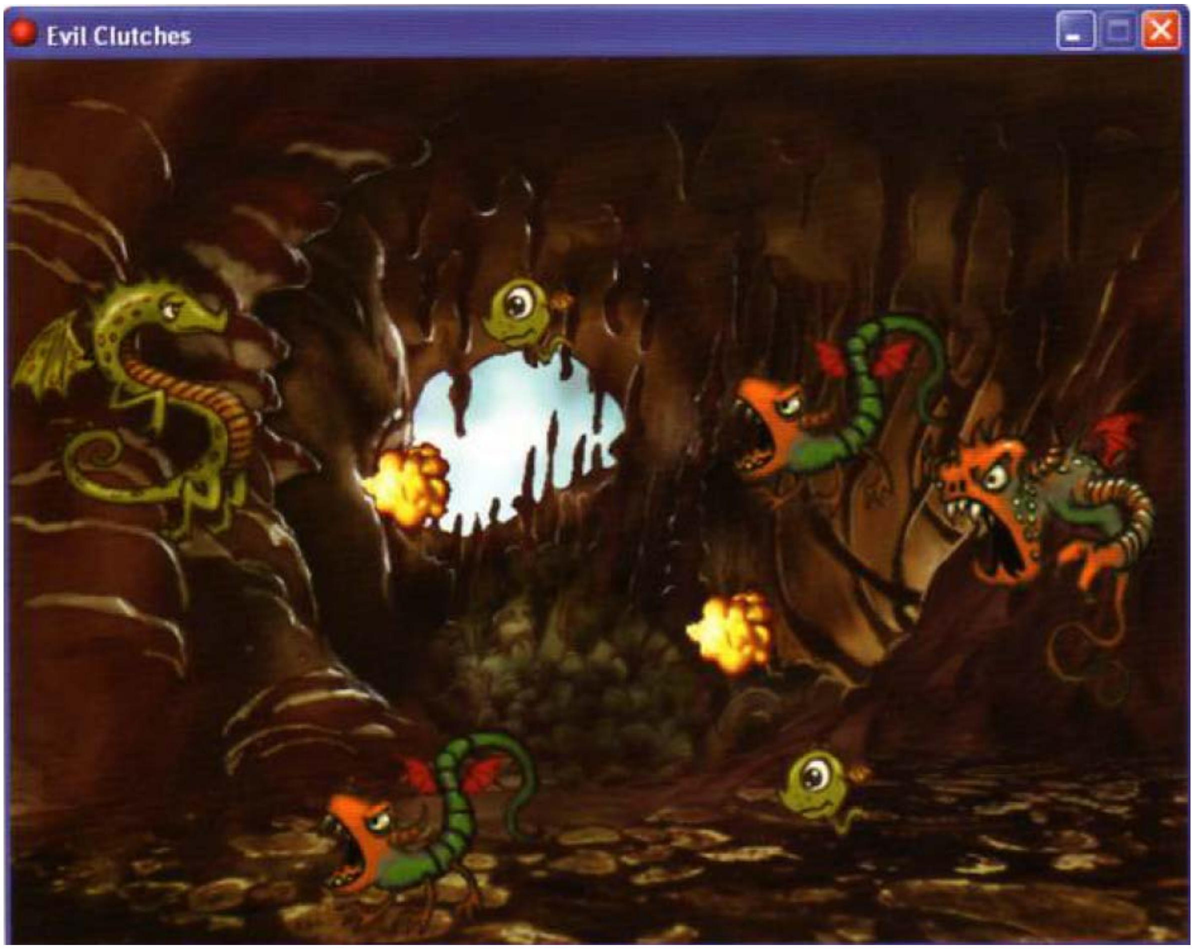
Learning something new is always a little daunting at first, but things will start to become familiar in no time. In fact, by the time we finish our second game today you'll have a much better understanding of what you can do with the Game Maker Program.

So let start designing our second Game: Evil Clutches

Normally we would start by writing a long design document. However, that is no fun, so let us just call our game Evil Clutches and I will give you a short description of our game.

You are going to play a mother dragon who must recue her hatchling from an unpleasant band of demons that have kidnapped them. The band's boss sends a stream of demons to destroy the dragon as the hatchling make their escape. The mother can fend off the boss's minions by shooting fireballs, but be careful not to accidentally shoot the hatchlings!

The arrow keys will move the dragon up and down and the spacebar will shoot fireballs. The player will gain points for shooting demons and rescuing young dragons, but will lose points for any hatchlings that accidentally get shot. The game is over if the dragon is hit by a demon, and a high score table will be displayed. **Below is a screen shot of what the game will look like.**

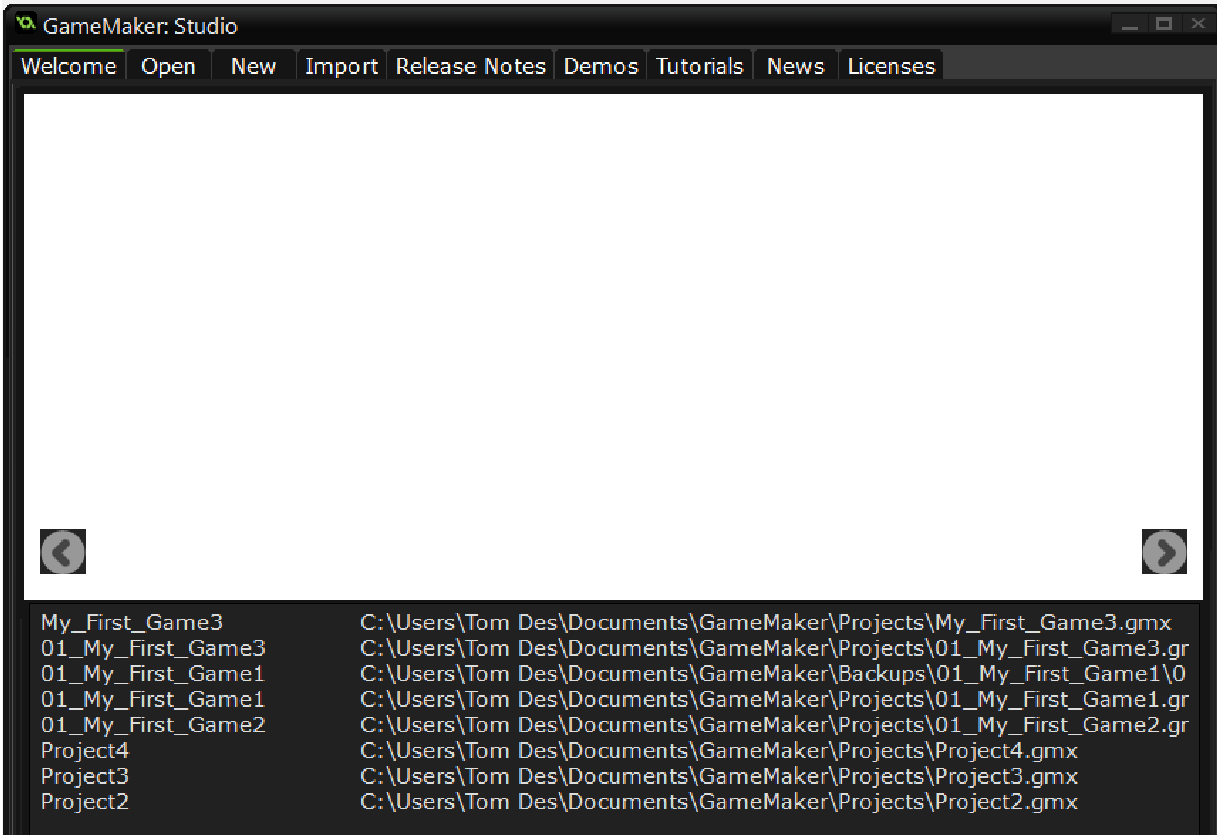


OK, using this description we can list all the different elements needed to create our game: a dragon, a boss, hatchlings, and fireballs. Making the game requires pictures of each of these as well as a background image, some sound effects, and music. We call of the different parts that make up the game resources, and the resources for this game, have already been created for you, and are located in the Game 2 resources folder.

Sprites

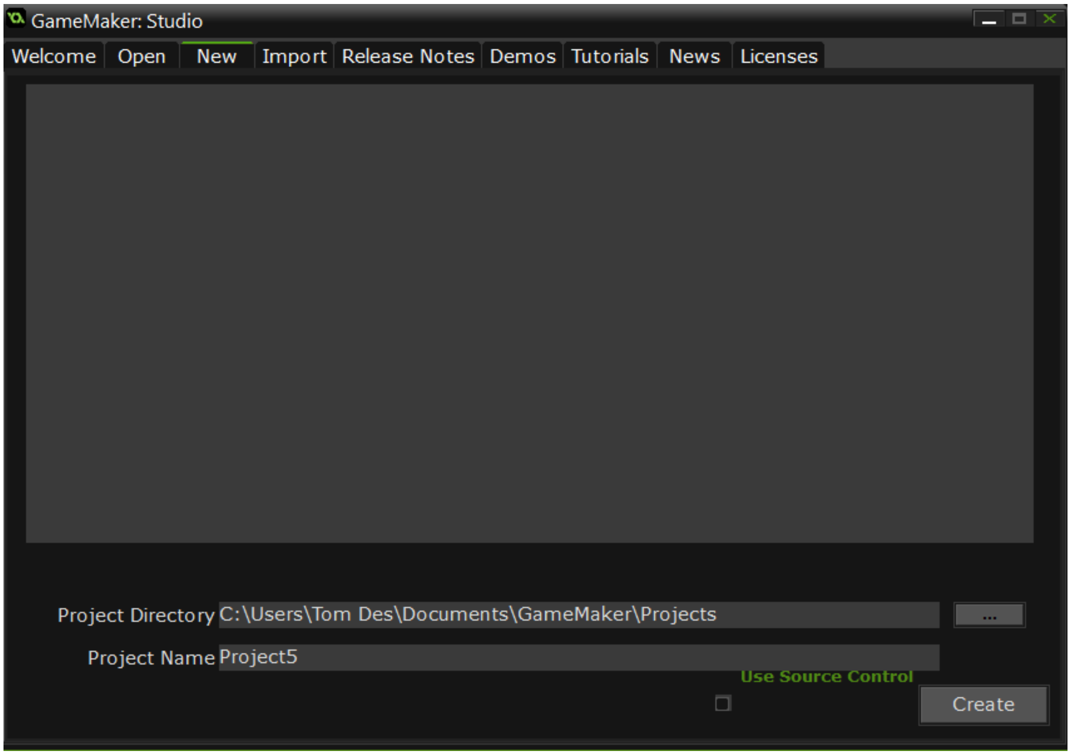
In Game Maker, pictures of dragons, demons, and other game objects are all called SPRITES. Sprites are one kind of resource used in games, and they can be made from images that have been created in programs like Photoshop and a different program, or downloaded from the Internet. Game Maker does however, include a simple sprite editor for drawing your own sprites, but you can use any drawing program you like to create your own sprites. However, creating sprites is or can be a time consuming process, so I have already provided some professionally drawn sprites for the game. They are in the Resource/Game2 folder.

Let us go ahead and start Game Maker now. You should see the screen below or something like it.



```
My_First_Game3       C:\Users\Tom Des\Documents\GameMaker\Projects\My_First_Game3.gmx
01_My_First_Game3    C:\Users\Tom Des\Documents\GameMaker\Projects\01_My_First_Game3.gr
01_My_First_Game1    C:\Users\Tom Des\Documents\GameMaker\Backups\01_My_First_Game1\0
01_My_First_Game1    C:\Users\Tom Des\Documents\GameMaker\Projects\01_My_First_Game1.gr
01_My_First_Game2    C:\Users\Tom Des\Documents\GameMaker\Projects\01_My_First_Game2.gr
Project4             C:\Users\Tom Des\Documents\GameMaker\Projects\Project4.gmx
Project3             C:\Users\Tom Des\Documents\GameMaker\Projects\Project3.gmx
Project2             C:\Users\Tom Des\Documents\GameMaker\Projects\Project2.gmx
```

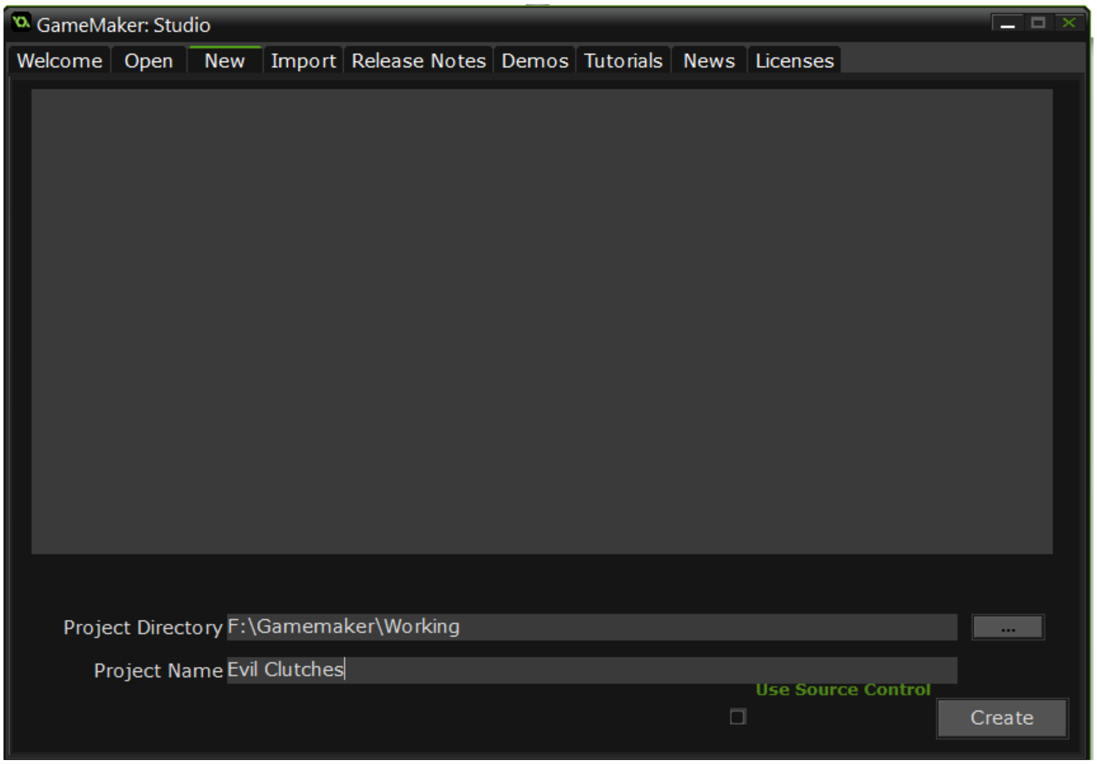
It is somewhat empty. So now, let us click on New in the menu.



Project Directory C:\Users\Tom Des\Documents\GameMaker\Projects

Project Name Project5

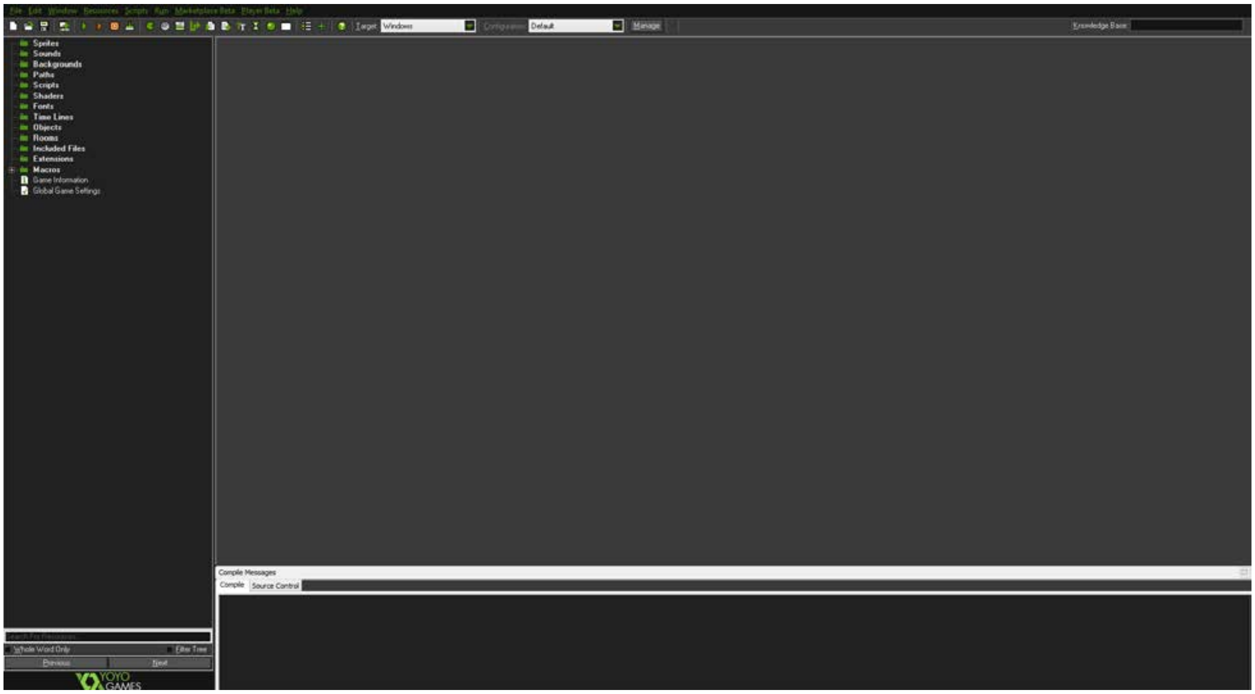
Now make sure you are saving your Game in the collect location and with the correct name. We will type Evil Clutches, and then click on the Create Button.



Project Directory F:\Gamemaker\Working

Project Name Evil Clutches

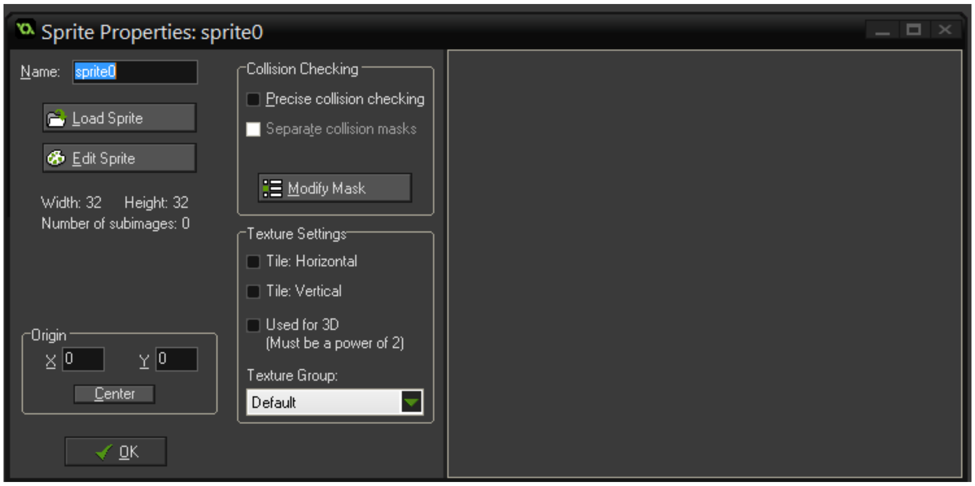
Once you click on the Create Button, you will see this screen. In the left window it shows the different types of resources that make up the game: sprites, sounds, backgrounds, paths, scripts, Shaders, Fonts, Time Lines, Objects, Rooms, Included Files, Extensions, and Macros.



