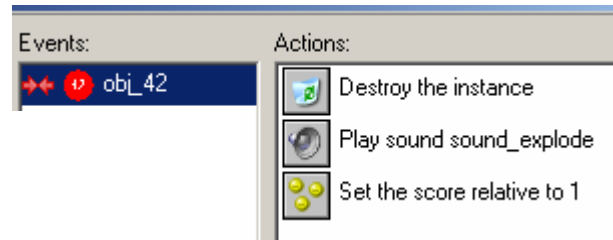


Gaining Points for shooting the right target:

Select **OBJECTS**

Select the missile object

Create a collision **Event**, **collision with** the target sprite.



Actions:

From the Main1 tab select **“Destroy the instance of”**, select **“Other”**

From the Score tab select **“Set Score Relative”** action and drag it to the Actions column, select **“relative to 1”**.

Step 7

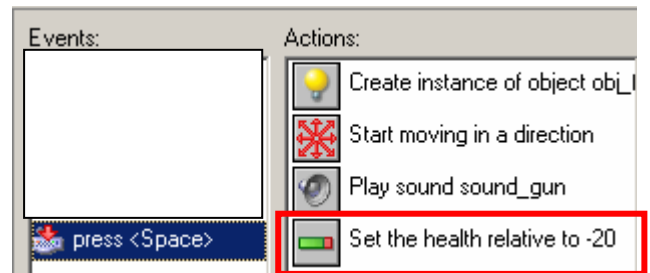
Lose health for shooting missiles.

In this example I am actually using the health bar as a ammunition bar.

In the **Objects**, select the **player object**

For the **Space bar event** add an action:

From the Score tab select **“Set Health Relative”** action and drag it to the Actions column, select **“relative to -20”**.

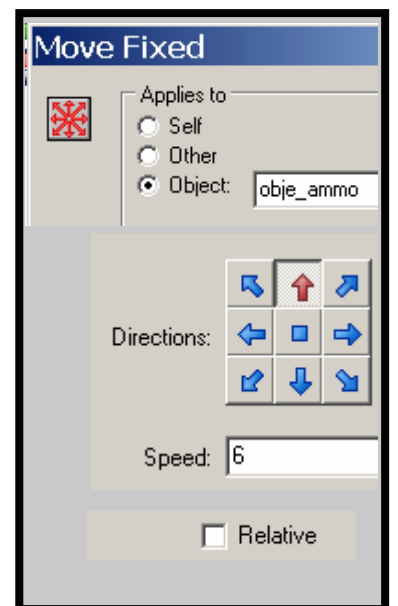
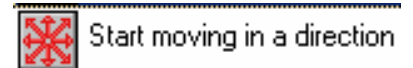
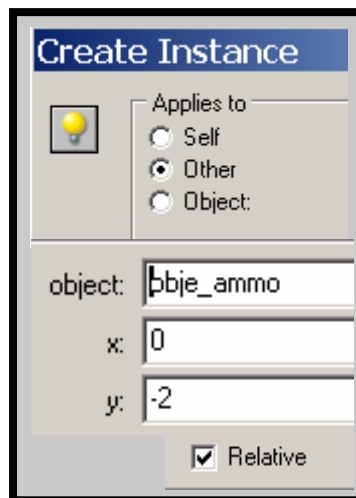
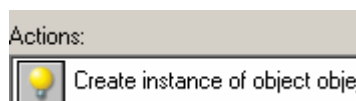


This means that when the space bar is pressed the missile is created and moves but I loose -20 health (ammo) every time I press the space bar.

