

V0.6



SPYDER

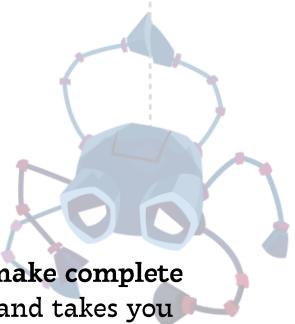
**INTRODUCTION TO GAME DEVELOPMENT
PART 1 : SPY-WIRE**



INTRODUCTION

Important Note

These worksheets complement the accompanying video tutorial and **neither will make complete sense without the other**. The video introduces the concepts behind the workshop and takes you step-by-step through the first stages of making the game to help you get started (equivalent to pages 2-9 of these worksheets). However, at some point you'll find it easier to follow written instructions, and that's when to switch over to this booklet instead. Finally, once you reach the end of the worksheets you can return to the video for a summary of the key concepts and some suggestions of how to extend and improve your game. Good luck!



Resources

You will also need to obtain the archive of images and sound effects that are used to make the game in this tutorial. You should find a download link for this archive in the text description of the video. Download this file and unzip the archive's contents into a directory on your computer before beginning (usually this involves right-clicking on the .zip file and selecting "Extract Here"). The files and folders this creates are referred to in the tutorial, so keep a note of where you have put them for future reference.

Legal Notices

This course has been created and taught by Dr. Jacob Habgood, who would like to kindly acknowledge Dr. Mark Overmars and APress for the use of example materials derived from *The Game Maker's Apprentice* (Habgood and Overmars, 2006).

The tutorial assets are derived from the original Spyder™ game by Sumo Digital Ltd (© 2020 Sumo Digital). Permission is granted to use these resources for educational use only.

A SPRITE WITH SIX LEGS

Creating a new sprite resource:

1. Right-click on the **Sprites** section of the Resources window and choose **Create Sprite**. The Sprite Properties form appears, like in Figure 1-1 below:

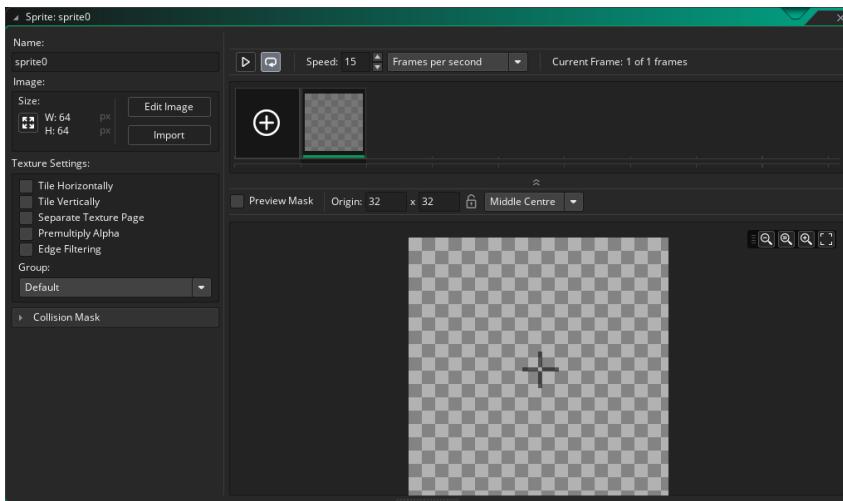


Figure 1-1: Open the Sprite Properties form for a new sprite.

2. Click the **Name** field, where it currently says `sprite0`. This is the default name created by GameMaker (Studio 2) for the new sprite, but you should rename it to `spr_agent8`.
3. Click the **Import** button. This opens the standard Windows file requester.
4. Select the required image using the file requester. The image for the spider is called `agent8_strip4.png`, and you'll find it in the **Resources** folder of the downloaded archive.