All of them are currently empty, but the names of the new resources will appear here as they are added to the game. The Menu Bar along the top contains all the commands that allow us to work with resources. However, most common tasks can also be accessed using the buttons on the toolbar.

So let us get started by creating a sprite.

1) So click on Resources in the menu, then click on create Sprite. The Sprite Properties Window appears.


Sprite Properties: sprite0

Name: sprite0

Width: 32 Height: 32
Number of subimages: 0

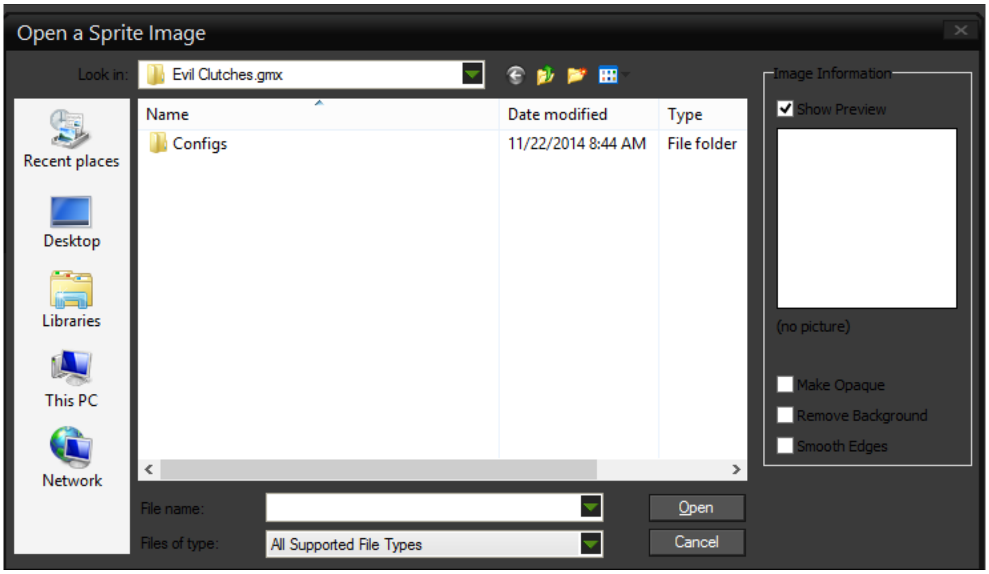
Collision Checking
□ Precise collision checking
□ Separate collision masks

Texture Settings
□ Tile: Horizontal
□ Tile: Vertical
□ Used for 3D (Must be a power of 2)
Texture Group: Default

Origin
X 0 Y 0

2) Click in the name filed where it currently says sprite0 and type sprite_dragon.

3) Click on Load Sprite Button. This opens the standard Window file requester.



Open a Sprite Image

Look in: Evil Clutches.gmx

Name	Date modified	Type
Configs	11/22/2014 8:44 AM	File folder

Image Information

☑ Show Preview

(no picture)

□ Make Opaque

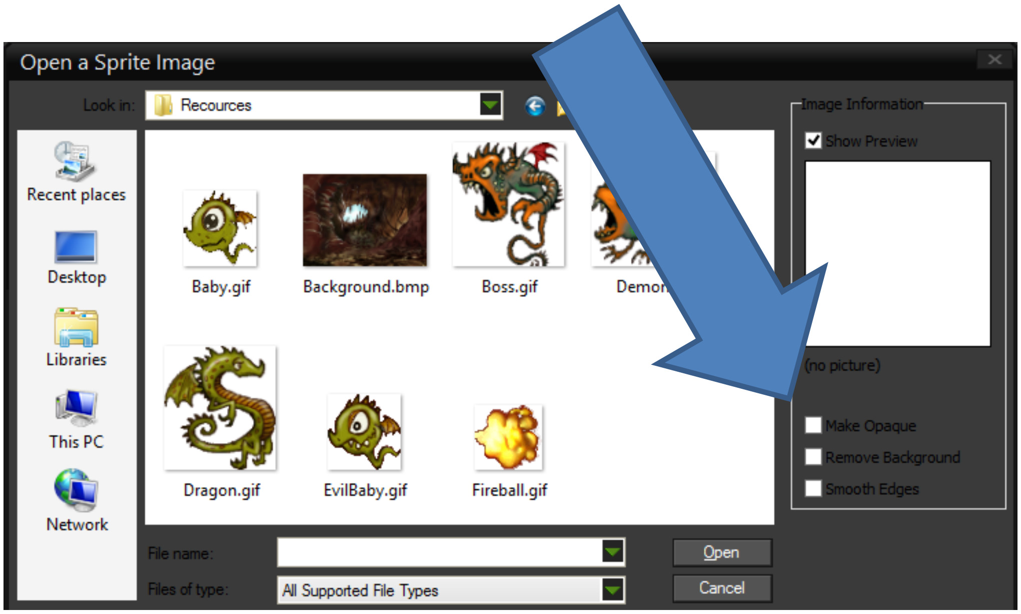
□ Remove Background

□ Smooth Edges

File name:

Files of type: All Supported File Types

4) You will go to the location of your resources and click on the Dragon.gif -- **make sure to** click the box that says remove background and smooth edges.
Then click Open



Open a Sprite Image

Look in: Recources

Image Information

☑ Show Preview

(no picture)

□ Make Opaque

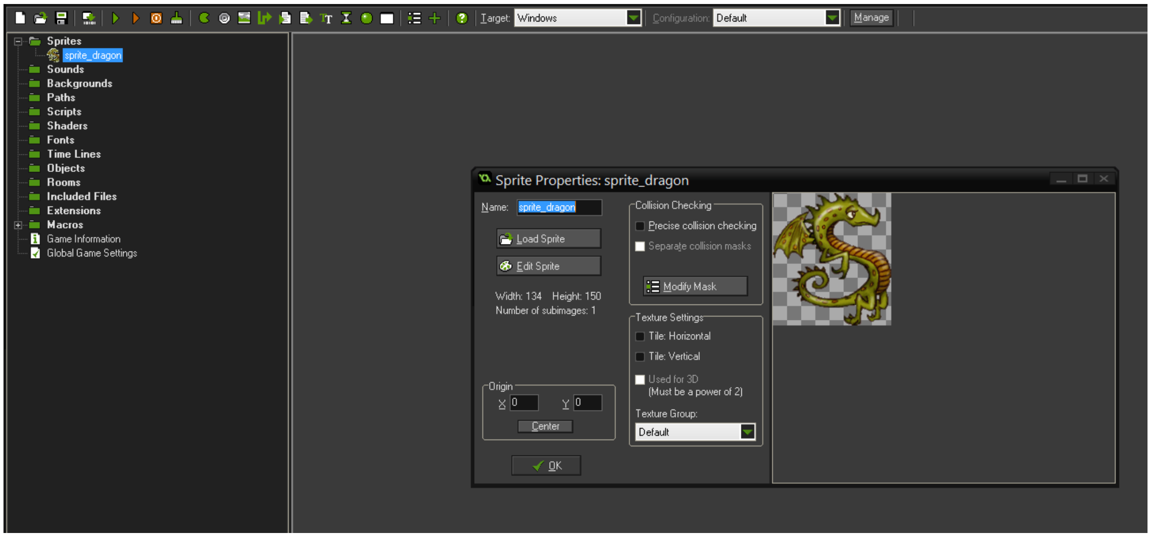
□ Remove Background

□ Smooth Edges

File name:

Files of type: All Supported File Types

5) You will now see the Dragon sprite properties window and in the left window you will see the sprint_dragon has been added. Click the **OK button** to close the properties window.



Name: sprite_dragon

Width: 134 Height: 150
Number of subimages: 1

□ Precise collision checking
□ Separate collision masks

□ Tile: Horizontal
□ Tile: Vertical
□ Used for 3D (Must be a power of 2)
Texture Group: Default

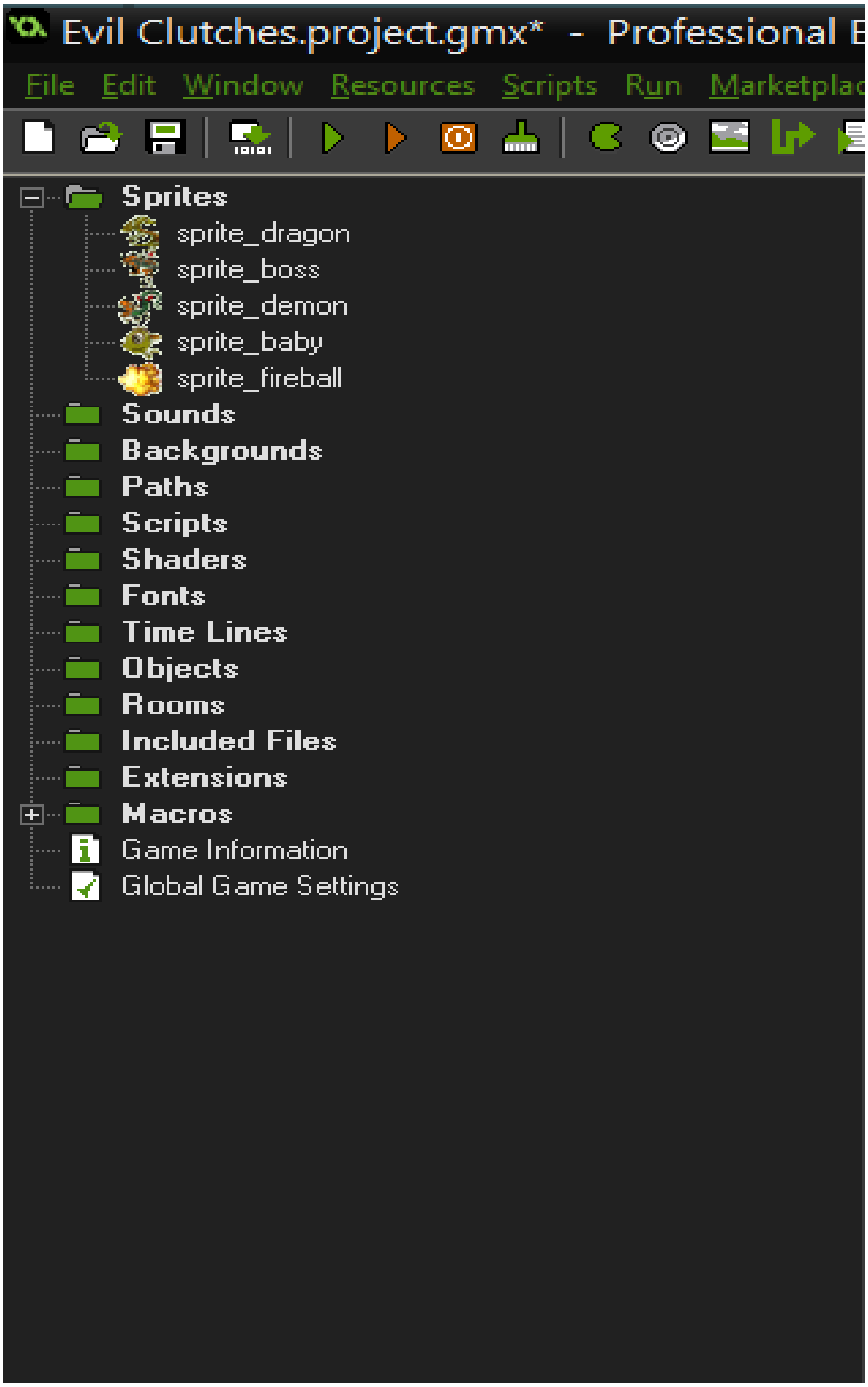
X 0 Y 0

6) Ok now let us create the remaining sprites for our game.
   a) From the Resources menu, choose Create Sprite.

      1) In the Name Field type sprite_boss
      2) Click the Load Sprite button and choose the file Boss.gif, make sure the Remove background and Smooth Edges boxes are still checked.
      3) Click OK to close the Sprite Properties Window.

7) Now repeat the process, and create the following sprites: Demon sprite, baby sprite, and fireball sprite, using the Demon.gif, Baby.gif, and fireball.gif files in the same way as you created the Dragon and boss sprites.

8) You have created all the sprites needed for your Evil Clutches Game.

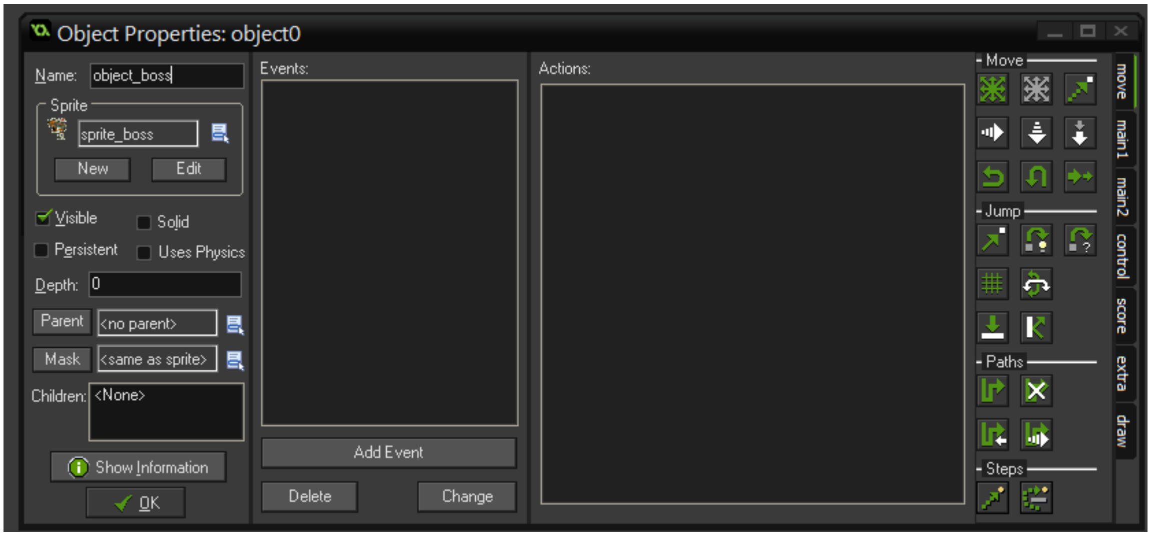


Objects

Sprites do not do anything on their own; they just store pictures of the different elements in the game. Objects: are the parts of the game that control how these elements move around and react to each other. We will begin by creating our first object to tell Game Maker how we want the demon boss to behave.

The Boss Object

1) From the Resources menu, choose Create Object. An Object Properties Window opens:



Object Properties: object0

Name: object_boss

Sprite: sprite_boss

☑ Visible □ Solid

□ Persistent □ Uses Physics

Depth: 0

Parent <no parent>

Mask <same as sprite>

Children: <None>

2) In the name field, give the object a name. Type object_boss.

3) Click the icon at the end of the sprite field and a list of all the available sprite will appear. Select the sprite_boss sprite, by click on it.

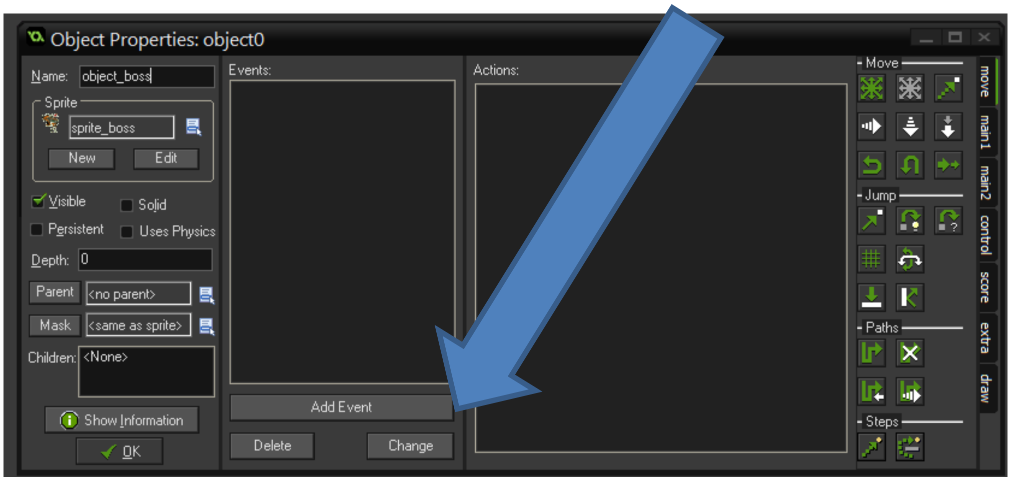
Events and Actions

Game Maker uses events and actions to specify how objects should behave. Events are important things that happen in the game, such as when objects collide or when the player presses a key on the keyboard. Actions are things that happen in response to an event, such as changing an objects direction, setting the score, or playing a sound. Game Maker games are just a collection of objects with actions to tell them how they should react to different events. Therefore, to set the behavior of an object in Game Maker you must define which events the object should react to and what actions they should perform in response.

The boss object's lists of actions are currently empty. So let us begin to add some evens and actions that will start the boss moving up the screen at the beginning of the game. This will be complemented, by an action that reverses the vertical direction of the boss in the event that it collides with the edge of the screen. As a result, the boss will continually move up and down between the top and bottom of the screen.

Adding a create event for the boss object:

1) With the object_boss window still open click the Add Event Button



Name: object_boss

Sprite: sprite_boss

☑ Visible □ Solid

□ Persistent □ Uses Physics

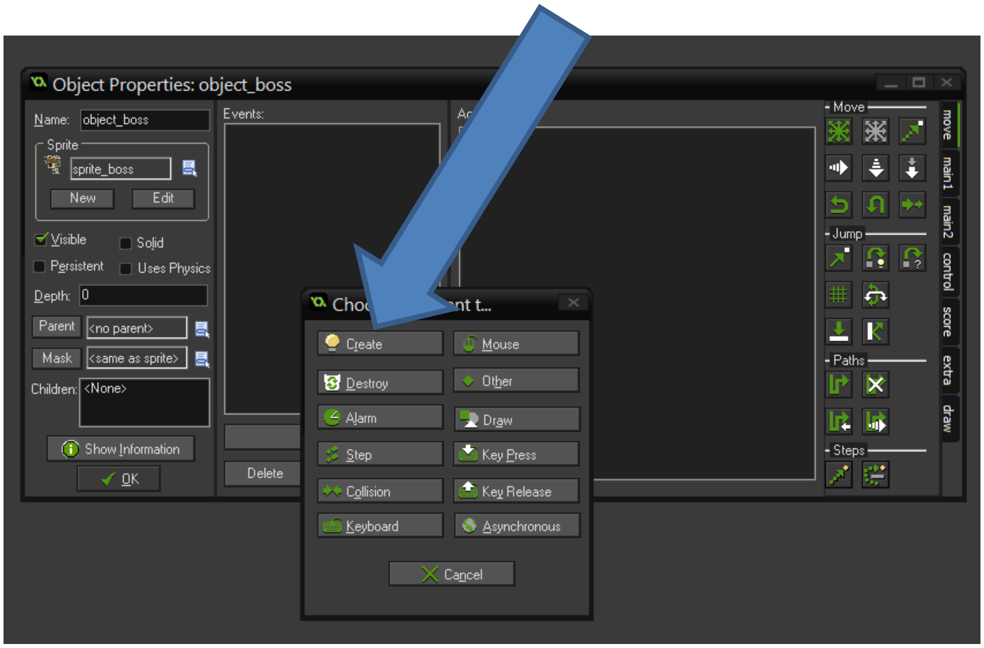
Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

2) Now click on Create to add it to the list of events.