For the player OBJECT : Other Actions for the space Bar Event - You will also need to create the missile and start it moving

Create the missile so that it springs out of the player

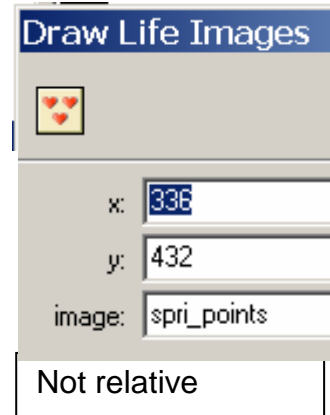
Then cause the missile to move upwards



Draw Lives according to your location on the bar. In my sample it was X = 336 y = 432

If you wish to show lives you will also have to create a sprite that represents each life or you could call it trophies or money, it still requires a sprite and an object that represents it.

Create this "lives: sprite and object if you want it to show.



Step 5

Re-set the room : remove the obj_score and replace it with the controller object as shown in step 3.

This will cause the score bar sprite to disappear and be replaced by the blue ball with the red question mark.

We actually only temporarily made the obj_score (the ribbon at the bottom of the screen where scores, lives etc. will be shown) visible. The controller object will now cause this to be shown on the screen when the game is played, we used it temporarily to help us work out the x and y positions of the scores , lives etc.

Step 6

Decide how the scores, life and health will be allocated.

In my game the missile (or bullet, mine is obj_fire1) will destroy the obj_target
It will play a sound and add 1 to the score.

Your game will be different so you need to have thought about how you gain points, lives , trophies etc and loose these.

For **OBJECTS** : Decide what Event will cause your game to restart the room, restart the game , gain/loose lives, gain/loose points, gain/loose health.

Now we need to code these Actions for each Object and Event

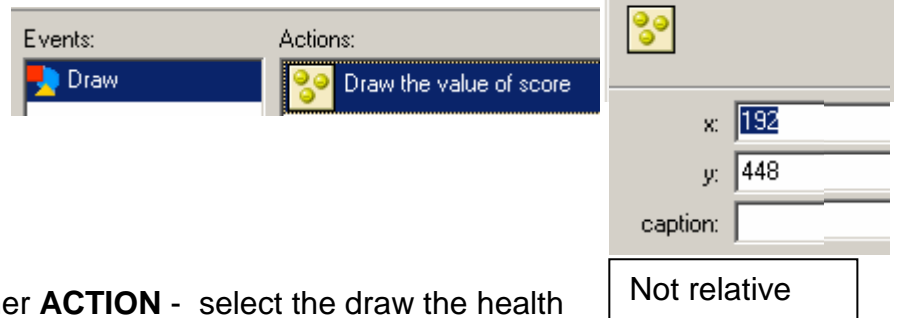
The **y 1** and **y2** positions are for the width of the life bar
 i.e. Life bar :Width top = **y1** Width bottom = **y2**

If you also wish to draw lives (or trophies or whatever) you can also show these on the score bar but you will need to nominate **x** and **y** positions.

Write down the locations for all of these x – y co ordinates , you will need them when you configure the Draw Event – Draw actions

Configure the Draw Event for Action **Draw Score** according to your position
 Check – NOT relative

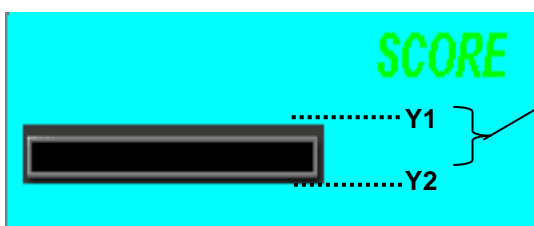
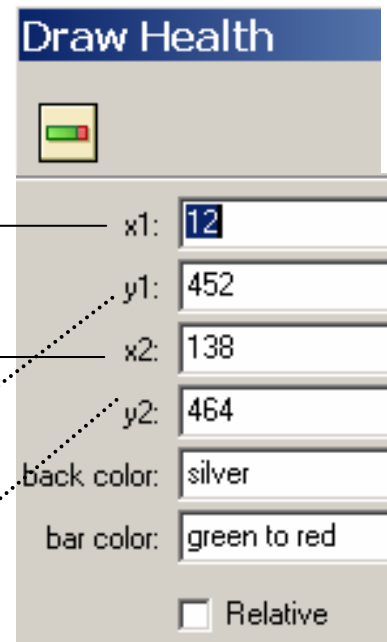
Drawing the health bar:



For the **DRAW EVENT**, add another **ACTION** - select the draw the health bar from the Score tab

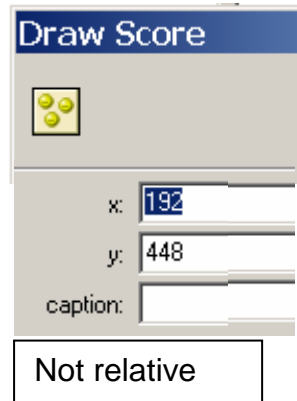
Settings x1 and x2 indicate the start location and stop location for the LENGTH of the health bar. These settings together determine the length of the health bar eg: from location x 16 to 128

Settings y1 and y 2 refer to where the top and bottom (height) of the health bar will start and end. These settings together create the width of the health bar



Height of our sample is 12 i.e. 464 - 452

For the **DRAW EVENT**, add another **ACTION** - select the draw value of the score action from the Score tab



Configure the x and y location settings so that the score will be drawn under the word "SCORE" in the scoring sprite. In this sample for my score bar this position is :

X = 192

Y = 448

For your score bar artwork this x, y position will be different.

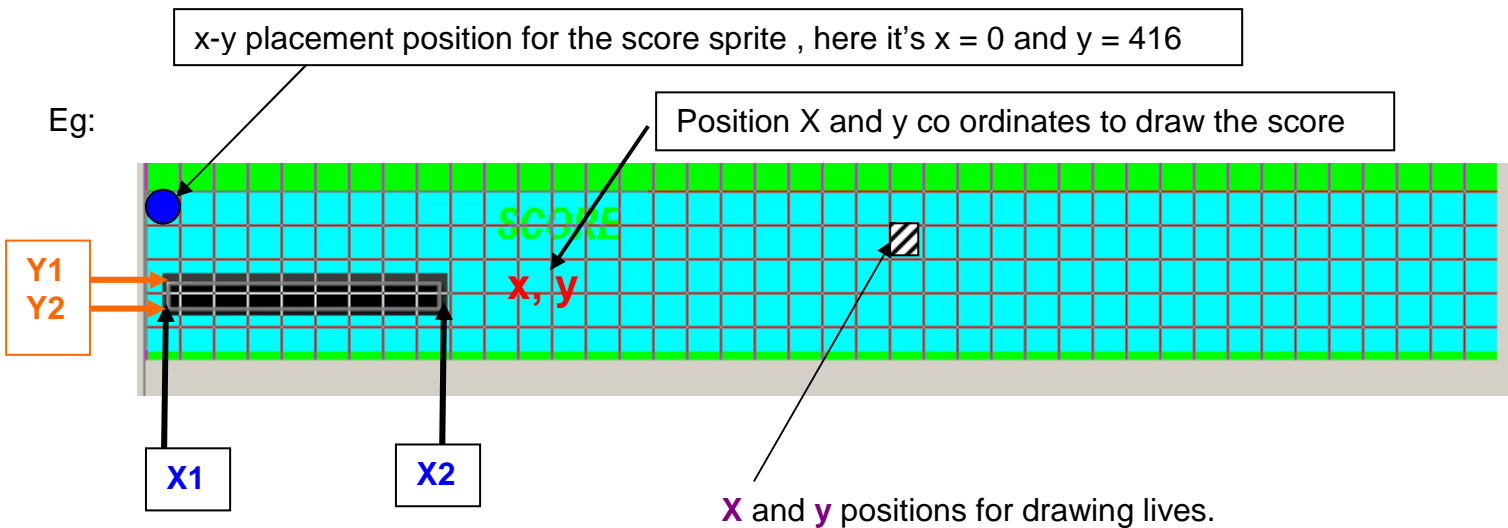
Where exactly? Temporarily placing the score sprite in the room so that you can see it and work out the exact locations for drawing scores, health bar etc.:

Finding the exact position for each of these items is not always easy.