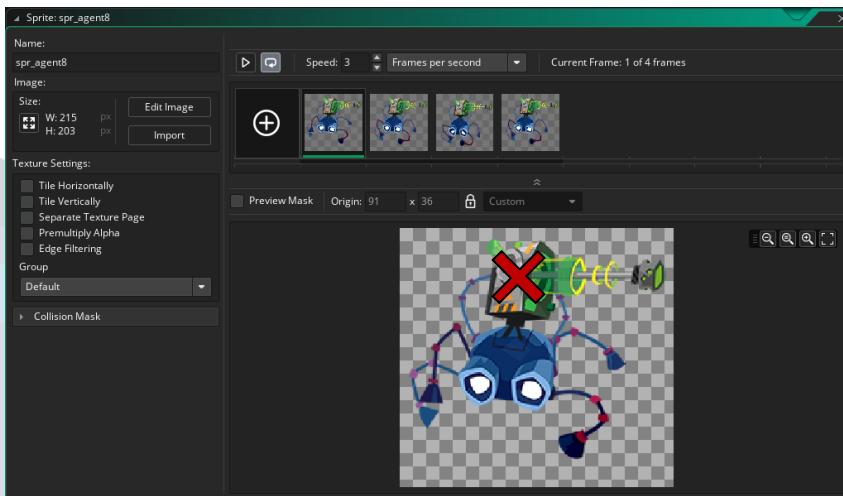


Figure 1-2: The Sprite Properties form looks like this after we load the Agent 8 sprite.

- Click on the sprite image approximately where the red **X** appears in the diagram above. This changes the **Origin** of the sprite to that point. You can think of the Origin as the point where GameMaker ‘holds’ the sprite from as it moves it around the screen. The co-ordinates of the origin are shown just above the sprite: ours are **91 x 36**.
- Click on the small lock icon to prevent you from accidentally changing the origin with a stray mouse click (you can press it again to make the origin editable again).
- Finally set the sprites’ running speed to **3** Frames per second. Your Sprite Properties form should now look like Figure 1-2. Click on the cross icon to close the form. You have now created your first sprite!

Creating the remaining game sprites:

- Right-click on the **Sprites** section of the Resources window and choose **Create Sprite from Image(s)**.
- Choose the file **fan_strip3.png**.
- Notice that the sprite name is automatically highlighted for editing in the Resource Window (on the right-hand side). Type the name **spr_fan** and press return. Its new name should now appear in the **Name** field in the Sprite Properties form as well.
- Select Middle Centre from the drop-down menu above the sprite to move the **Origin** to the exact centre of the sprite. Close the Sprite Properties form using the cross icon.
- In the same way, create a tools sprite, coin sprite, laser sprite and background sprite using **tools_strip2.png**, **coin.png**, **laser_strip2.png** and **background.png**. Give each sprite an appropriate name (using only letters and the underscore symbol). Don’t forget to set the **Origin** to the Middle Centre for each one.

YOUR BIGGEST FAN

Creating a new fan object and assigning it a sprite:

- Right-click on the **Objects** section of the Resources window and choose **Create Object**. An Object Properties form like the one in Figure 1-3 appears containing two windows, one for the object and one for its **Events**.

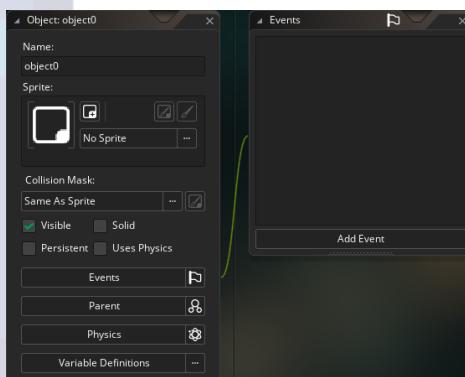
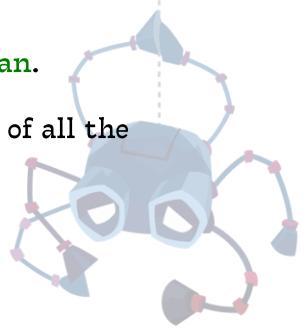


Figure 1-3. Open the Object Properties form for your new object.

2. In the **Name** field, give the object a name. You should call this one `obj_fan`.
3. Click the ellipsis icon at the end of the **sprite** field (three dots) and a list of all the available sprites will appear. Select the `spr_fan` sprite.