Name: object_boss

Sprite: sprite_boss

☑ Visible □ Solid

□ Persistent □ Uses Physics

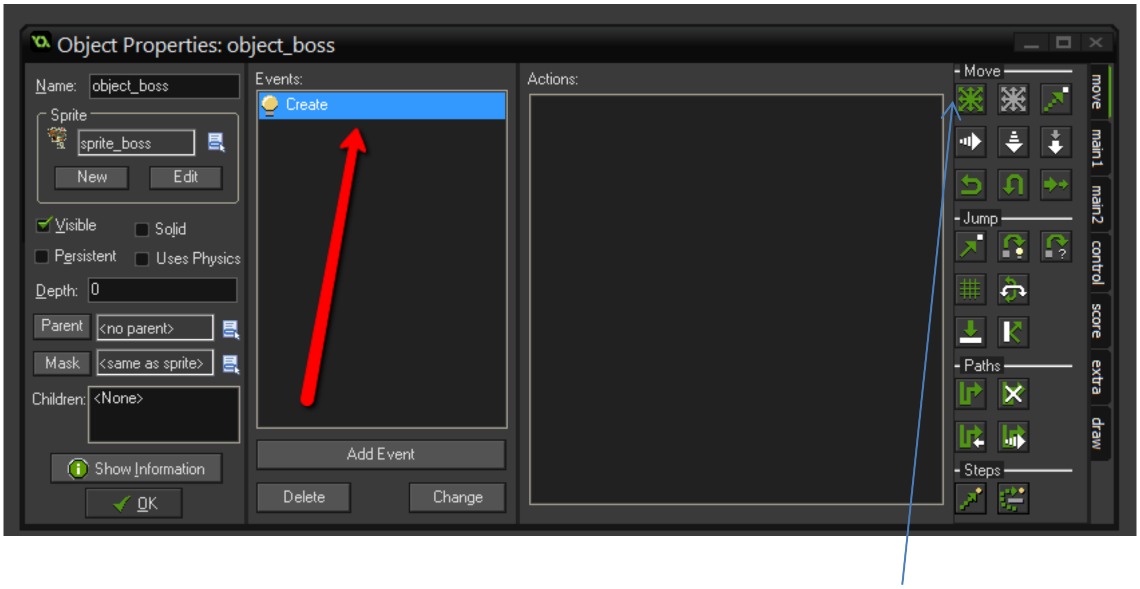
Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

3) A new event is automatically selected with (with a blue highlight) in the events list.



Object Properties: object_boss

Name: object_boss

Sprite: sprite_boss

☑ Visible □ Solid

□ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

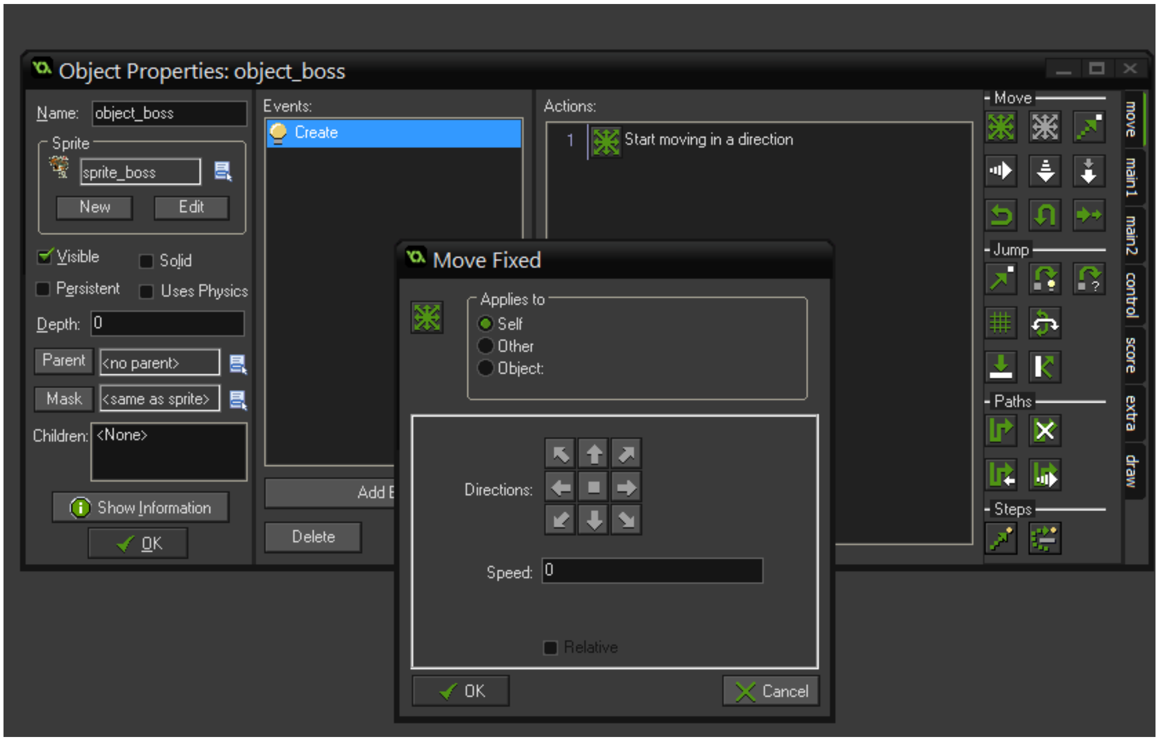
Mask: <same as sprite>

Children: <None>

Events: Create

Actions:

4) Now we need to include the Move Fixed action in the list of actions. To do this, press and hold the left mouse button on the action image named Fixed Move, listed in the Move tab. Drag it to the Actions area and release the mouse. An action form will popup asking you for particular information about the action.



Object Properties: object_boss

Name: object_boss

Sprite: sprite_boss

☑ Visible □ Solid

□ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events: Create

Actions:

1 Start moving in a direction

Move Fixed

Applies to

● Self

○ Other

○ Object:

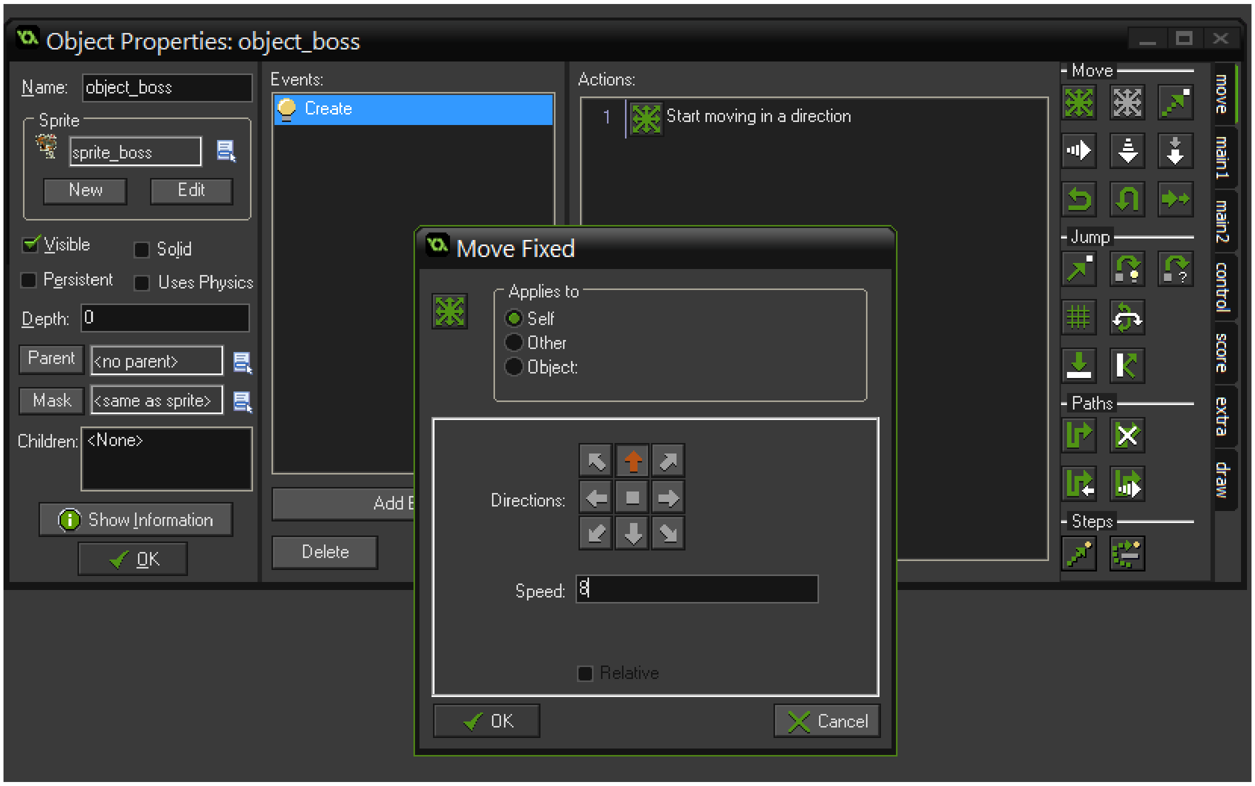
Directions:

Speed: 0

□ Relative

5) Make sure the **Self** Option, is selected in the Applies to section.

6) Now select the Up Arrow and enter a value of 8 for speed. This will make the object move vertically 9 pixels (the tiny squares that make up a monitor display) for every step that it takes.



Object Properties: object_boss

Name: object_boss

Sprite: sprite_boss

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent <no parent>

Mask <same as sprite>

Children: <None>

Events: Create

Actions:

1 Start moving in a direction

Move Fixed

Applies to

● Self

○ Other

○ Object:

Directions:

Speed: 8

☐ Relative

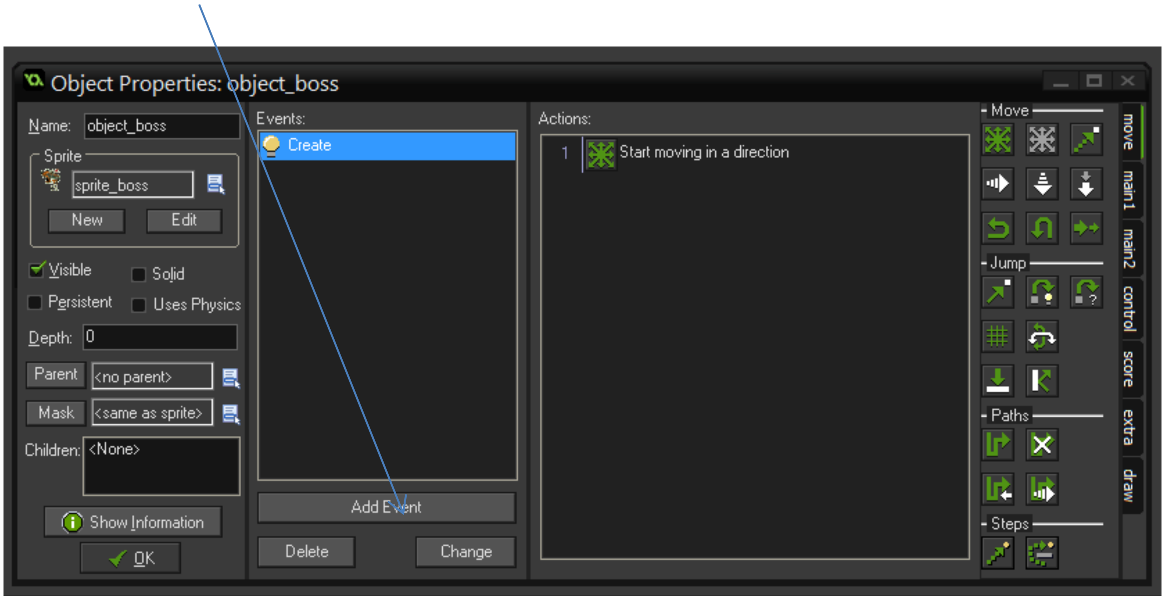
7) Press the OK Button

NOTE: -- All of Game Maker's Actions are organized into tabbed pages of icons on the right of the Actions list. Browse through the different actions tabs and hold your mouse over each one to reveal its name.

8) This event should start the boss moving upward. Now we will add an event to reverse an object's vertical direction when it collides with the edge of the screen. The event is called the Intersect Boundary event because it gets called when the object's sprite intersects the screen's boundary by being partly in and partly out of the screen.

Adding an Intersect boundary event for the boss object:

1) Click the Add Event button:



Object Properties: object_boss

Name: object_boss

Sprite: sprite_boss

☑ Visible □ Solid

□ Persistent □ Uses Physics

Depth: 0

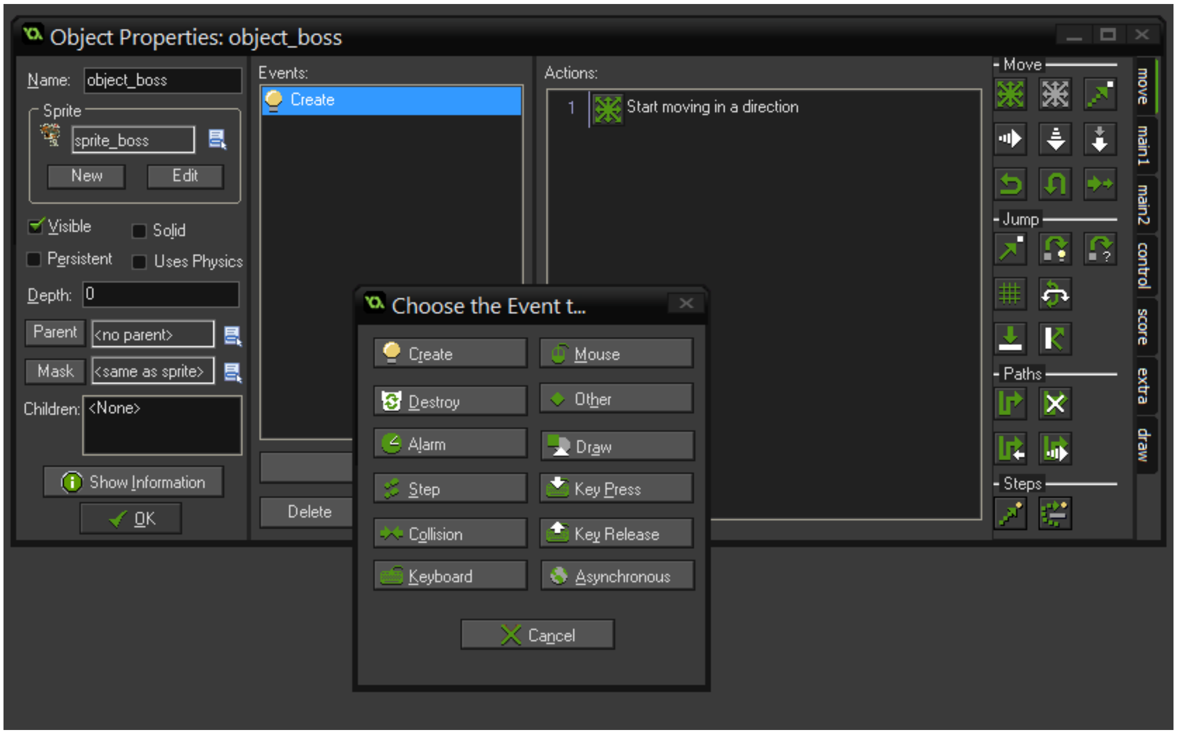
Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events: Create

Actions: 1 Start moving in a direction



Object Properties: object_boss

Name: object_boss

Sprite: sprite_boss

☑ Visible □ Solid

□ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

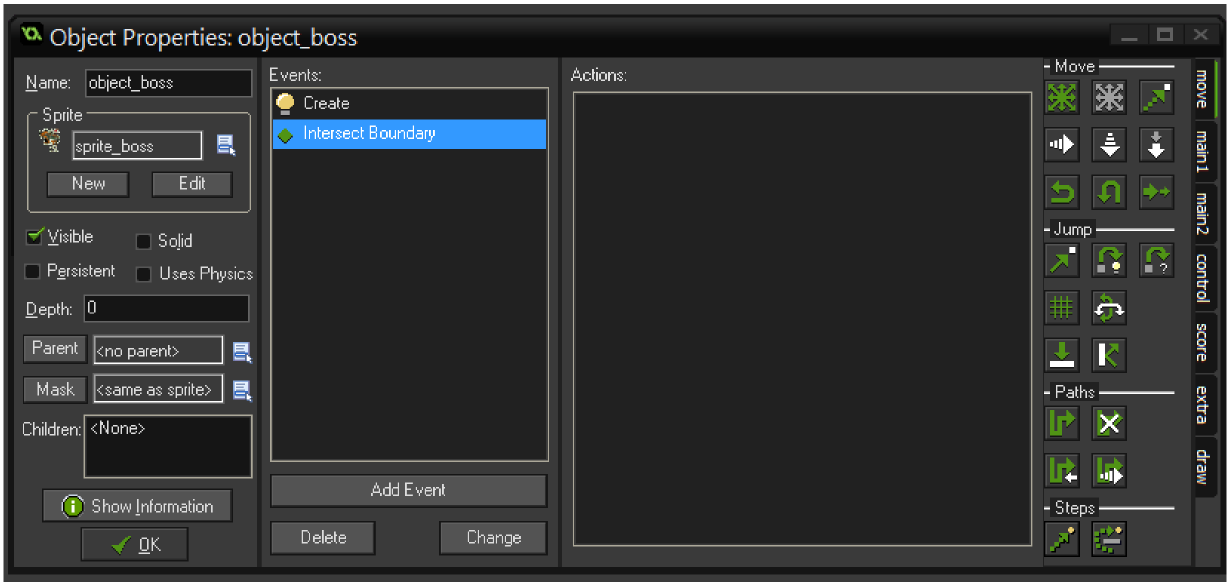
Mask: <same as sprite>

Children: <None>

Events: Create

Actions: 1 Start moving in a direction

2) Now Select **Other** from the Event Selector pop-up window and select **Intersect boundary** from the drop down menu that appears. This action will then be added and selected in the list of events.