Here's a little trick that temporarily places the score bar in the room so that it is visible and you can work out it's co-ordinates.

- Create a score object and give it the score sprite

Go to **ROOMS**, click **Objects** and temporarily add the **object_score** to the bottom of the room in the position you want it at the bottom of the screen.



Now use your mouse to hover over the game room grid to determine the x and y positions.

In the example above the red **X and y** positions are where the **score** will be drawn at positions **X = 192** and **Y = 448**.

The life bar has 2 x positions and 2 y positions.

The **x1** position is the start of the life bar, the **x2** position is the end of the life bar i.e. Life bar : Length = start : **x1** end: **x 2**

For the DRAW Actions x 4: *This event will draw all the scores, lives and health on your screen*

This is possibly the most difficult element of creating the health bar etc as you have to know exactly the X and y coordinates for each item you want drawn on the screen. Placement, size and length of things such as the health bar will depend on where you drew these on your sprite. Read these instructions through first so you understand how to work out where to draw these elements:

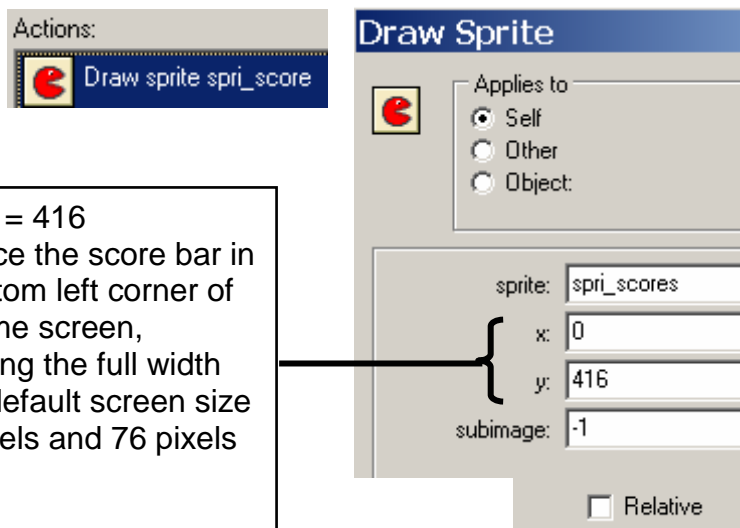
The **Draw Sprite Action** actually causes the score bar sprite to be drawn in the game room where you specify, this is used as a background for the scores, health etc.



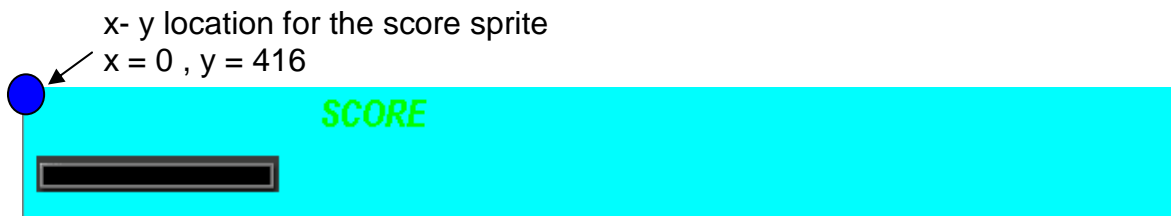
Select the **Draw Sprite icon from the drawing actions tab** and drag it onto the Actions column.



(See step 1 if you haven't created a score sprite)



e.g. of a Score bar sprite



Controller Events and actions

One by one, create each of these **Events** with the following **Actions**

1

The Create Event causes the scores, health and lives to be set

2

3

4

Finally, the Draw Event causes all the information to appear on your game screen.

Setting the correct x and y positions for drawing each of these items is a little tricky

We will use a simple trick to work out their exact positions, read the following before you key in the DRAW Event and Actions.