Events and Actions

Adding a create event for the fan object:

1. Click the **Add Event** button. The Event menu appears, as shown in Figure 1-4.

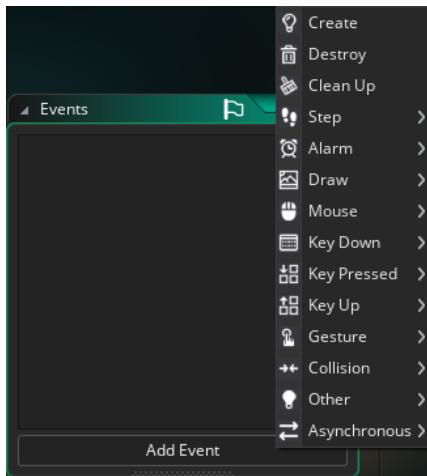


Figure 1-4. Click **Add Event** to open the Event pop-up menu

2. Click the **Create** event (lightbulb icon) to add it to the list of events. A new event window is created connected to the Events window, as shown in Figure 1-5. While the Create event is selected (highlighted in grey) this window will show us all the **Actions** that correspond to the Create event (currently none, and thus it is empty).

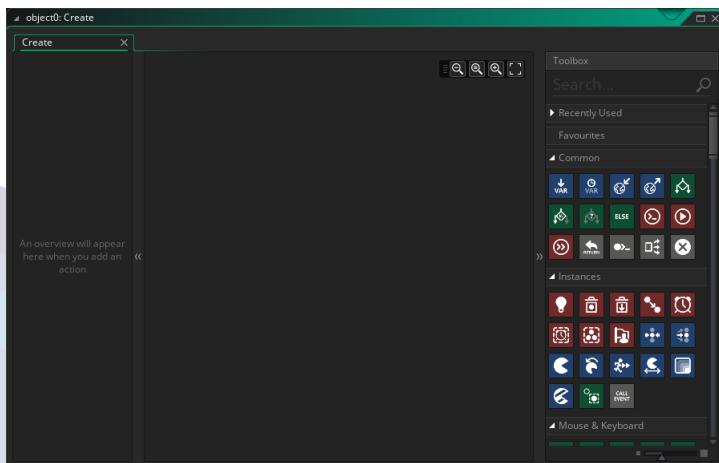


Figure 1-5. This is how the Object Properties form should look once the name, sprite, and Create event have been added.



3. Next you need to include the Set Direction Fixed action in the list of actions for the Create event. You'll find it in the Toolbox on the right-hand side, but you'll need to scroll down to the Movement section or search for "Move" to find it. Once you've located it, click and drag the icon to the empty Actions list box. The icon will expand into an action which allows you to configure specific details about how this action should operate (see Figure 1-6). Select the up arrow to select an upward direction.



Figure 1-6. You can click on one of eight arrows for the Set Direction Fixed action.



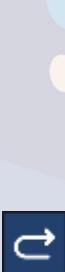
Note You can navigate around the action workspace view using the middle mouse button to drag, the scroll wheel to scroll, and the magnifying icons to zoom in and out. There are also icons to re-frame the actions in the workspace view and to reset the zoom to 100%.



4. Drag in a second action from the Movement section called Set Speed. Leave the Type set to Direction and set the Speed to 8. This will make the object move vertically 8 pixels (the tiny squares that make up a monitor display) for every step that it takes.



Note All of GameMaker's actions are organized into sections in the Toolbox. Scroll through the different sections to see all the various actions and hold your mouse over one to reveal its name and function.



Adding an intersect boundary event for the fan object:

1. Click the Add Event button.
2. Choose Other from the Event menu and select Intersect Boundary from the drop-down menu. This event will then be added and selected in the list of events.
3. Include the Reverse action in the list of actions for this event. You'll now see the form shown in Figure 2-10.



Figure 1-7. The Reverse action allows you to choose whether the direction is reversed completely or just in a single axis (horizontally or vertically).

4. Nothing needs changing on this action. The Object Properties form for the fan object now looks like the one shown in Figure 1-8.

- Double click on the fan object in the Resources list to refocus the workspace on the fan object. Click the cross in the top right of the form to close it.