Object Properties: object_boss

Name: object_boss

Sprite: sprite_boss

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

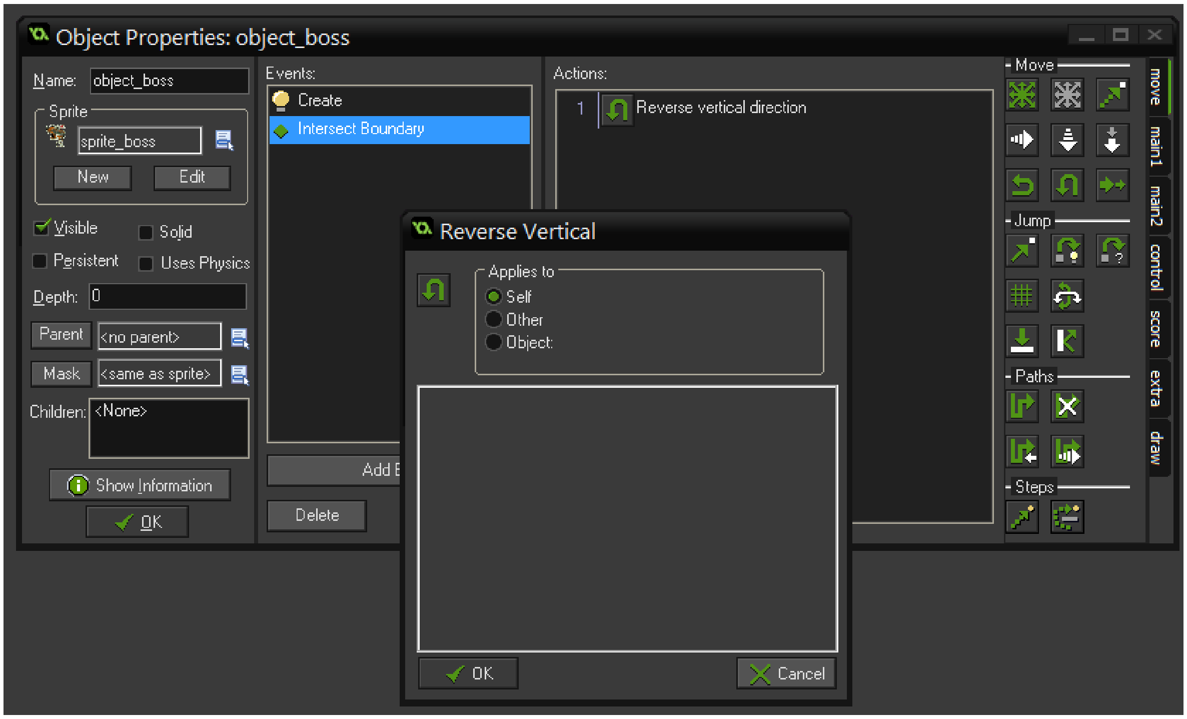
Mask: <same as sprite>

Children: <None>

Events:
- Create
- Intersect Boundary

Actions:

3) Ok we now need to add an action for thee Intersect boundary Event. A **Reverse Vertical direction** action. We click and drag it from the Move Action Tab Area to the Actions: and you will see the following pop-up box, except the default actions and click the OK Button.



Object Properties: object_boss

Name: object_boss

Sprite: sprite_boss

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:
- Create
- Intersect Boundary

Actions:

1 Reverse vertical direction

Reverse Vertical

Applies to

● Self

○ Other

○ Object:

4) We have added two events along with their corresponding actions. These are all of the events and actions we need for the boss right now. We can switch between the different events by clicking on them in the **Events** list. The selected event is highlighted in blue and the actions for that even are then shown in the **Actions** list. You can edit the properties of each action by double-clicking on them, but we are done with the boss object for now.

The Dragon Object

Let state by creating an object for the dragon in the same way as for the boss.

1) From the **Resources** menu, choose **Create Object**.

2) Give the object a name by entering object_dragon in the **Name** field.

3) Select the sprite_dragon from the drop-down sprite menu.

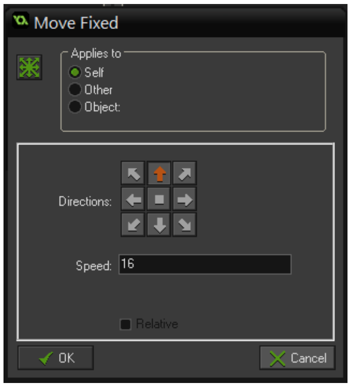
The Dragon also needs actions to make it move up and down the screen, but this time only when the appropriate keys are pressed on the keyboard. We do this by using the **keyboard** events.

Adding keyboard events for the dragon object:

1) Click the **Add Event** button

2) Choose a Key Press event and select the <**UP**> action in the Actions list.

3) In the action for, select the **upward** direction arrow and set the **Speed** to 16 and Press the OK Button.



Move Fixed

Applies to

● Self

○ Other

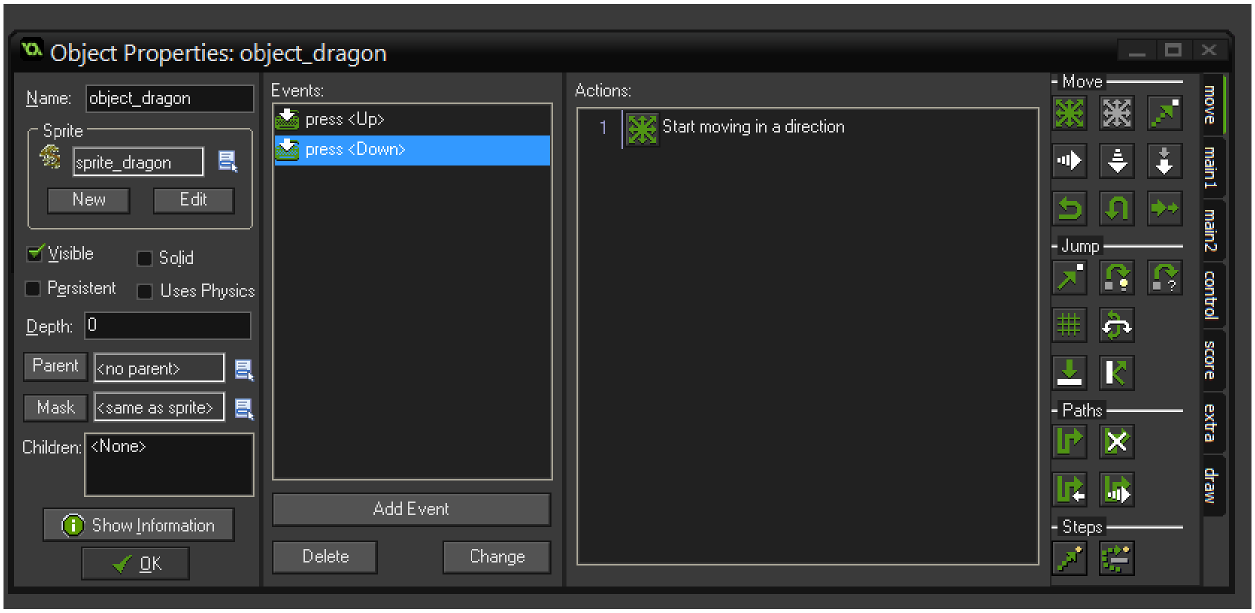
○ Object:

Directions:

Speed: 16

□ Relative

4) Repeat the previous process to add **a KeyPress** event for the <**Down**> key that includes a **Move Fixed** action with a **downward** direction arrow and a speed of **16**. The Object Properties Window should now look like below.



Object Properties: object_dragon

Name: object_dragon

Sprite: sprite_dragon

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

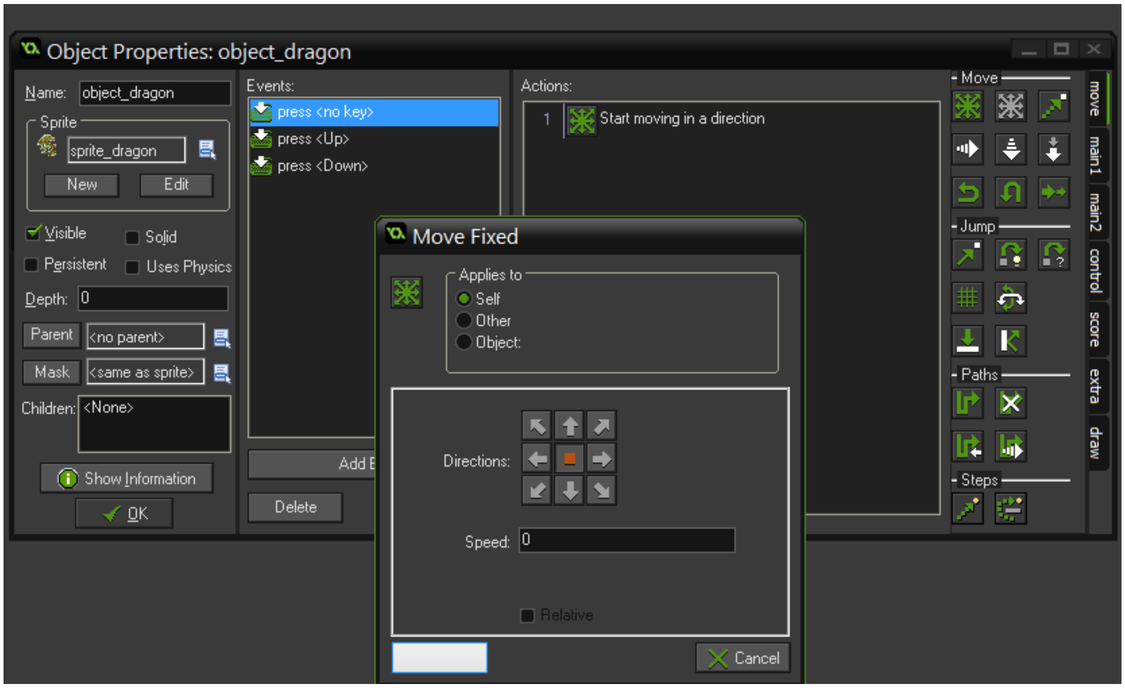
Events:
- press <Up>
- press <Down>

Actions:

1 Start moving in a direction

5) We only need one more event and action to make the dragon's movements work correctly. Our Key Press events will start the dragon moving when the player presses the arrow keys, but there are currently no events to stop it from moving again when the keys are no longer being pressed. We use the **KeyPress** <**no key**> event to test for when the player is no longer pressing any keys.

6) Ok, so now we will add the <**no key**> event.
7) Click the Add Event button
8) Choose Key Press event and select <no key> from the pop-up menu.
9) Add the Move Fixed action
10) Select the center square in the action window and set seed to 0
11) Now click the OK Button



Object Properties: object_dragon

Name: object_dragon

Sprite: sprite_dragon

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:
- press <no key>
- press <Up>
- press <Down>

Actions:
1 Start moving in a direction

Move Fixed

Applies to
● Self
○ Other
○ Object:

Directions:

Speed: 0

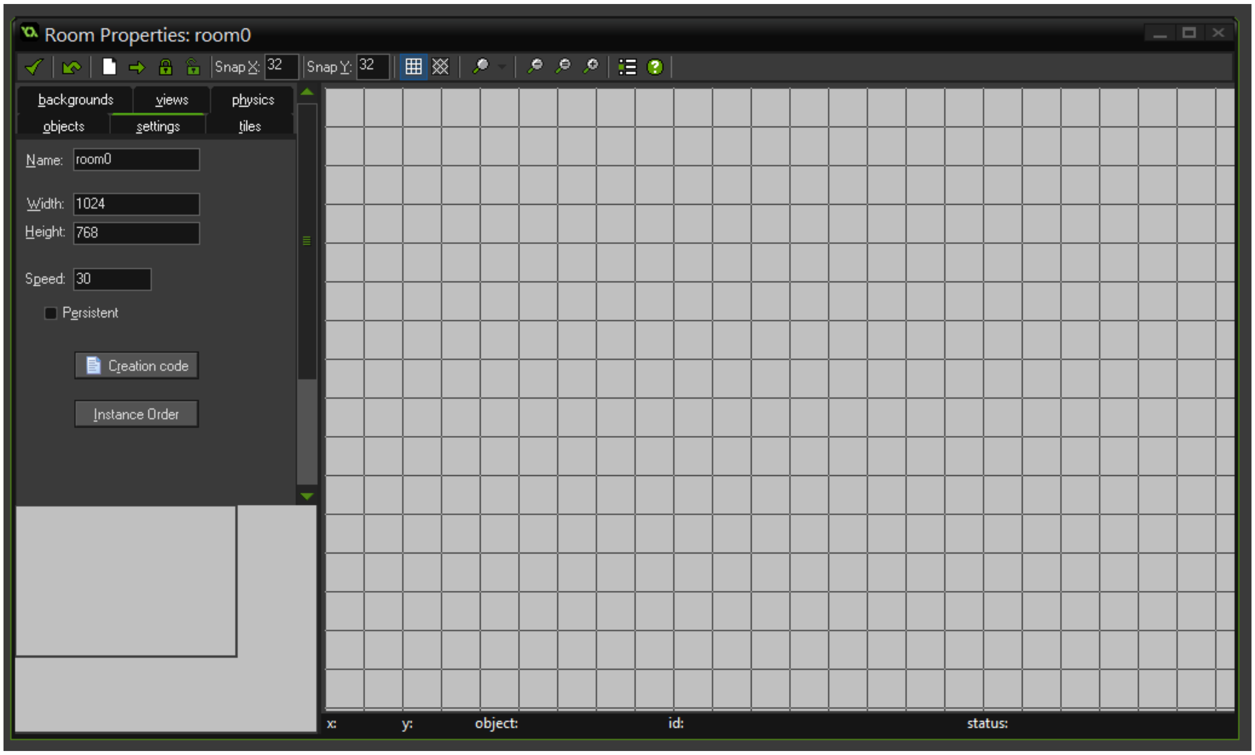
☐ Relative

Rooms

Our dragon and boss object are all ready to go now, but in order to see them we need to put them into a level. Levels in Game maker are made using **rooms**, and putting objects into a room defines where they will appear at the start of the game. However, not all objects need to be there at the start of the game, and they can be created on the fly as well (fireball, for example). So let us create a new room.

Creating a new room resource:

1) Select **Create Room** from the **Resources** menu. A Room Properties window will appear:



Room Properties: room0

Snap X: 32

Snap Y: 32

Name: room0

Width: 1024

Height: 768

Speed: 30

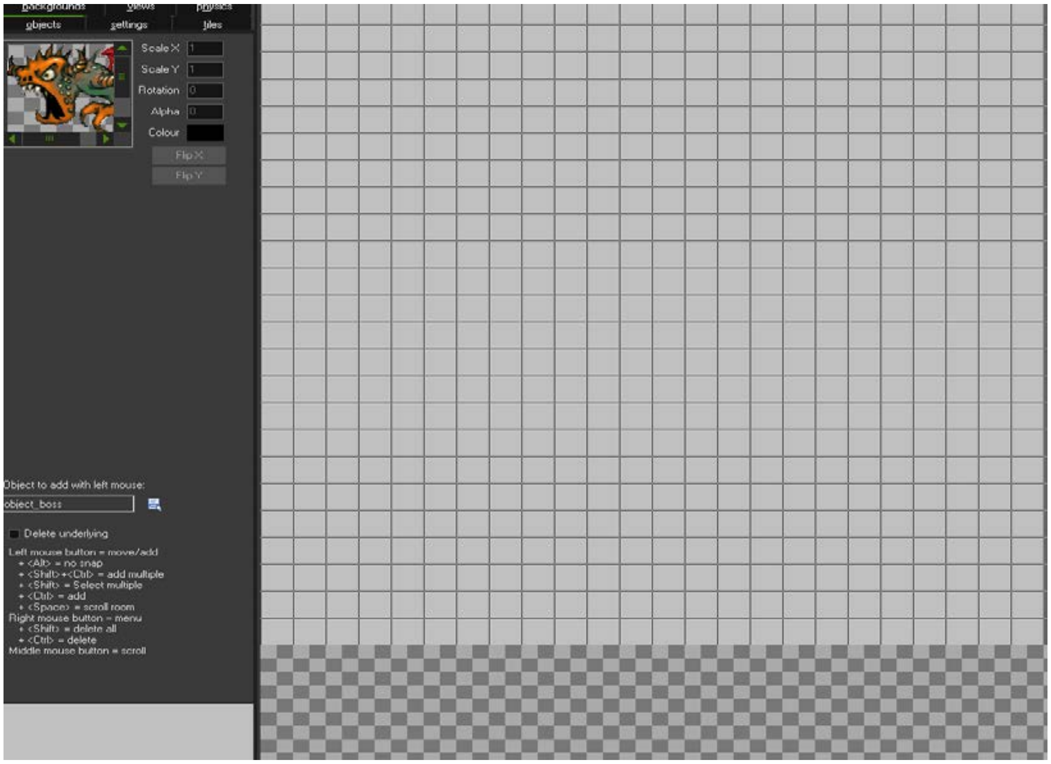
□ Persistent

NOTE: If there are sliders along the edges of the room grid, then the window is not currently large enough to see the entire room. Maximize the Game Maker window and the Room Properties window to see more of the room, or use the sliders to scroll around the entire room.

2) If needed click on the setting tab, and enter a name for the room in the Name field. Call this room_main.
The room setting should be good now. Next, we ill place our objects in the new room.

Adding a dragon and boss to the room:

1) Click the objects tab in the menu. You should see the dragon object is already selected as the object "to add with the left mouse."



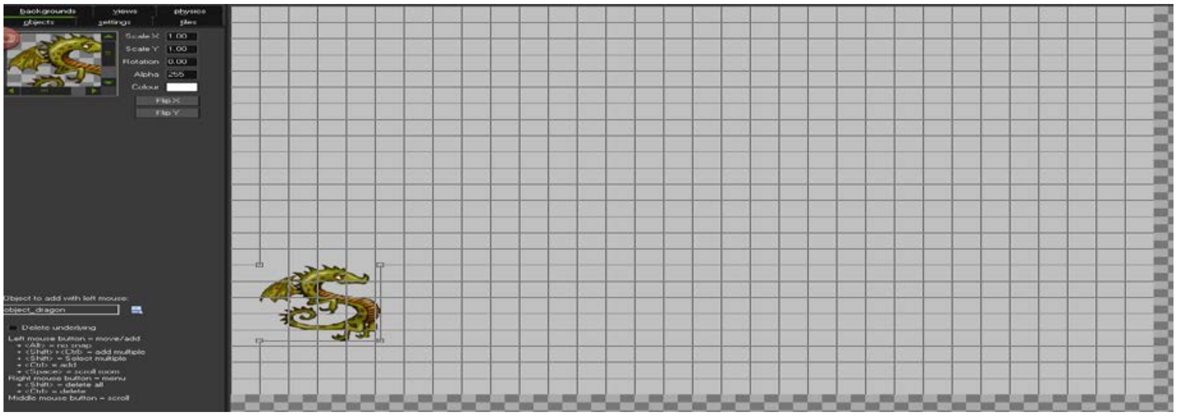
Scale X 1
Scale Y 1
Rotation 0
Alpha 0
Colour

Object to add with left mouse:
object_boss

Delete underlying

```
Left mouse button = move/add
   + <Alt> = no snap
   + <Shift>+<Ctrl> = add multiple
   + <Shift> = Select multiple
   + <Ctrl> = add
   + <Space> = scroll room
Right mouse button = menu
   + <Shift> = delete all
   + <Ctrl> = delete
Middle mouse button = scroll
```

2) Click on the room grid with the left mouse button. An instance of the dragon object will be placed with its top-left corner at the point at which you click. The position you place the dragon becomes its starting position in the game, so put just one dragon close to the left boundary of the room area. If you add it in the wrong place, use the right mouse button to remove it.