Event 1: Create

- Set the score to 0
- Set lives to 3
- Set the health to 100
- Set the score caption info
- Play sound snd_background

Event 2: No More Lives

- Show the highscore table
- Restart the game

Event 3: No More Health

- Set the health to 100
- Restart the current room

Event 4: Draw

- Draw sprite spr_bottom
- Set the color
- Draw the value of score
- Draw the health bar
- Draw the lives as image

Score Display Style Sheet:

- background: back_score
- border: don't show
- new color: [Red]
- other color: [Black]
- font: AaBbCcDd

i.e. If you select $x = 0$ and $y = 0$ the top left corner of item will appear in the top left corner of the game screen

The x - y co ordinates for the controller refer to the top left corner of the sprite that will be drawn as the background for lives, scores etc. Remember to allow for the depth (height) of the score sprite.

If you select $x = 0$ and $y = 416$ the item will appear in the lower left corner of the game screen as illustrated above.

In **ROOMS** , click on the **Objects** tab for your first game room and select the **controller object** to add with the left mouse click. In this example we will place it at $x = 0$ and $y = 416$. Left mouse click at these co-ordinates and a blue ball with a red question mark will appear, this places the controller in your room.



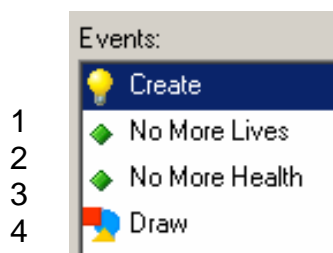
Step 4

Return to the “controller” object. We will now program it’s properties

It’s important that you have decided what exactly you want to happen if all the “health” is used up, lives etc. Think about this before you program these events.

Even though Game Maker calls these items health .life etc, you can rename them on the screen to show things like “ammo” etc.

In the controller Object properties , check “**Persistent**”. This means that your scoring system: actions controlled by the controller, will happen on all levels (rooms) in your game



Step 2



Create Controller Object

Showing scores, lives and health:

Create an object called controller, it has no sprite :

The controller object in this example will control lives, health and scores, it does not have a sprite that represents it in the game as it actually controls several things and causes several things to be shown in the room. As part of the “controller” you will cause the score sprite you created to be drawn.

Step 3

Adding the controller object to your rooms:

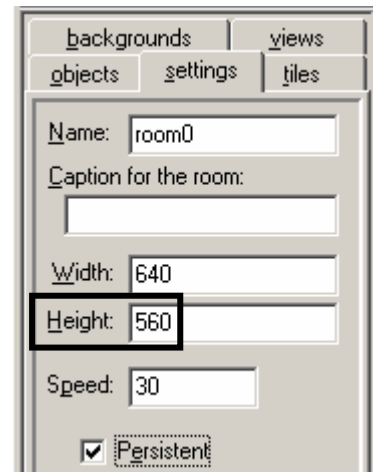
Where to place the controller and draw the score, inventory, health?

Drawing at positions – x and y co-ordinates

The controller cause the scores, lives and health to be drawn at certain co-ordinates on the X and Y axes. Later in the Controller Object – Events, you will be asked where exactly to draw items like the life bar etc. Usually they are drawn along the top or bottom of the room screen so that they don't interfere with the game-play. In this example we will draw the bar at the bottom of the screen so if you have already created your game this should not interfere with your game design and can be added without any changes to the actual game.