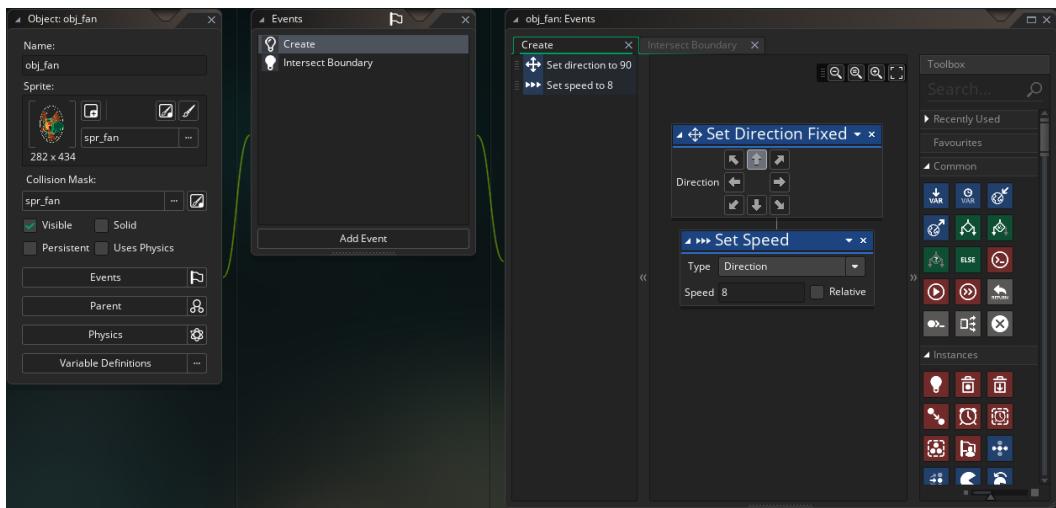


Figure 1-8. The events and actions for the fan object.

ENTER AGENT 8

Creating an Agent 8 object:

- Right-click on Objects in the Resources window and choose **Create Object**.
- Give the object a name by entering `obj_agent8` in the Name field.
- Select the `spr_agent8` sprite from the sprite selection menu.

Adding keyboard events for the Agent 8 object:

- Click the **Add Event** button.
- Choose a **Key Down** event (for when key is held down) and select **Up** from the pop-up menu (to indicate the up arrow key).
- Include the **Set Direction Fixed** action in the Actions list and select the upward direction arrow.
- Include a **Set Speed** action. Leave the Type set to Direction and set the Speed to **12**.
- Repeat the steps above to add a **Key Down** event for the **Down** key that includes a **Set Direction Fixed** action with a downward direction and a **Set Speed** action with a speed of **36** (spiders tend to drop down faster than they climb up). The Agent 8 object should now look like Figure 1-9.

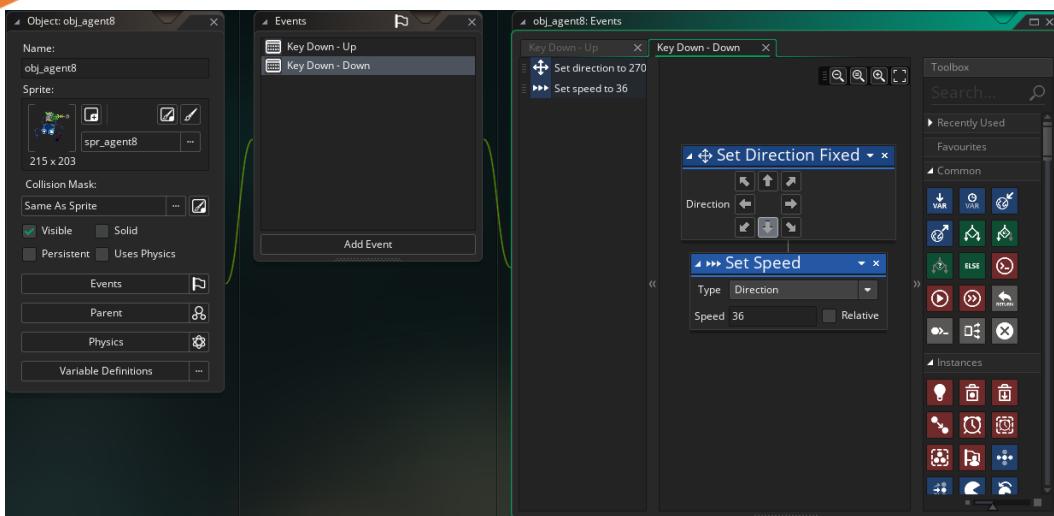


Figure 1-9. The Agent 8 object looks like this once we add the Up and Down events.