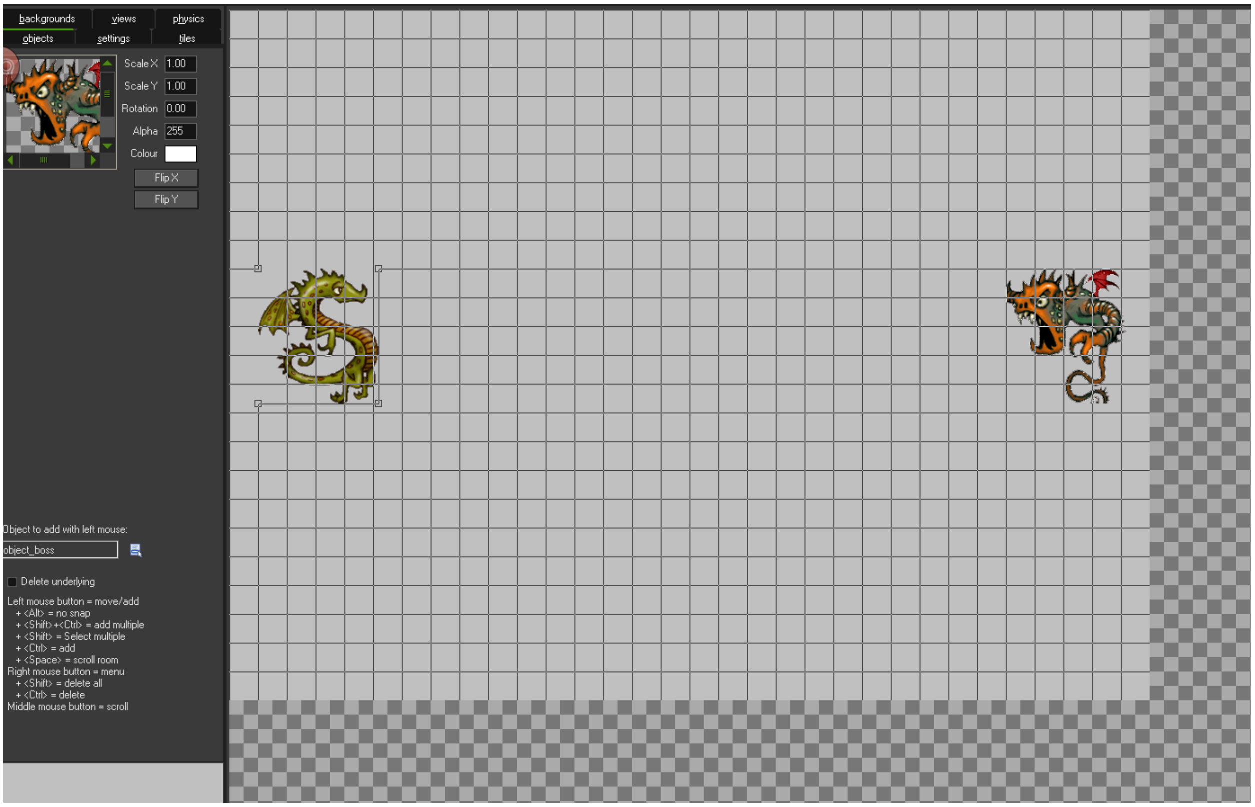
Scale X 1.00
Scale Y 1.00
Rotation 0.00
Alpha 255
Colour

Object to add with left mouse:
object_dragon

Delete underlying

```
Left mouse button = move/add
   + <Alt> = no snap
   + <Shift>+<Ctrl> = add multiple
   + <Shift> = Select multiple
   + <Ctrl> = add
   + <Space> = scroll room
Right mouse button = menu
   + <Shift> = delete all
   + <Ctrl> = delete
Middle mouse button = scroll
```

3) Click on the dragon's image on the objects tab (or on the image of the pop-up menu next to where it says object_dragon) and select the boss object from the menu that appears.
4) Place a instance of the boss close to the right edge of the room, but make sure that the whole of his sprite is completely inside the room – otherwise his events will not work correctly!! The room should now look like below:



Scale X 1.00
Scale Y 1.00
Rotation 0.00
Alpha 255
Colour

Object to add with left mouse:
object_boss

□ Delete underlying

Left mouse button = move/add
+ <Alt> = no snap
+ <Shift>+<Ctrl> = add multiple
+ <Shift> = Select multiple
+ <Ctrl> = add
+ <Space> = scroll room
Right mouse button = menu
+ <Shift> = delete all
+ <Ctrl> = delete
Middle mouse button = scroll

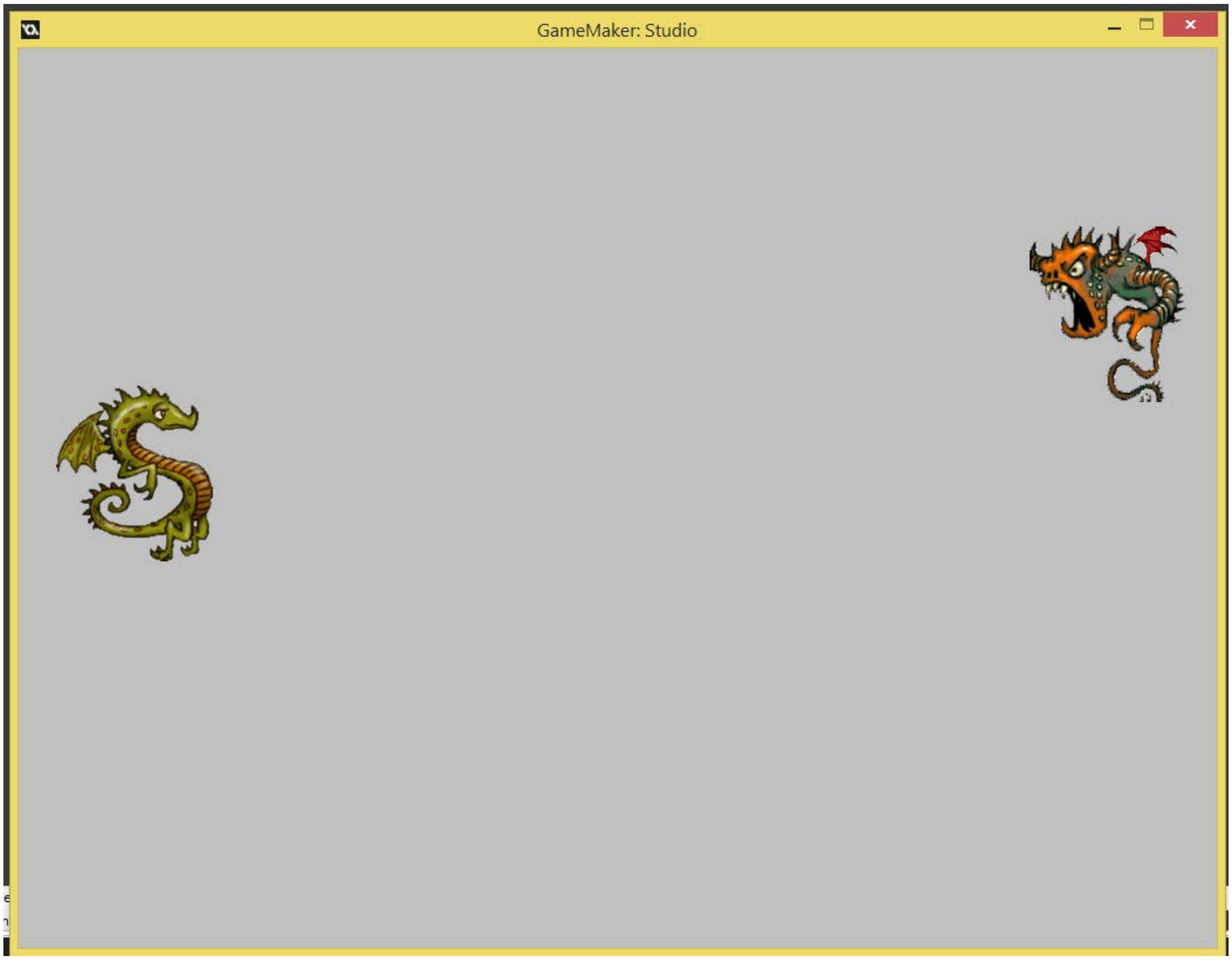
Our very first version of the game is now ready. Click the green checkmark in the top-left corner of the window to close it.

NOTE: You can also click and hold the mouse button to move instances within a room.

:::SAVE AND RUN:::

It is always a good idea to save our work as often as possible – just in case our computer crashes. So please go to **file** and click **Save**.

Now go to the menu and click on Run and Run Normally – Let's see how things have turned out so far for you. Use you arrow keys to move up and down.



You should be able to move the dragon up and down using the arrow keys, and the bass should float up and down by itself. If your game does not work in this way, then you might want to check through all the steps in the previous sections.

NOTE : Pressing the F-4 while the game is running will maximize the game to fill the entire screen. Press F-4 again to return to the previous window size.

So far, we have two object resources in our game and two characters appearing on the screen.

No to create some challenges and goals, we are going to need to bring our remaining objects into the game. So let us start by giving the dragon the ability to breathe fireballs – as dragons often do!

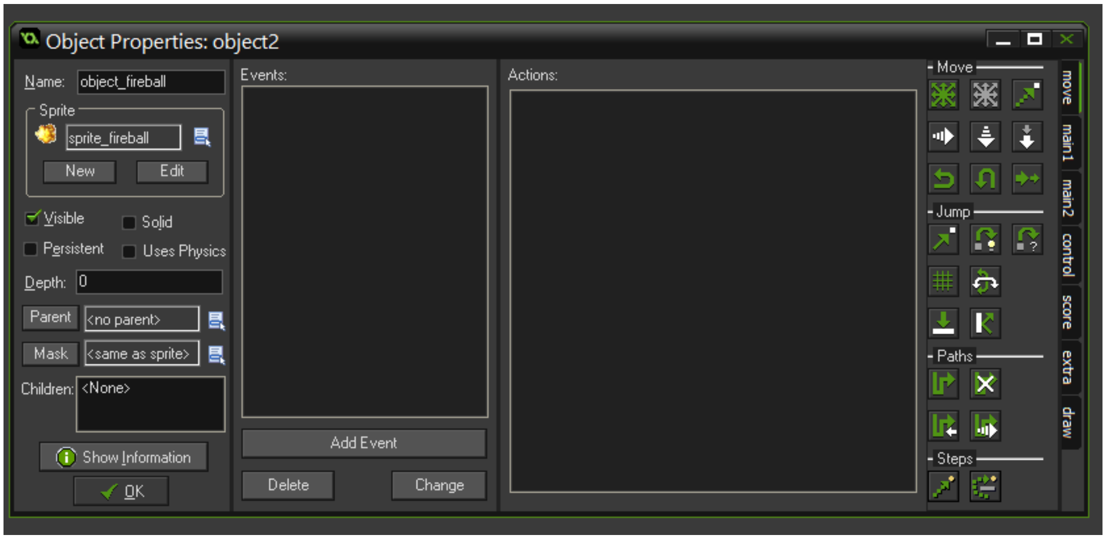
The Fireball Object:

To create the fireball object we will need the fireball sprite.

1) Select **Create Object** from the **Resources** menu.

2) Call the object **object_fireball**

3) Select the **fireball sprite**



Object Properties: object2

Name: object_fireball

Sprite: sprite_fireball

☑ Visible □ Solid

□ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

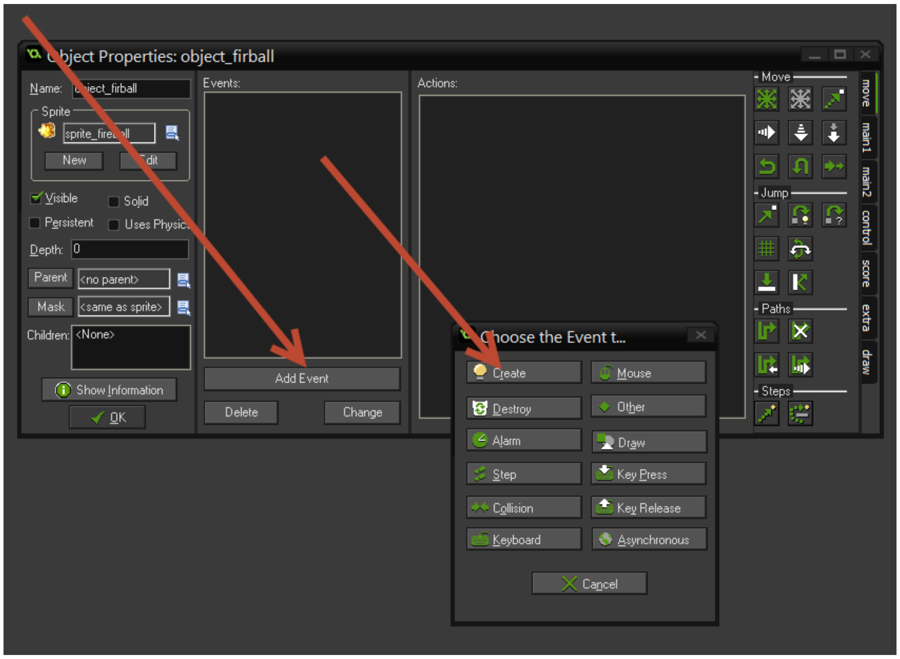
Mask: <same as sprite>

Children: <None>

We now need to think about how we want fireballs to behave. When the dragon creates a fireball, we want it to move across the screen toward the boss and be destroyed when it reaches the other side of the screen.

Adding the fireball object’s events:

1) Click the Add Event Button and Choose the Create event.


Object Properties: object_firball

Name: object_firball

Sprite: sprite_fireball

☑ Visible □ Solid

□ Persistent □ Uses Physics

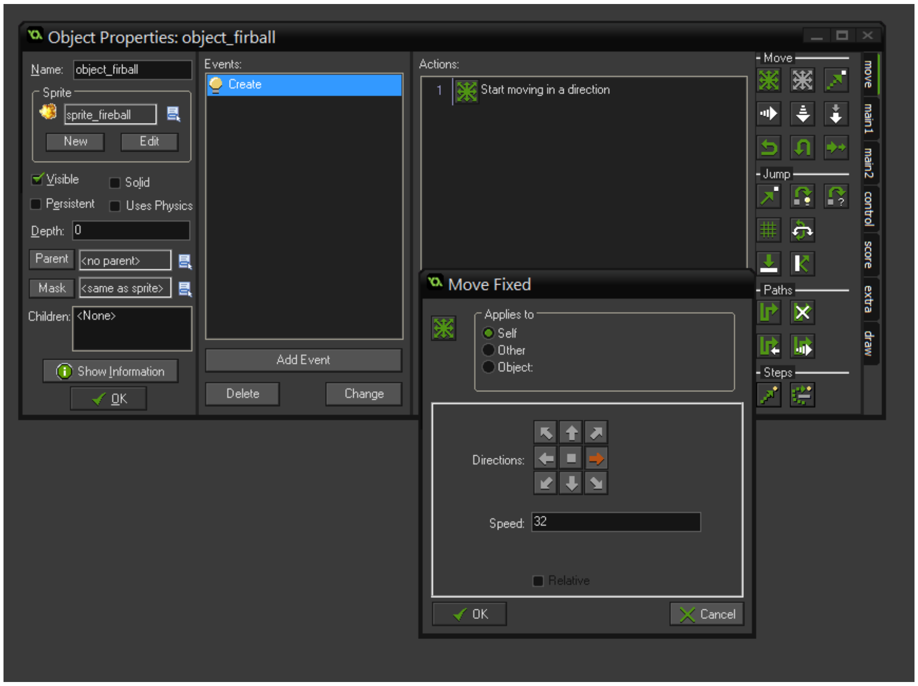
Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

2) Include the Move **Fixed** action in the **Actions** list. Select the right Arrow to indicate the direction and set Speed to 32 (Fireball fly Fast), click OK.


Object Properties: object_firball

Name: object_firball

Sprite: sprite_fireball

☑ Visible □ Solid

□ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events: Create

Actions: 1 Start moving in a direction

Move Fixed

Applies to

● Self

○ Other

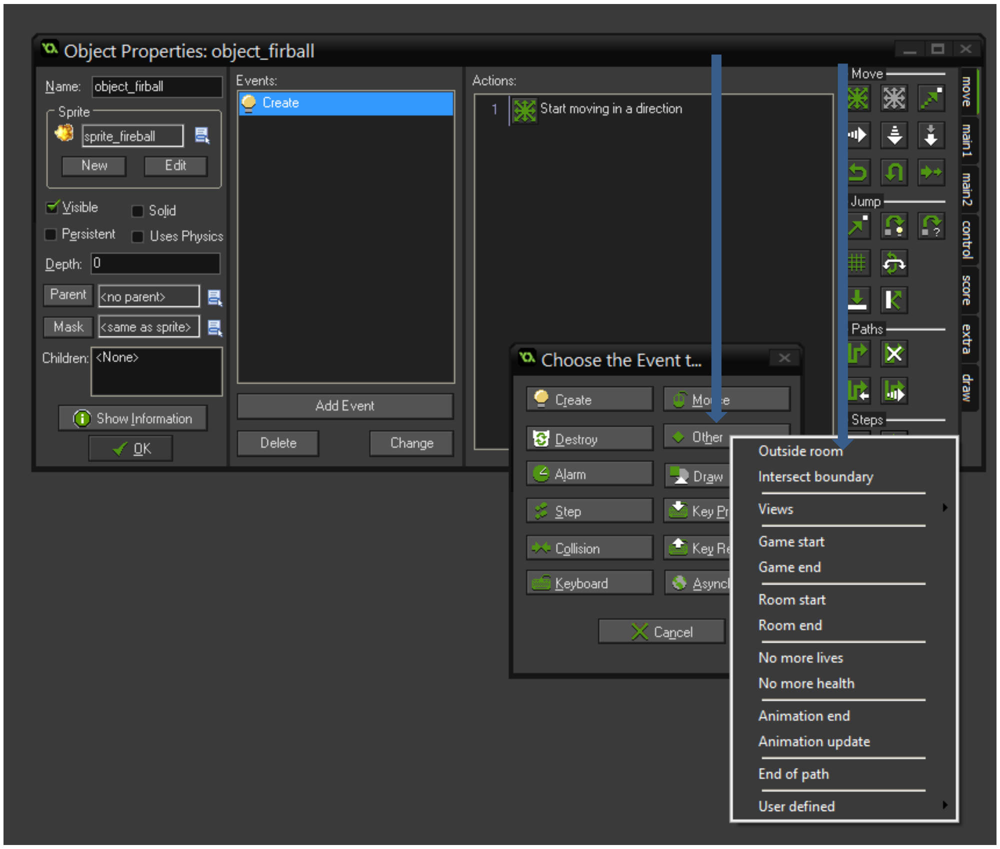
○ Object:

Directions: →

Speed: 32

□ Relative

3) Click the Add Event button again, select Other events, and pick Outside room.