Make your Room Larger so that the score_sprite fits in the room

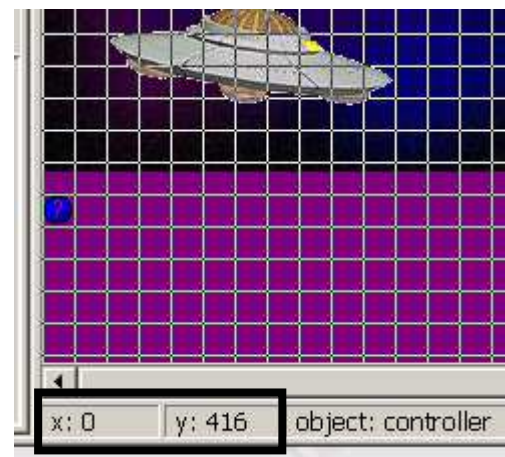
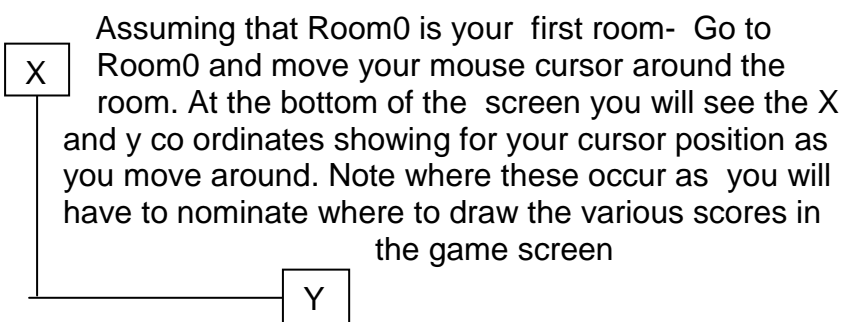
If you have already created your game and the controller is an add-on it's simpler to just make the game room a little deeper to accommodate the health bar etc.



To do this , click on ROOMS
Click the settings tab

Change the **room height from 480 to 560** .This will cause the game room to drop down an additional 80 pixels at the bottom.

Placing the controller on the room



Create Event for making the Enemy appear moving in the room.

Outside room event for the player, enemy and missiles – destroy action. Re create the enemy again in the room, moving, after it has gone out of the room

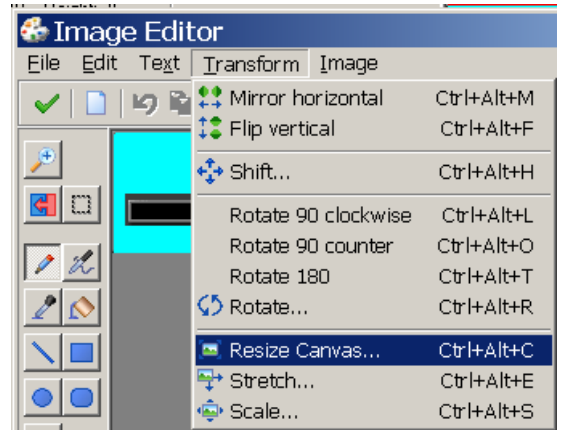
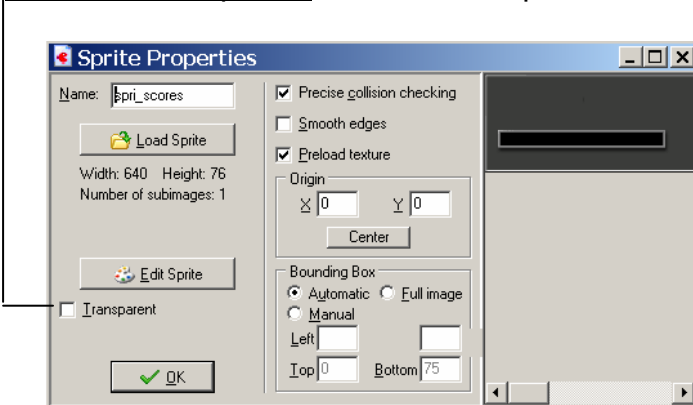
Now For your Scoring System

Step 1

Create a Score Sprite

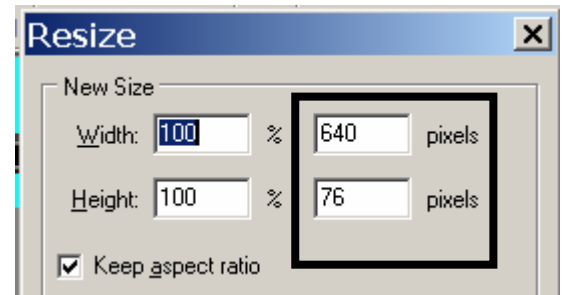
The Score Sprite is used to represent the background for the scores, lives and health. In this example the sprite is quite large, it is as wide as the game room (640 pixels) and 76 pixels high. 640 pixels is the default width of the game room unless you have changed it.