Adding a no key event for the agent8 object:

1. Click the Add Event button.
2. Choose a Key Down event and select No Key from the pop-up menu.
3. Include the Set Speed action in the Actions list for this event and set Speed to 0.
4. That's all the actions we need to make Agent 8 move up and down. Close the Agent 8 object by clicking on the cross in the top right of the object panel.

Note It is not necessary to set the direction again in order to bring an object to a halt.

ROOM WITHOUT A VIEW

Creating a new room resource:

1. Open up the list of Rooms in the Resource window (click on the little triangle) and double click on room0. The Room Editor windows will appear (see Figure 1-10).

Adding the agent8 and fan objects to the room:

1. Click the Instances layer in the top left of the Room Editor window.
2. Click and drag the Agent 8 object from the Resources window into the room grid. An instance of obj_agent8 will be placed with its origin at the mouse cursor. The position you place it becomes its starting position in the game, so put just one Agent 8 close to the left boundary of the room area. If you add it in the wrong place, just click and drag it to somewhere else or press the delete key while it is selected to remove it.

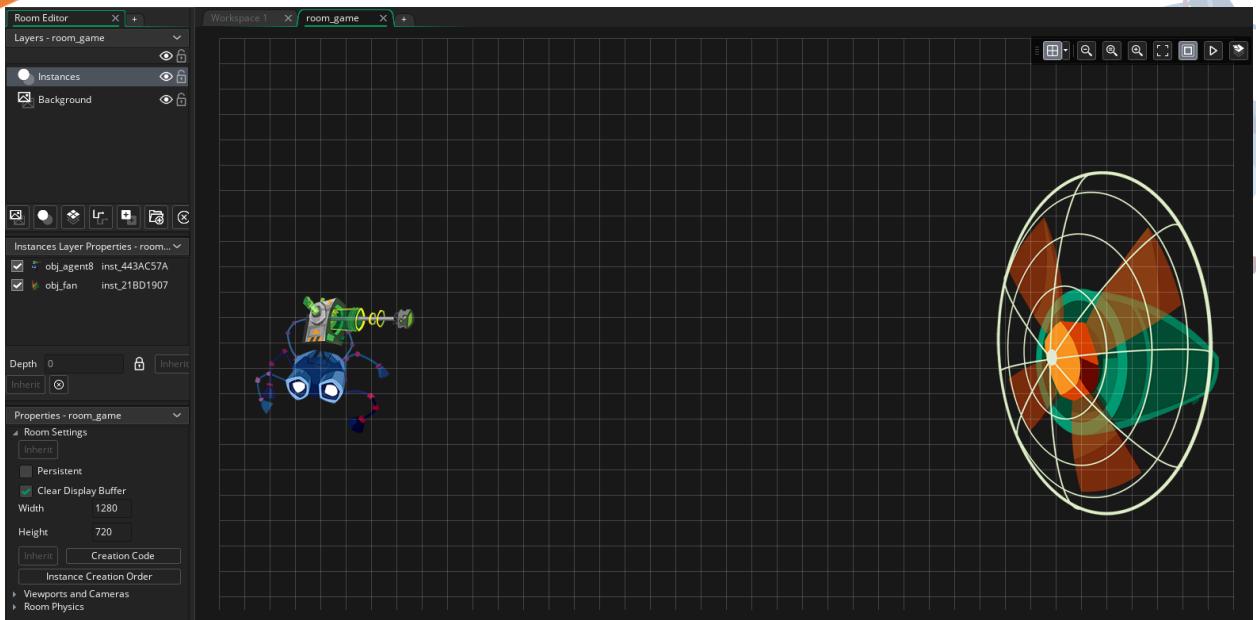


Figure 1-10. The Room Editor with Agent 8 and the fan.