Object Properties: object_firball

Name: object_firball

Sprite: sprite_fireball

☑ Visible □ Solid

□ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

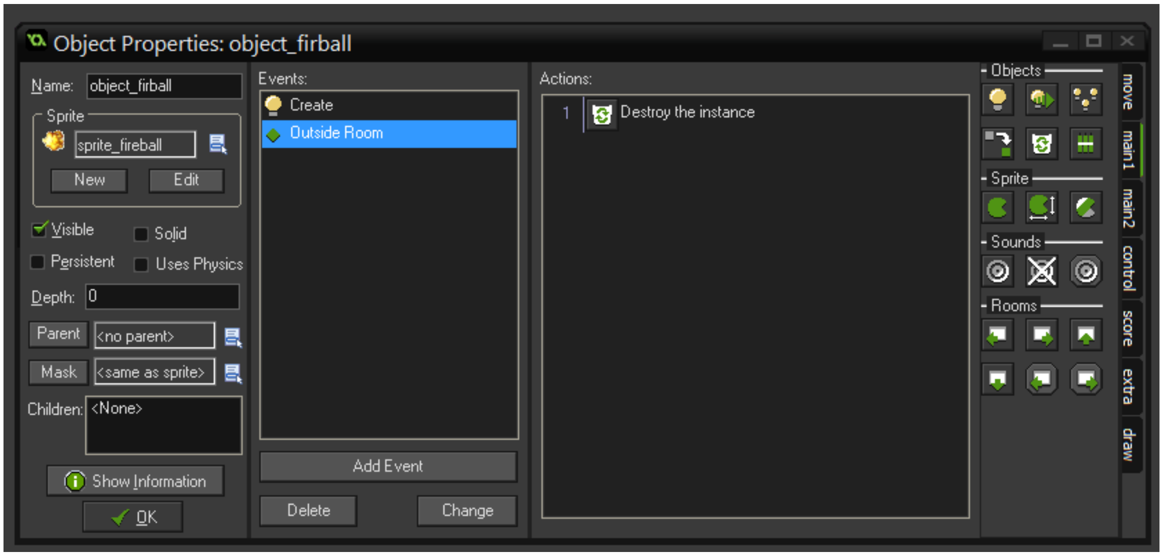
Children: <None>

Events: Create

Actions:

1 Start moving in a direction

4) Select the main1 action tab and include the **Destroy Instance** action in the **Actions** list. In the action window that pops up, simply click OK. The fireball object Properties form should now look like the figure below. Click OK.



Object Properties: object_firball

Name: object_firball

Sprite: sprite_fireball

☑ Visible □ Solid

□ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:

- Create
- Outside Room

Actions:

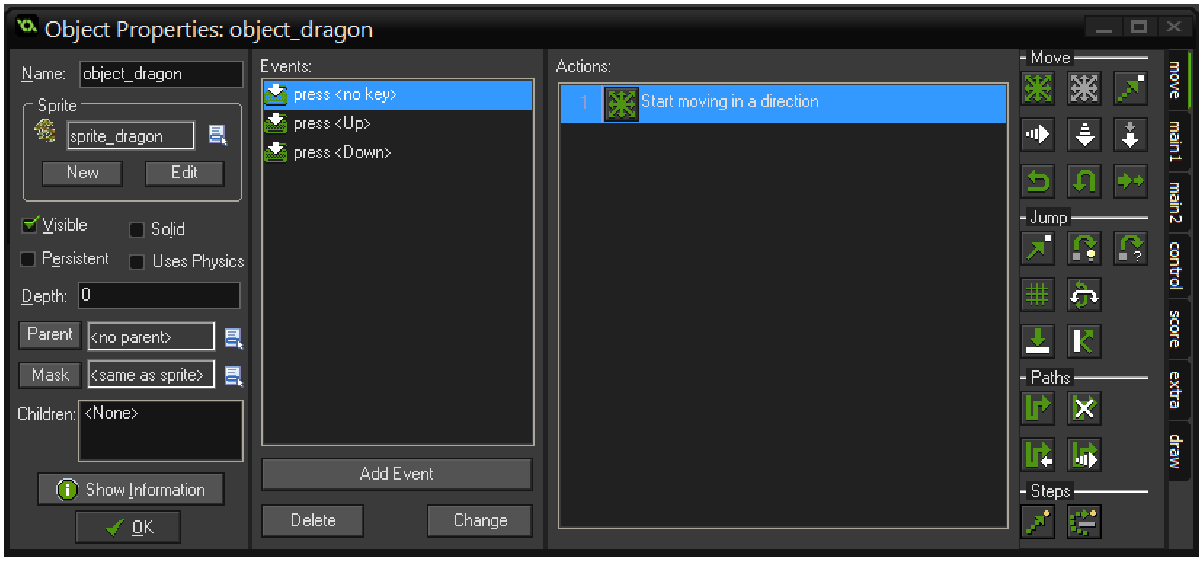
1 Destroy the instance

CAUTION: It is always a good idea to make sure that instances are deleted when they are not needed any more (when they go off the edge of the screen, for example). Even though you cannot see them, Game Maker still has to spend time updating them, and too many instances will eventually slow down the program.

Now we need to tell the **dragon object** to create instances of the fireball object when the player presses the spacebar. We do this in a similar way to the events that make the dragon move, but this time using a **Key Press** event rather than a **Keyboard** event. **Keyboard** events happen as long as the player continues to hold down the key, but the **Key Press** events happen only once when the key is first pressed. Using a **Keyboard** event for the fireballs would create a continuous stream of fireballs and make the game too easy, so that is why we are using **Key Press** instead.

Creating a Key Press event fir the dragon object

1) Double-click the dragon object in the resource list (not the dragon sprite). This will bring back the Object Properties from for the dragon object.



Object Properties: object_dragon

Name: object_dragon

Sprite: sprite_dragon

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent <no parent>

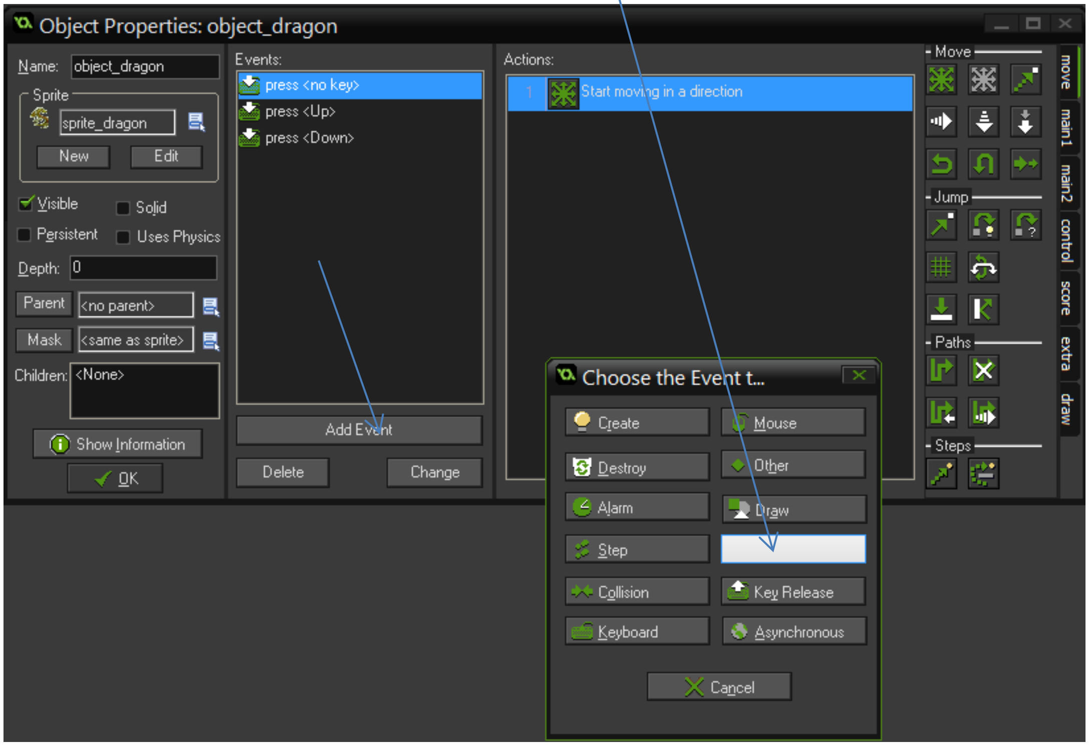
Mask <same as sprite>

Children: <None>

Events:
- press <no key>
- press <Up>
- press <Down>

Actions:
1 Start moving in a direction

2) Click the Add Event button. Select the **Key Press** event and then choose <**Space**> from the pop-up menu.



Object Properties: object_dragon

Name: object_dragon

Sprite: sprite_dragon

☑ Visible ☐ Solid ☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

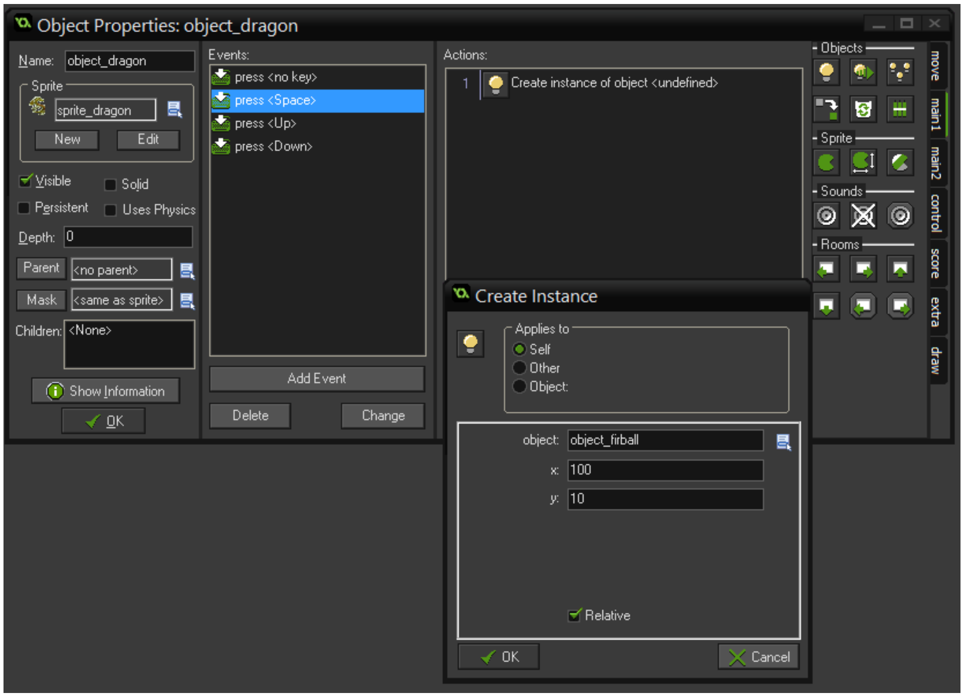
Events:
- press <no key>
- press <Up>
- press <Down>

Actions:
1 Start moving in a direction

Choose the Event t...

3) Select the main1 action tab and include the **Create Instance** action in the Action list.

4) In the action window that appears, you need to specify which type of instance to create and where on the screen it should be created. Select the **fireball object** from the menu, enter a value of 100 into **X** and 10 into **Y,** and select the **Relative** checkbox. As shown in the below screenshot.



Name: object_dragon

Sprite: sprite_dragon

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:
- press <no key>
- press <Space>
- press <Up>
- press <Down>

Actions:

1 Create instance of object <undefined>

Create Instance

Applies to: ● Self ○ Other ○ Object:

object: object_firball

x: 100

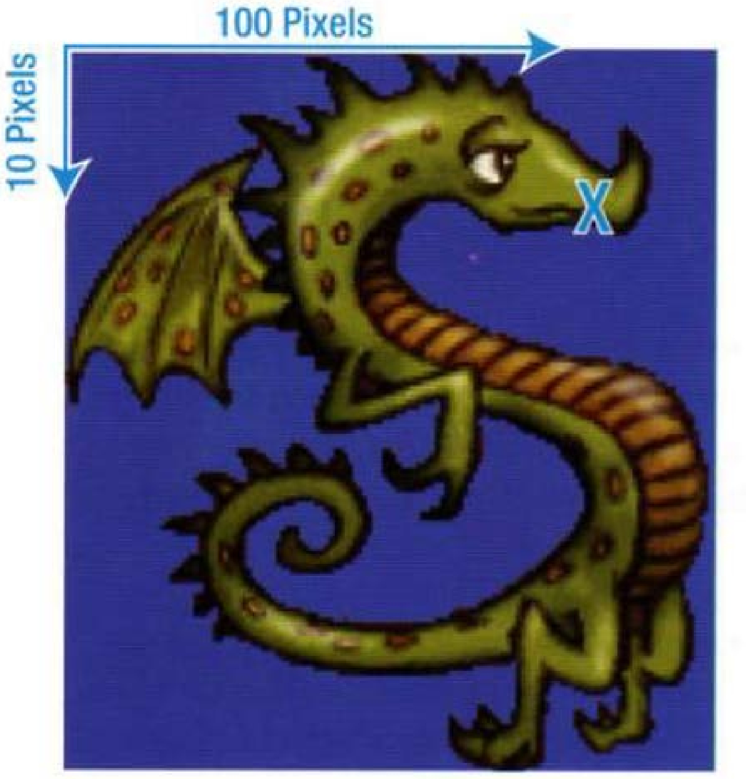
y: 10

☑ Relative

5) Click **OK** to close the action window and click **OK** again to close the Object Properties window.

NOTE: The x and y values you just entered are screen coordinates, which are used to indicate the positions on the game screen. Screen coordinates are measured in pixels (the tiny squares that make up a monitor display), with x value indicating the number of pixels horizontally, and y values indicating the number of pixels vertically.

We need to select **Relative option** because the fireball needs to be created on the screen in front of the dragon, in other words, relative to the dragon's position. However, the dragon's position is measured from the top-left corner of its sprite – just above the wings – and this would be a crazy place for the fireball to appear! Giving an x-coordinate of 100 moves, the fireball across 100 pixels to the right (to just above the head) and a y-coordinate of 10 brings it 1- pixels down. This creates the fireball right in front of the dragon's month and exactly where we need it. Test the game now to check that you can use the spacebar to shoot fireballs, and that they appear in the correct position.



The Demon Object

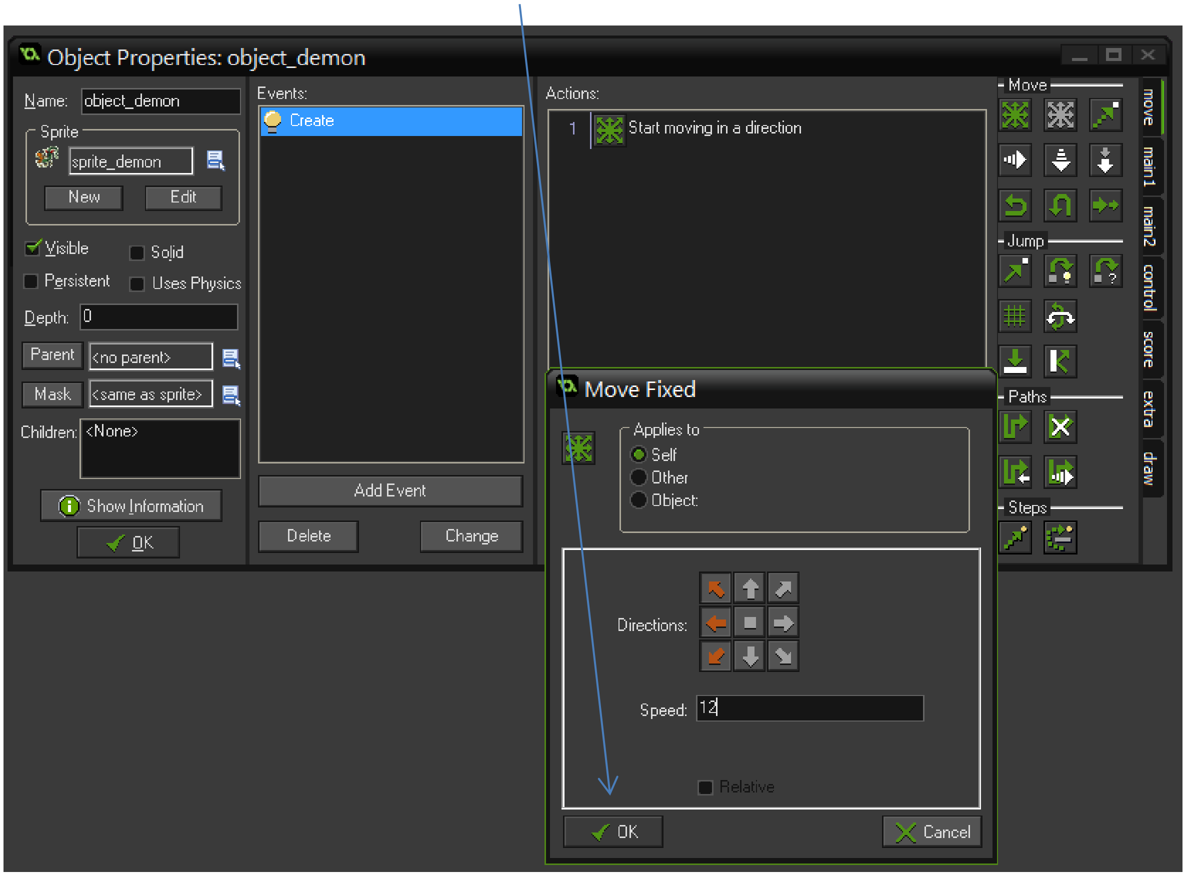
The demon object will work in the same way as the fireball, except that demons fly from right to left and are created by the boss. Also, to make demons a bit more interesting, we'll start some moving diagonally as well as horizontally. Those that head diagonally for the top or bottom of the screen will need to reverse their vertical direction when they intersect the boundary – like the boss object does. We'll also need to destroy demons when they go outside the room, like the fireball. Next are the steps to do all this.

Creating the demon object

1) **Create a new object** called **object_demon** and give it the **demon sprite**.

2) Add a **Create event** and include the **Move Fixed** action

3) **Select all three left-pointing** direction arrows and set **Speed to 12.** Selecting more than one direction causes Game Maker to randomly choose between them when an instance is created. Click the OK Button.



Object Properties: object_demon

Name: object_demon

Sprite: sprite_demon

☑ Visible □ Solid

□ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events: Create

Actions:
1 Start moving in a direction

Move Fixed

Applies to
● Self
○ Other
○ Object:

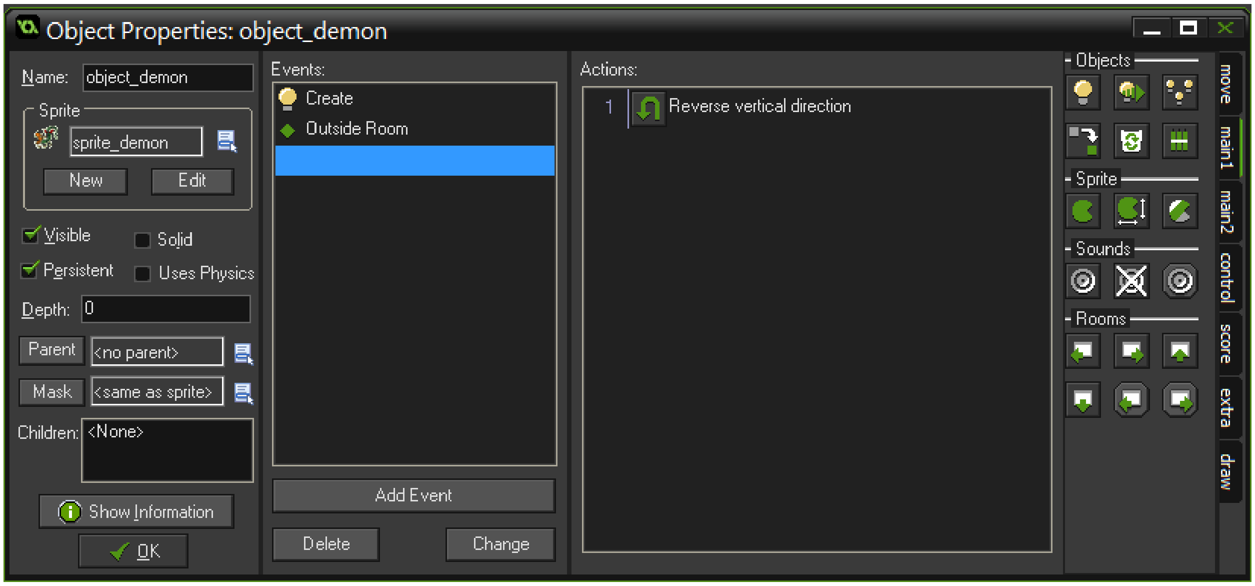
Directions:

Speed: 12

□ Relative

4) Add an **Intersect boundary event** listed under the **Other** events list, also add the **Reverse Vertical** action in it.