Un-check transparent for the score sprite



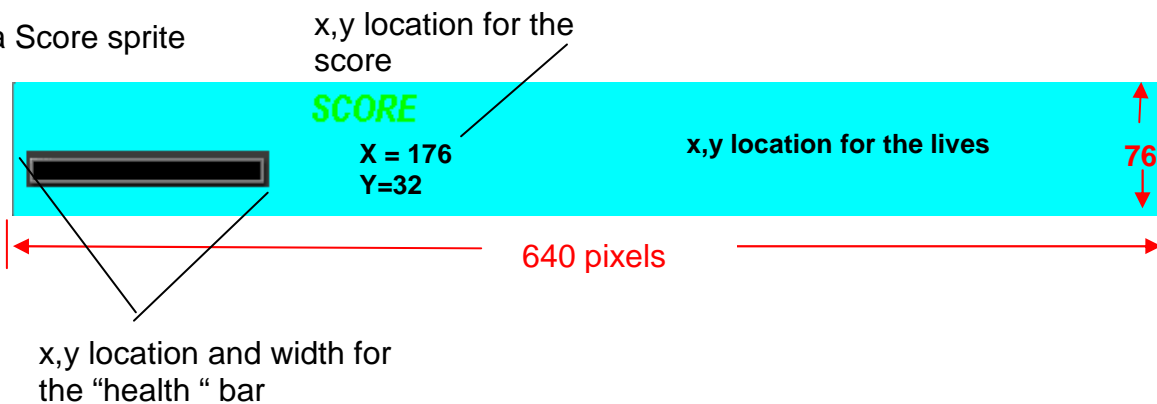
Create a score sprite and name it spri_score , it must be set as transparent and 640 pixels wide and around 76 pixels high

To change the size of the sprite from the default size - do this in the Create Sprite→Sprite Editor→Transform →resize canvas.



If you don't want to make your own score sprite use the one provided in Resources →sprites→score_ribbon

e.g of a Score sprite



Showing Health, Lives , Amo etc. Using Game Maker

Main Steps

1. Create a Score sprite
2. Create a controller object
3. Make the room bigger to accommodate the score bar at the bottom (or top)
4. Create a score object and give it the score sprite
5. Temporarily place the score sprite in the room and determine the x – y positions for the score sprite, scores, health bar and ammunition etc.
6. Create the Events and Actions for the controller that will initiate the scores and draw them on top of the score object that you placed in the room.
7. Remove the score object from the room and place the controller object in the room, it is represented by a blue dot.
8. For other objects such as the missile and targets: create Events and actions that add scores, life etc These items will then be shown on the top of the score sprite. Also add actions for re-starting the room and going to another room for the various objects and actions.

Here's a more detailed explanation of how to do this.

Before you start the scoring configuration your game will need a minimum of :

- 1 player – create sprite and object for player
- 1 enemy - create sprite and object for enemy
- 1 missile that the player throws at the enemy - create sprite and object for the missile
- 1 bullet or missile that the enemy throws at the player - create sprite and object for the enemy missile

At least 2 rooms

Configure Events and actions for :

Collision Event - players missile with enemy

Collision Event - enemies missile with player

Collision Event - players missile with enemy missile

Collision Event - enemies s missile with players missile

Keyboard Events and actions for moving the player

Keypress Event and actions for firing the players moving missile

Create Event for creating the enemies moving missile