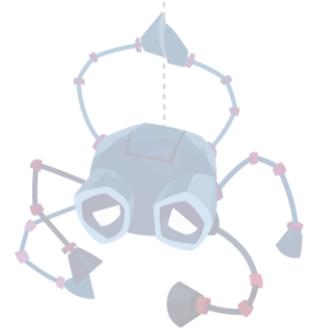
Note	To see all of the room workspace, make sure you have GameMaker application maximized and use the controls in the Room workspace to zoom around the room.
------	--

3. In the bottom left, under Room Settings set the Width of the room to **1280** pixels and the Height to be **720** pixels. The room should now look something like Figure 1-10.
4. Click and drag an instance of the fan object from the Resources window and place it close to the right edge of the room. Make sure that the whole of this sprite is completely inside the room—otherwise the object's events will not work correctly!

Saving your work and running the game:

1. Choose **Save Project** from the **File** menu (or click the disk icon).
2. When you first created your project you were prompted for a location and filename to save the project into. Note that GameMaker Studio 2 project files end with the extension **.yyp**, but GameMaker will automatically create a new directory using your chosen filename. Here it will store the project file and a number of subdirectories to hold all the different resources from your project too. Keep the project in a place where you can easily find it again (the default is in your documents directory, which is fine).
3. To run the game, press the play button on the toolbar, or press the F5 button on your keyboard. After a brief pause, a game window should appear. Check that everything works as it should, but if not then you can compare your version to **SpyWire1.yyz** in the **Projects** directory of the archive.

SPANNERS, AND LASER BEAMS



The Laser Object

Creating the laser object:

1. Right-click on Objects and choose Create Object.
2. Name the object `obj_laser`.
3. Select the laser sprite.

Adding the laser object's events:

1. Click the Add Event button and choose the Create event.
2. Include the Set Direction Fixed action in the Actions list and select the right arrow to indicate the direction.
3. Include the Set Speed action and set Speed to **64** (lasers move really fast!)
4. Click the Add Event button again, select Other events, and pick Outside Room.
5. Look for the Destroy Instance action with the Instances section of the Toolbox and add it to the Actions list. The Object Properties form should now look like Figure 1-11.
6. Close the laser object properties.

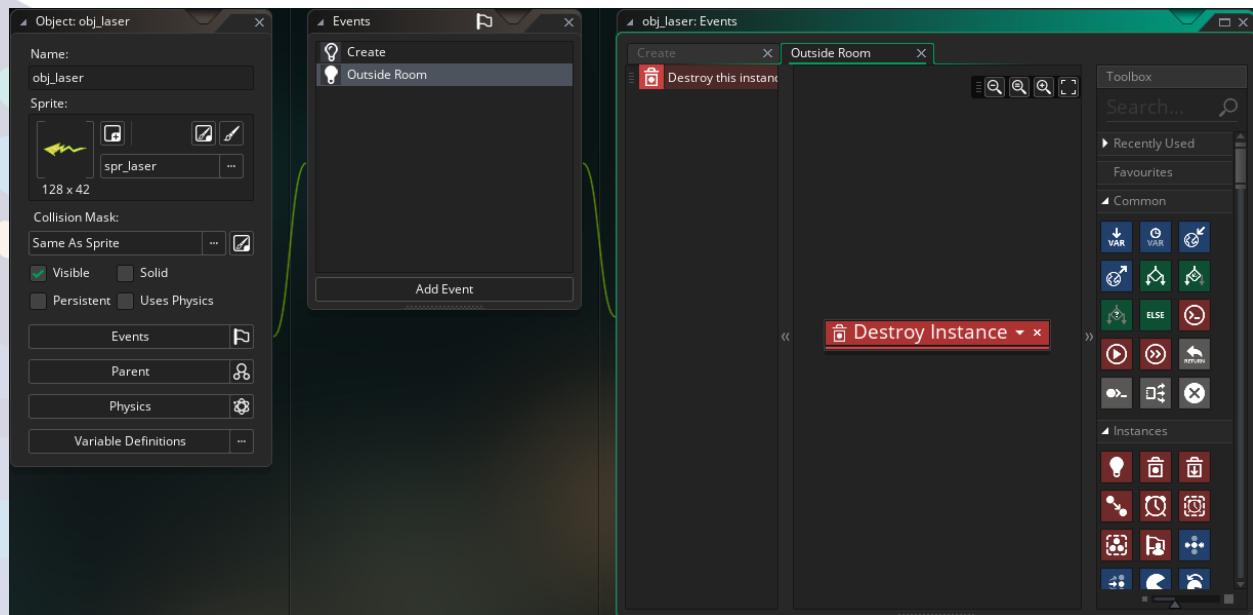
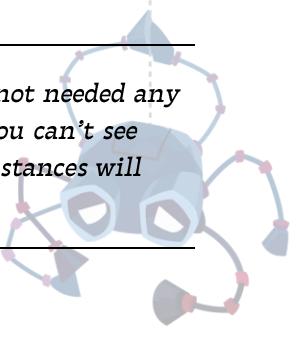