5) Add an **Outside room event** listed under the **Other** events list, also add the **Destroy Instance** action in it.



Object Properties: object_demon

Name: object_demon

Sprite: sprite_demon

☑ Visible ☐ Solid

☑ Persistent ☐ Uses Physics

Depth: 0

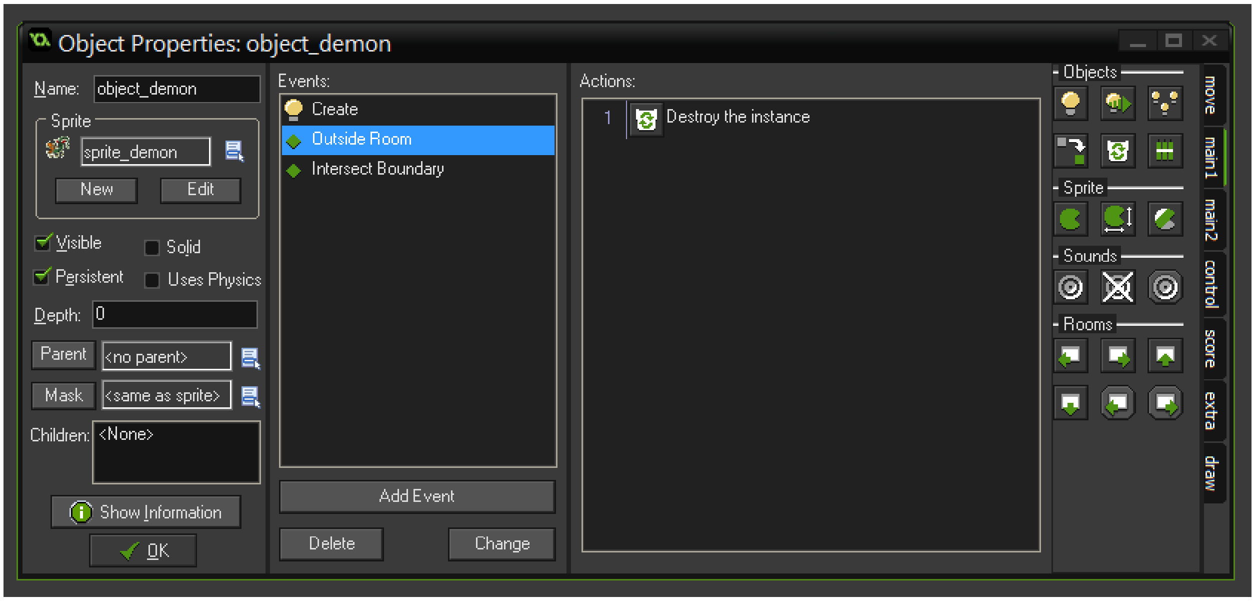
Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:
- Create
- Outside Room

Actions:
1 Reverse vertical direction



Object Properties: object_demon

Name: object_demon

Sprite: sprite_demon

☑ Visible ☐ Solid

☑ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:
- Create
- Outside Room
- Intersect Boundary

Actions:
1 Destroy the instance

NOTE: In the move, action for the demon **note** that we selected all three arrows to the left, so a random direction out of the three is selected for each demon created.

The demons will now bounce back and forth between the top and the bottom of the screen, but we also need them to react to collisions with other instances. For this, we use a collision, which happens when two sprites of different objects overlap on the screen. The first collision event we need is for when a demon collides with a fireball. This event should destroy the demon, and reward the play by increasing their score. There are a number of different actions dealing with scores, health, and lives in the

score action tab. As soon as the score changes, it will automatically be displayed in the game window caption.

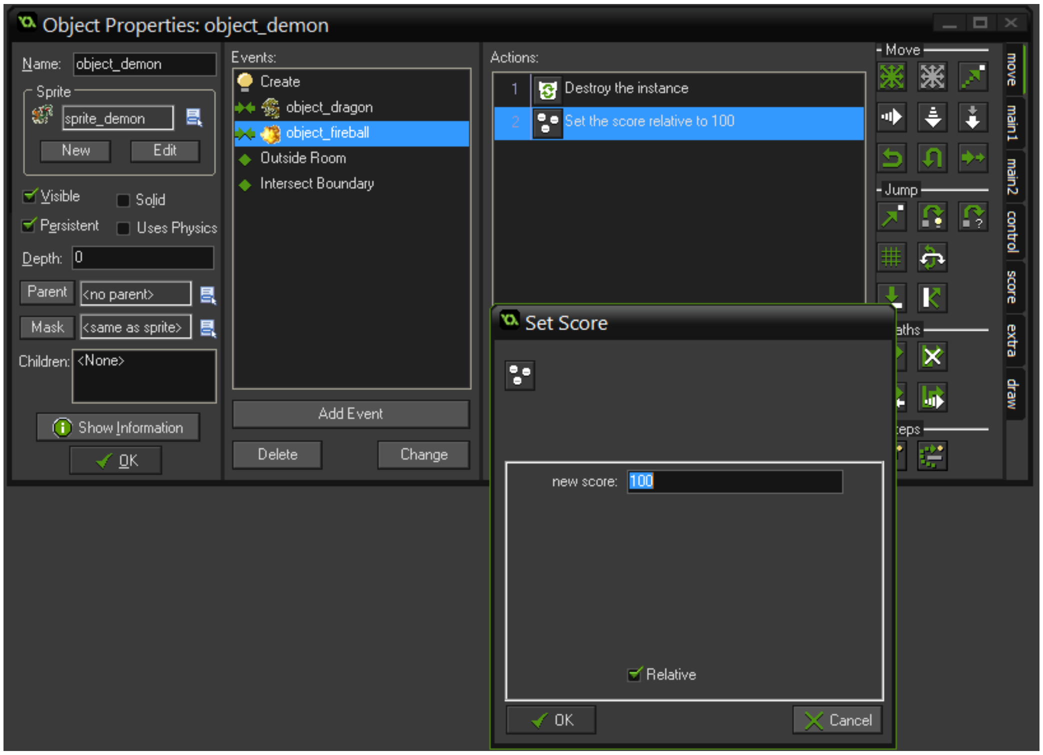
Adding an event to the demon object for colliding with the fireball

1) Click the **Add Event** button, choose the **Collision** event, and select the fireball object from the pop-up menu.

2) Include the **Destroy Instance** action from the **main1** action tab.

3) Also, include a **Set Score** action from the **score** tab. This should automatically appear below the **Destroy Instance** action in the **Actions** list. List of actions like this are carried out one after another, starting from the top of the list and working down.

4) Enter a value of **100** in the **Set Score** action form, and click the **Relative** property. This property makes the action set the score relative to the current score, so 100 will be added to the score rather than setting the score to 100.



Object Properties: object_demon

Name: object_demon

Sprite: sprite_demon

☑ Visible ☐ Solid

☑ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:
- Create
- object_dragon
- object_fireball
- Outside Room
- Intersect Boundary

Actions:
1 Destroy the instance
2 Set the score relative to 100

Set Score

new score: 100

☑ Relative

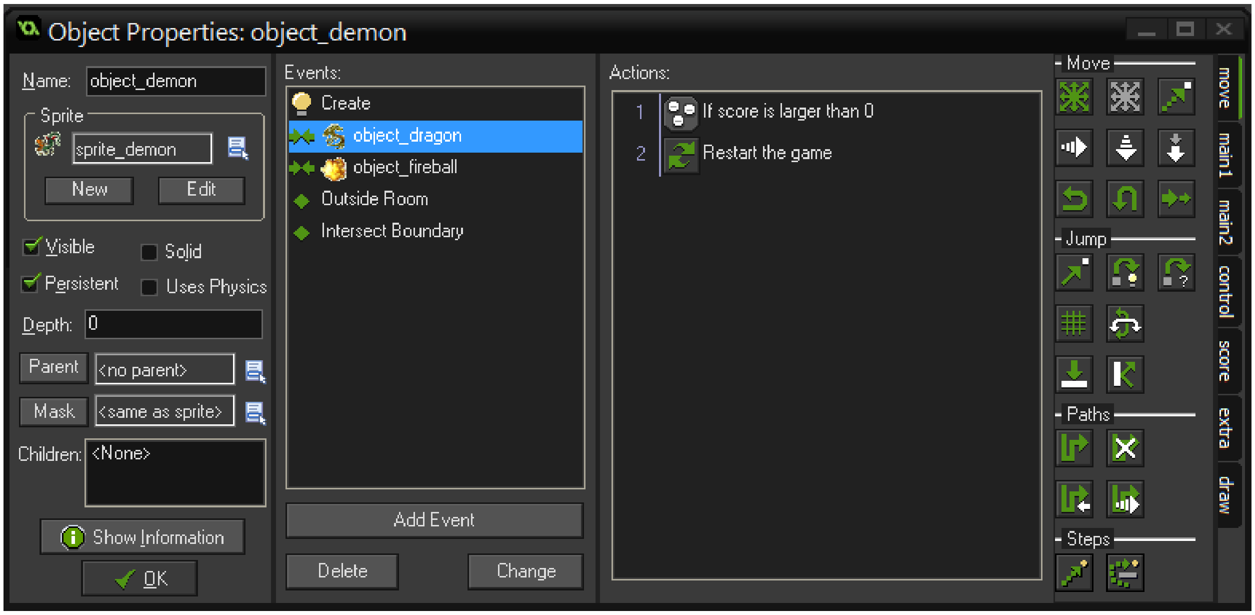
Adding an event to the demon object for colliding with the dragon:

1) Add a collision event for colliding with the dragon.

2) Include a **Test Score** action from the **score tab**.

3) Click OK to keep the default setting for this action's properties

4) Also, include **a Restart Game** action from the **main2** tab. This action has no properties.
5) The object properties window for the demon should now look like. Check to see that you have included all the demon objects events, now click **OK**.



Object Properties: object_demon

Name: object_demon

Sprite: sprite_demon

☑ Visible □ Solid

☑ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:
- Create
- object_dragon
- object_fireball
- Outside Room
- Intersect Boundary

Actions:
1 If score is larger than 0
2 Restart the game

Summoning Demons

That is it for the demon, but we still need the boss to create instances of the demon in the first place. However, we do not want the demon to appear at regular intervals because this would make the game too easy. Instead, we want there to be a random chance that a demon is created at each step of the game. A step is essentially just a short period of time in which everything on the screen moves a very short distance. There are normally 30 steps every second, so we only need there to be a very small chance that a demon is created in each step. We achieve this by using a Test Chance action, which acts like throwing a die with many sides. In each step, we throw the die, but only one side will trigger the chance action and create a demon. In this way, we create a steady, but unpredictable, flow of demons.

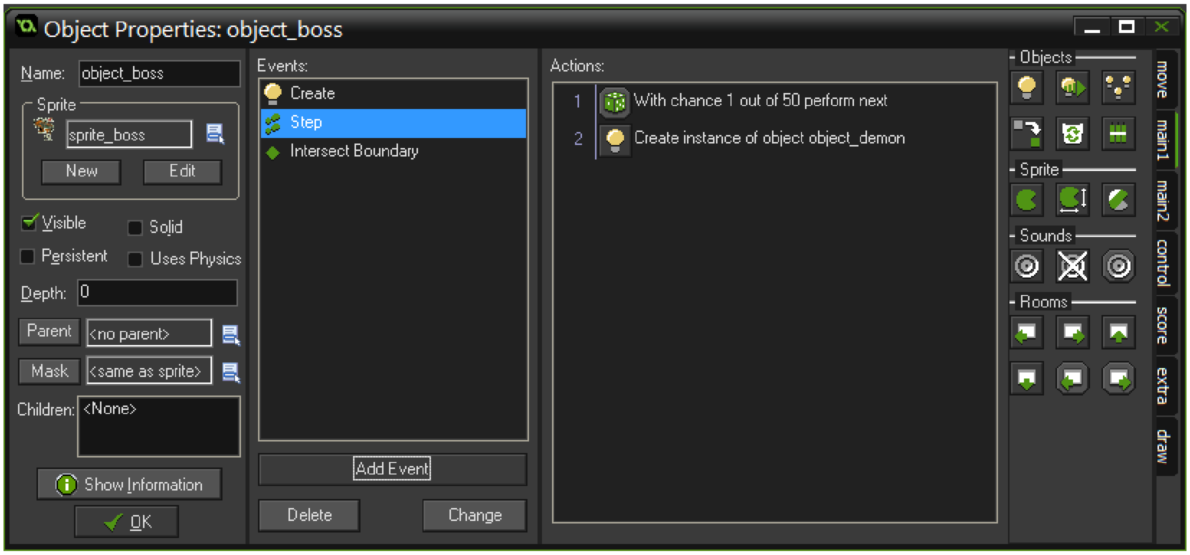
Adding a step event to the boss object:

1) Double-click the **boss object** in the resource list to bring back the Objects Properties window.

2) Click the **Add Event** button, select the **Step** event, and choose **Step** again from the popup menu.

3) Include the **Test Chance** action from the **control** tab. Set the sides of the die to **50** in the action's properties.

4) Also include the **Create Instance** action in the **Actions** list for this event. Set the properties to create a **demon object** and select the **Relative** option, so that the demon is created relative to the boss's position.



Object Properties: object_boss

Name: object_boss

Sprite: sprite_boss

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:
- Create
- Step
- Intersect Boundary

Actions:
1 With chance 1 out of 50 perform next
2 Create instance of object object_demon

The **Test Chance** action is an example of a conditional action. Conditional actions control the action that immediately follows them so that it is only performed if some condition is met. Therefore, in this case, the **Create Instance** action is only performed if the **Test Chance** rolls a 1 using a 50 sided die – otherwise it is skipped.
Click OK, save you work, and run the game to test it. Demons should now be appearing, and you should be able to shoot them with your fireballs to rack up score in the window caption. When you eventually are hit by a demon, the game restarts.

The Baby Dragon Object

We now have a game with two goals: shooting demons and staying alive. However, it is still not much fun to play, as it is far too easy too east to provide any real challenge. To increase the challenge, we are going to occasionally throw in a baby dragon along with the demons. If the players shoots a baby dragon, they will lose 300 points, but if they rescue one they will gain 500 points. This will mean the player will have to be much more careful about when to shoot, thereby increasing the challenge of the game.

Creating a new baby dragon object and its events:

1) Create a new object called **object_baby**, and give it the **baby dragon sprite**.

2) Add a **Create** event for the object and include a **Move Fixed** action in it. Set it to move **left** with a speed of **8** (slower than the demons to make life harder)

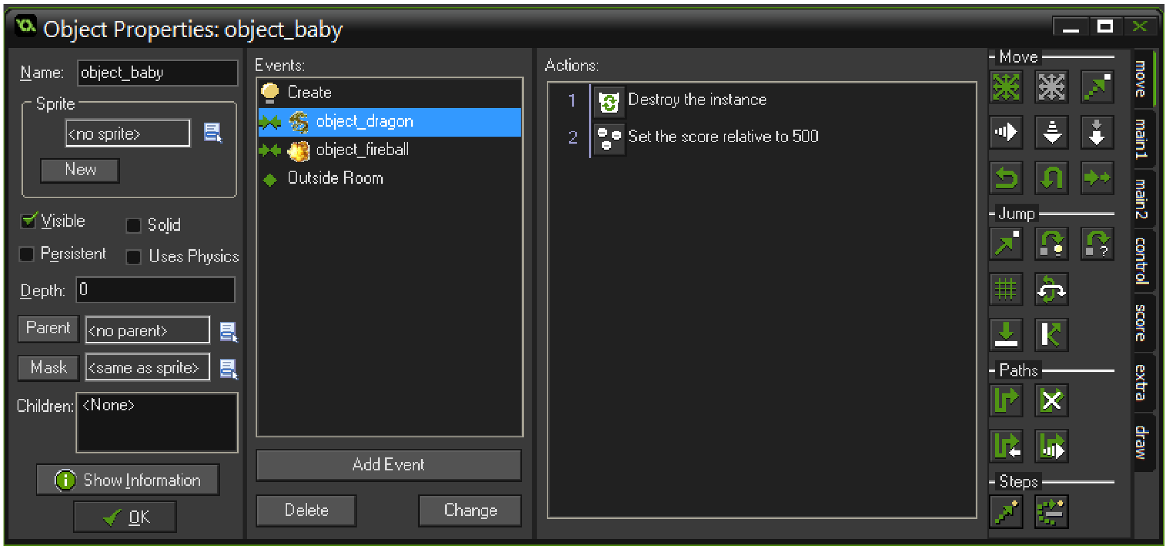
3) Add an **Outside room event** (in **Other** events) and include a **Destroy Instance** action from the **main1 tab**.

4) Add a **Collision event** with the **fireball object** and include a **Destroy Instance** action in that as well.

5) Also, include a **Set Score** action in the **collision event** with a value of **-300** and the **Relative** property selected. This will subtract 300 from the players score.

6) Add a **Collision event** with the **dragon** object and include the **Destroy Instance** action in it.

7) Also, include the **SetScore** actin with a value of **500** and the **Relative** property selected. This will add 500 to the players score. The baby dragon object should now look like shown below.


Object Properties: object_baby

Name: object_baby

Sprite: <no sprite>

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:
- Create
- object_dragon
- object_fireball
- Outside Room

Actions:
1. Destroy the instance
2. Set the score relative to 500

Now we need to make the boss randomly release baby dragons. This is exactly the same as for the demons except we will use a value of 100 for the die so that they are created less often.

Editing the boss object - too randomly create baby dragons:

1) Reopen the object properties window for the **boss object**.

2) Click on the existing **Step** event to select it and view its actions.

3) Include another **Test Chance** action in the **Step event.** Set the sides of the **die** to be **100** in the actions properties.

4) Include the **Create Instance** action below the new **Test Chance** action in the **Actions** list. Set the properties to create a baby object and select the **Relative** option.

That completes the second phase of our game! All the gameplay elements are now in place. Save the game and careful **test**, it to make sure it works **correctly**.

Backgrounds and Sounds

The first improvement we will make is to add a background to the room. Backgrounds are another type of resource, like sprites, rooms, and objects. I have created an image that is exactly the same size as the game window (640 x 480) This needs to be loaded into a new background resource, which can then be assigned to a room.