Figure 1-11. The laser object should now look like this.

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



Creating a Key Pressed event for the agent8 object:

1. Double-click the agent8 object in the resource list (not the agent8 sprite). This will bring back the object properties for the agent8 object.
2. Click the Add Event button. Select the Key Pressed event and then choose Space from the pop-up menu.
3. Look for the Create Instance action in the Instances section of the Toolbox and drag it into the Actions list.
4. There are a number of things to specify in this action, most importantly which type of instance to create and where on the screen to create it. Select the laser object from the menu, enter a value of 115 into X and 0 into Y, and select both Relative checkboxes. Figure 1-12 shows what the completed action should look like.
5. Close the agent8 object.

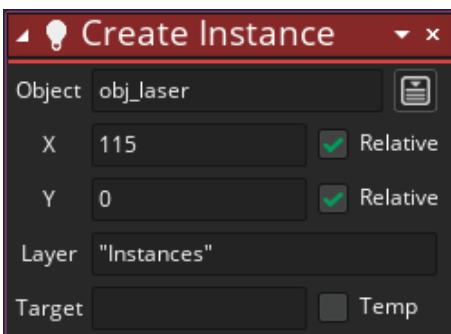


Figure 1-12. Note that we check the **Relative** property to make the laser appear relative to the position of Agent 8.

The Tools Object

Creating the tools object:

1. Create a new object called `obj_tools` and give it the tools sprite.
2. Reopen the tools sprite by clicking on the Edit Sprite button above where you just set the sprite. Note that it has different frames of animation representing different tools (a spanner and a screwdriver). The tool object will animate by rotating the sprite, so open the Collision Mask option for the sprite and set the Type to `Rectangle with rotation`. This ensures that collisions in the game will match the rotation of the object.





2. Return to the tools object and add a **Create** event. Include the **Set Direction Random** action from the **Movement** section of the Toolbox. Select all three left-pointing direction arrows. Selecting more than one direction causes GameMaker to randomly choose between them when an instance is created.



3. Include the **Set Speed** action and set **Speed** to **12**.



4. Include the **Get Random Number** action from the **Random** section of the Toolbox. Set the **Type** to **Integer** (**integers are whole numbers – this is important**) and set the minimum to **1** and maximum to **2**. Check the **Temp** checkbox to show that this random number is just being used for a short time.



4. Include a **Set Sprite** action with **Sprite** set to **spr_tools** and **Frame** set to **variable**. This will randomly select between the images for the spanner and screwdriver sprites.



5. Include a **Set Animation Speed** action with **Speed** set to **0**. This will stop the sprite from automatically animating (switching between the different tools). The action form should look like Figure 1-13.

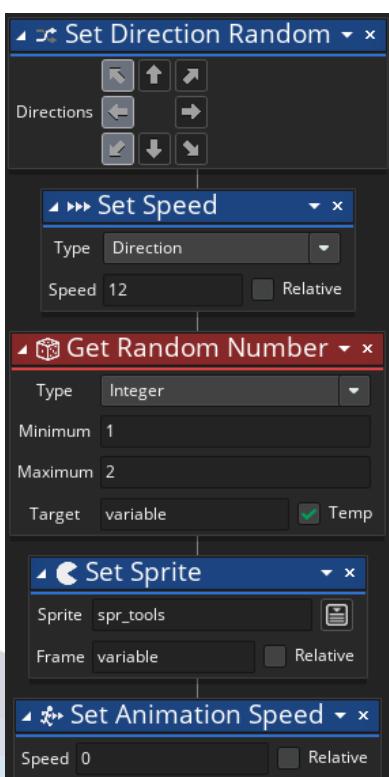


Figure 1-13. Note that all three arrows to the left are pressed in the **Set Direction Random** action, so that one of the three is selected at random for each tools instance created.