Creating a new background resource and assigning it to a room:

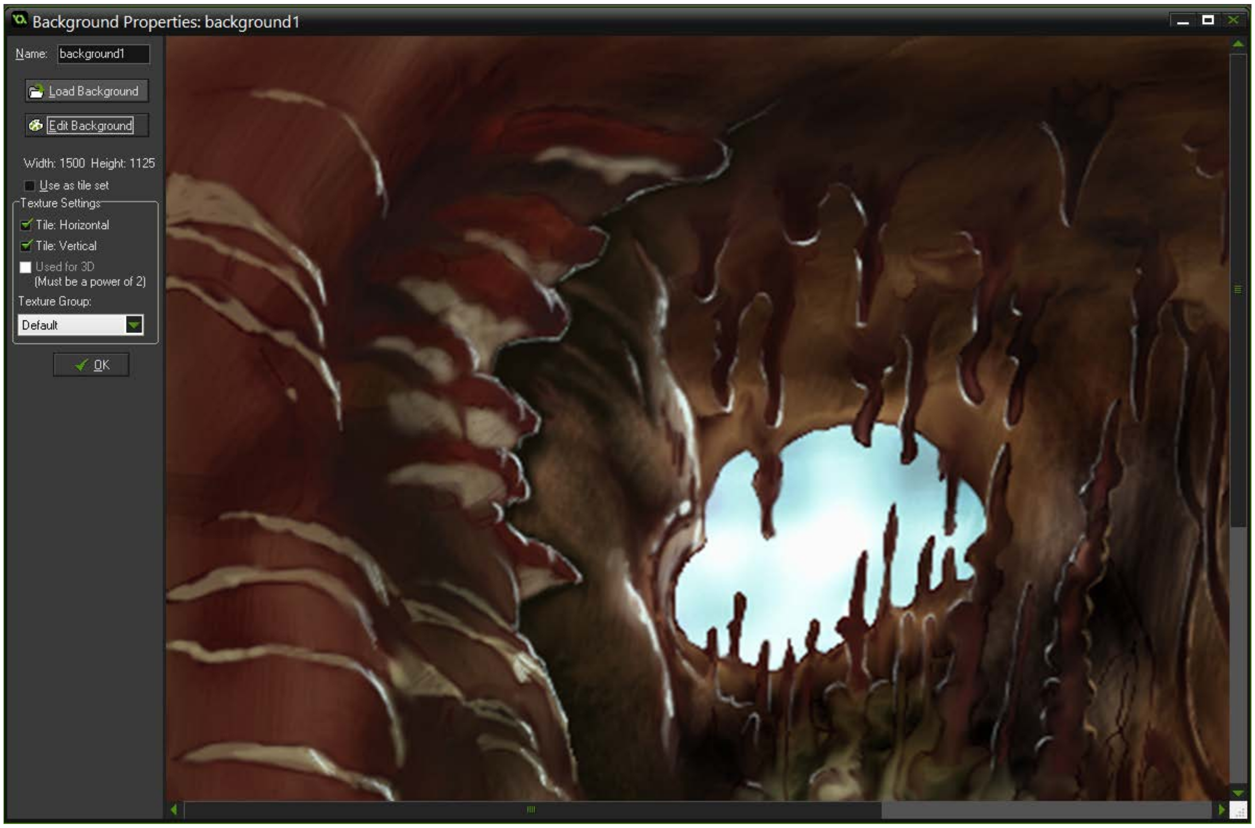
1) Select **Create Background** from the **Resource menu**

2) Call the background **background_cave**, and click the **Load background** button. Select the background.bmp image **from your resource folder**.

3) Click the OK button to close the Background Properties window.

4) Reopen the properties window for the **room_main** by double-clicking on it.

5) Select the **backgrounds** tab in the Room Properties Window. Click the menu icon to the right of where it says <no background> and select the new background from the popup menu. The Room Properties now look like the image below:



Name: background1

Width: 1500 Height: 1125

□ Use as tile set

☑ Tile: Horizontal

☑ Tile: Vertical

□ Used for 3D (Must be a power of 2)

Texture Group: Default

6) Close the Room Properties Window by clicking the green Check mark at the top left corner.

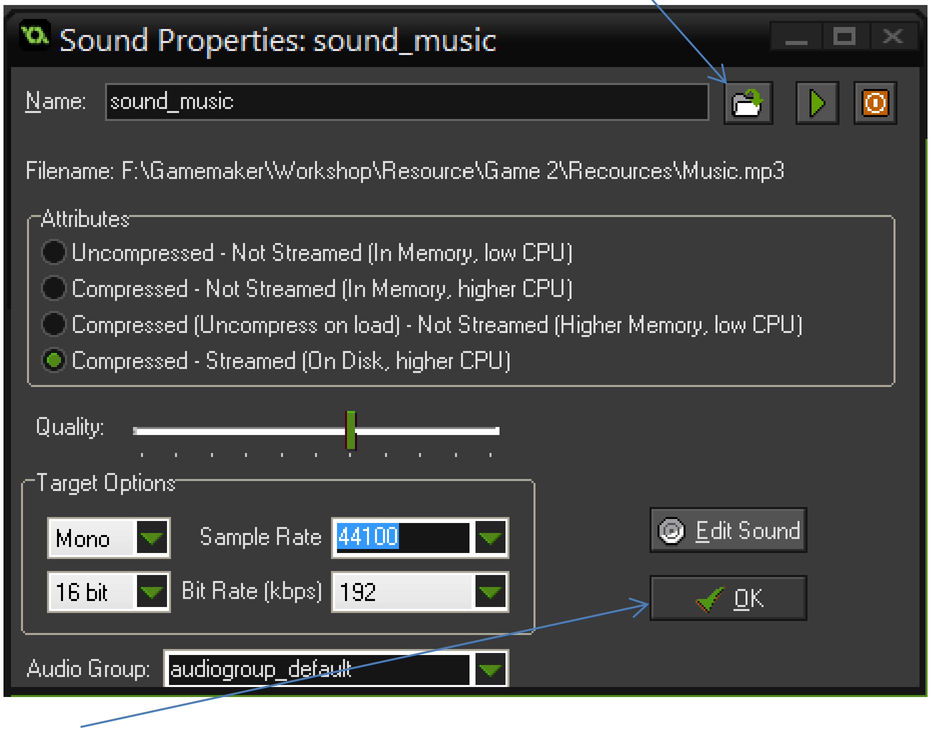
Background Music

The next step is to add some atmospheric music. Sounds are another kind of Game Maker resource for including both sound effects and music. We need to craet a sound resource for the music and them set up an action to start the music playing. We will include this action in the Create event of the boss object so that it starts playing at the beginning of the game, but it would work just as well in the dragon object too.

Creating a music sound resource and playing it in the boss object:

1) Select the Create Sound from the Resources menu and let 's call it sound_music.

2) In the properties window that appears, click **Load Sound** and select the **music.mp3** file from your Game 2 resource folder.



Sound Properties: sound_music

Name: sound_music

Filename: F:\Gamemaker\Workshop\Resource\Game 2\Recources\Music.mp3

Attributes

○ Uncompressed - Not Streamed (In Memory, low CPU)
○ Compressed - Not Streamed (In Memory, higher CPU)
○ Compressed (Uncompress on load) - Not Streamed (Higher Memory, low CPU)
● Compressed - Streamed (On Disk, higher CPU)

Quality:

Target Options

Mono   Sample Rate 44100
16 bit   Bit Rate (kbps) 192

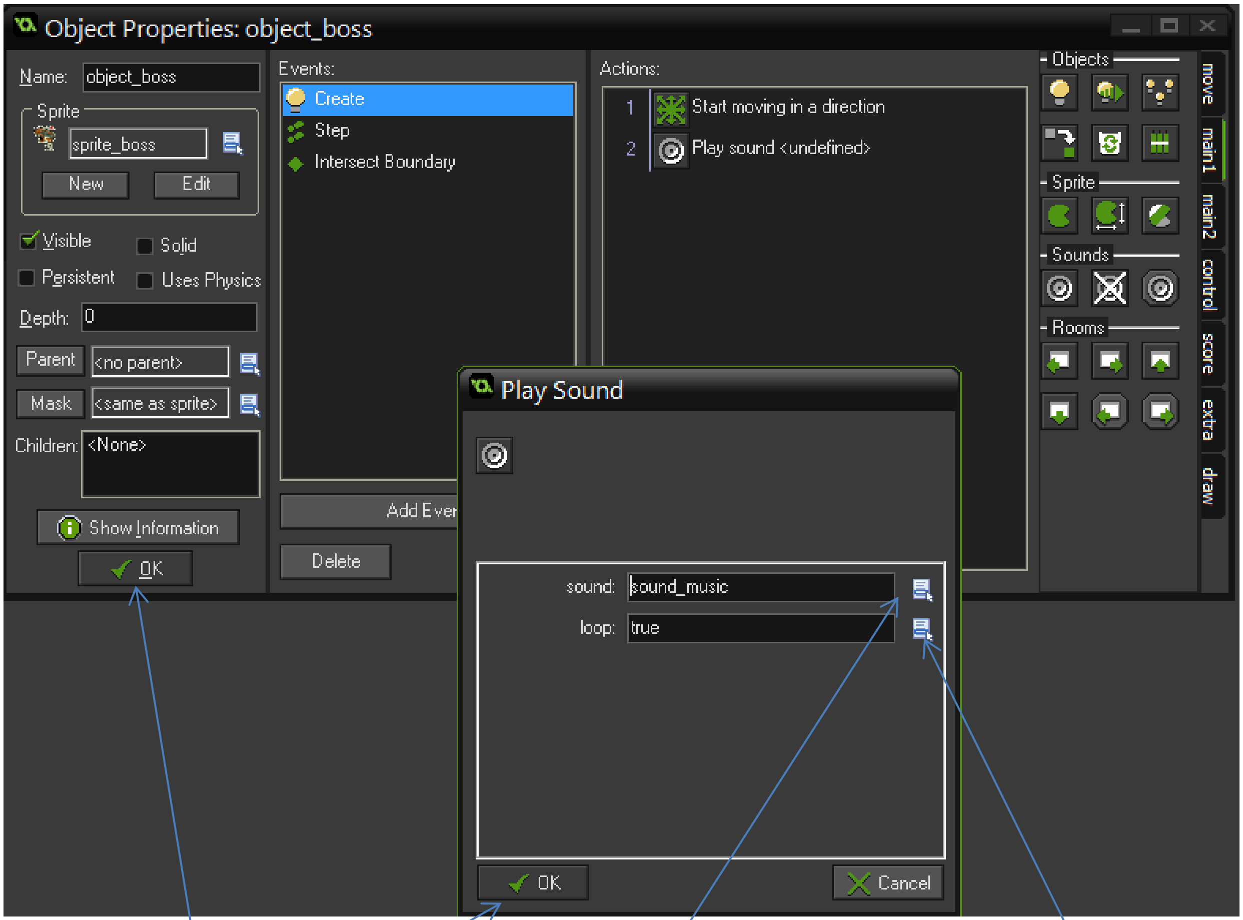
Audio Group: audiogroup_default

3) Click the **OK** button to close the window.

4) Reopen the Object Properties Window for the **boss object.**

5) Click the existing **Create event** to select it and view its actions.

6) Include a **Play Sound action** (main1 tab) in the Create event.



Name: object_boss

Sprite: sprite_boss

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:

- Create
- Step
- Intersect Boundary

Actions:

1 Start moving in a direction

2 Play sound <undefined>

Play Sound

sound: sound_music

loop: true

7) In the action properties, select the **music sound** and set the Loop property to **true**. This makes the music loop back to the start when it finishes

8) OK and OK for a second time.

Sound Effects:

Adding sound effect is another way to enhance the game, For now we will add two sound effects to our game; one for shooting a demon and one for shooting a baby dragon.

Creating and playing sound effect:

1) **Create** a new sound resource called **sound_demon**

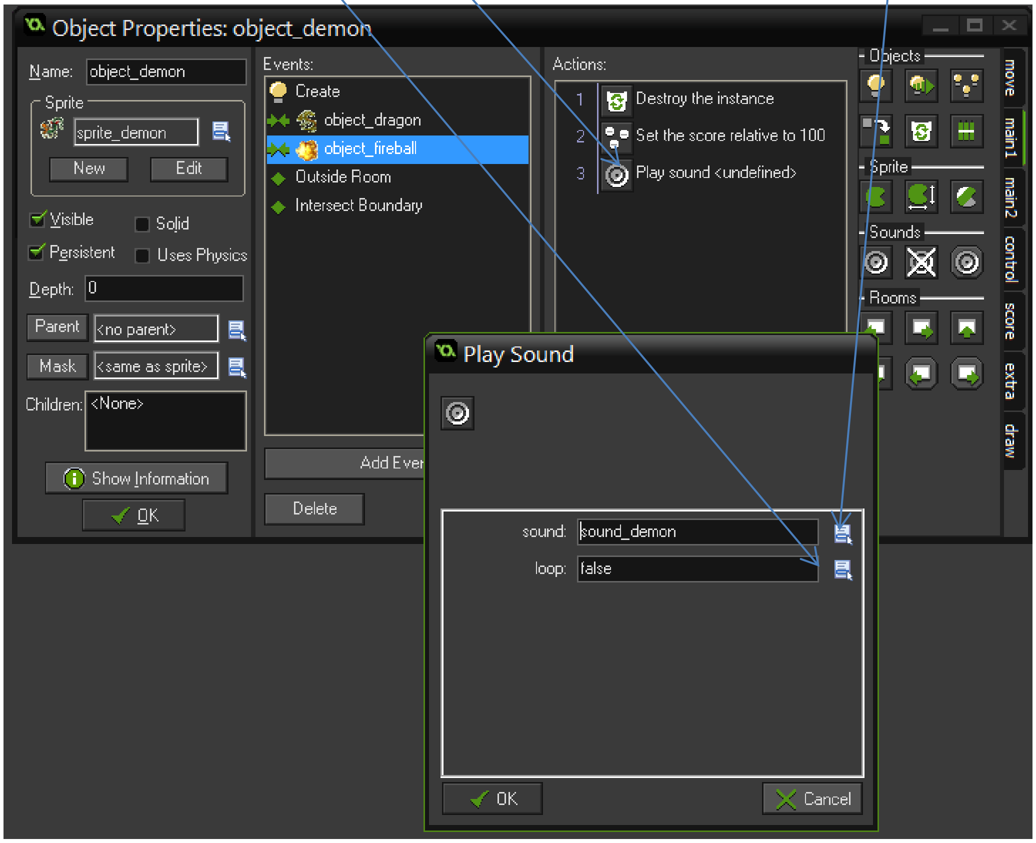
2) Load the **Demon.wav** from your resource folder.

3) Close the Sound Properties window.

4) Reopen the Object Properties Window for the demon and select the existing Collision event with the fireball object.

5) Include a Play Sound action in the collision event and select the new sound. Leave the Loop property set to false.

6) Close the action properties window and demon object Properties window.



Object Properties: object_demon

Name: object_demon

Sprite: sprite_demon

☑ Visible □ Solid

☑ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:
- Create
- object_dragon
- object_fireball
- Outside Room
- Intersect Boundary

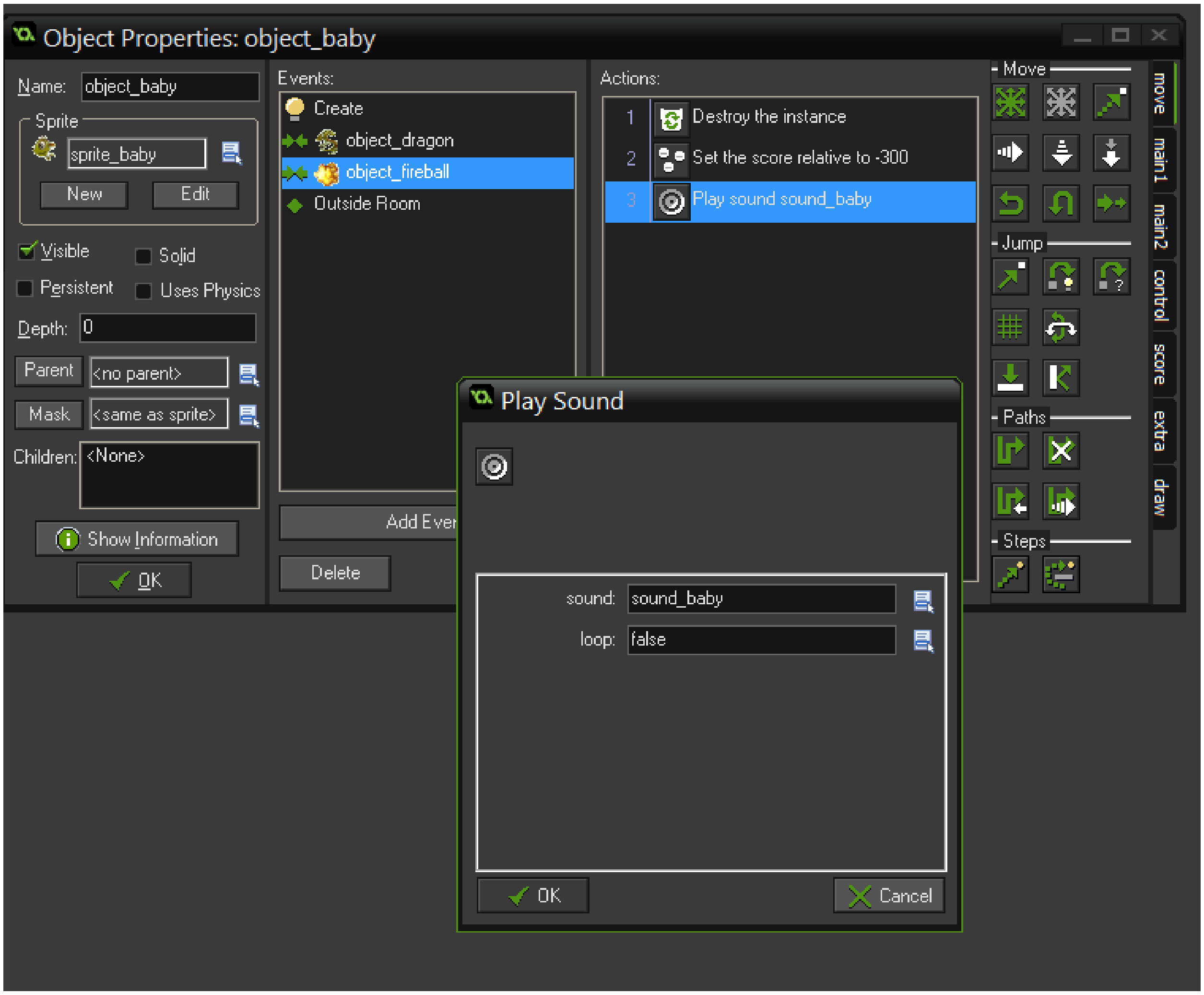
Actions:
1. Destroy the instance
2. Set the score relative to 100
3. Play sound <undefined>

Play Sound

sound: sound_demon

loop: false

7) Now please repeat the previous steps to create a sound resource for the Baby.wav. Include an action to play it in the baby dragon object's collision event with the fireball.



Object Properties: object_baby

Name: object_baby

Sprite: sprite_baby

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:

- Create
- object_dragon
- object_fireball
- Outside Room

Actions:

1 Destroy the instance

2 Set the score relative to -300

3 Play sound sound_baby

Play Sound

sound: sound_baby

loop: false

Congrats, you have finished your Second Game with Game Maker.

Great Job.

By Tom Des Lauriers November 2014