6. Add a **Step, Step** event and include a **Set Instance Rotation** action. Set **Angle** to **2** and check the **Relative** option. This will rotate the tool by 2 degrees in every frame.



7. Add an **Intersect Boundary** event (in the **Other** events) and include the **Reverse** action in it, like we did for the fan. However, this time set the **Direction** to **Vertical** so that it only reverses its vertical direction rather than doing a u-turn and heading back towards the fan!



8. Include a **Jump to Point** action with X set to **0**, Y set to **vspeed** and both **Relative** checkboxes selected.



9. Include a **Set Instance Rotation** action with Angle set to **-2** and the **Relative** option checked. These last two actions send the tool back to its previous vertical position and rotation, which helps prevent the tool getting stuck in the boundaries of the screen when it rotates.



10. Add an **Outside room** event (in **Other events**) and include a **Destroy Instance** action.



Adding an event to the tools object for colliding with the laser:



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



2. Include the **Destroy Instance** action from the **Instances** section of the Toolbox.



3. Also include a **Set Score** action from the **Instance Variables** section.



4. Enter a value of **100** in the **Set Score** action 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. See Figure 1-14.



5. This sets the score for the tool, which isn't what we're after. Carefully click on the downward pointing arrow next to the X button in the top right corner of the action. A popup will appear (see Figure 1-14) showing that this action currently **Applies To Self** (i.e. the tool). Select **obj_agent8** instead and it will now set Agent 8's score.

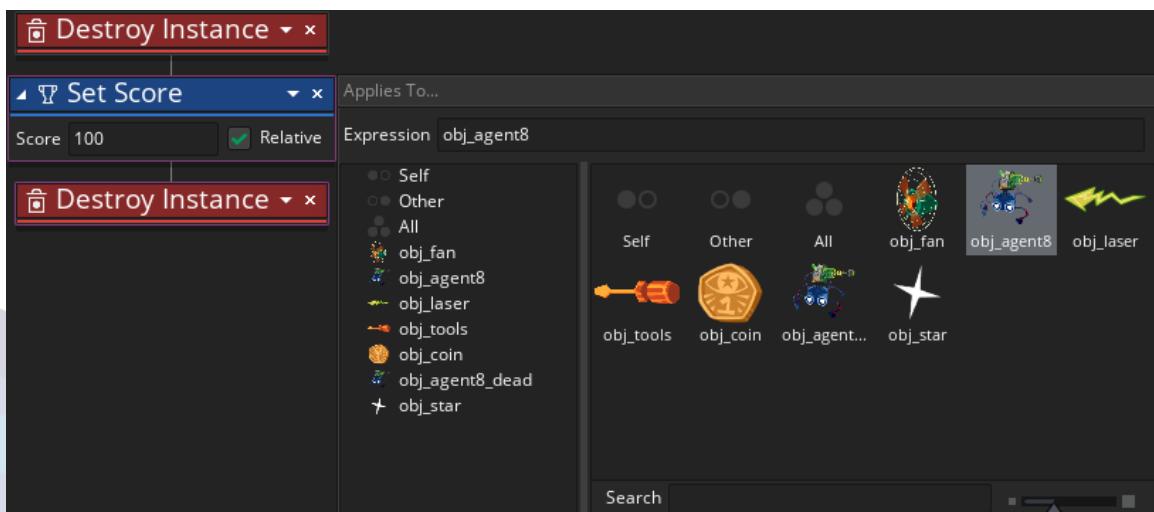


Figure 1-14. We add 100 to the player's score by setting the **Relative** property and setting **Applies To** to be the **Agent 8** object.



6. We also want the laser to be destroyed, so include a second **Destroy Instance** action and set **Applies To** to **Other** (the other thing involved in this collision event).

Note

You might wonder if you could have just selected `obj_laser` for **Applies To**, as you know that the **Other instance** is a laser object. However, it actually means something different. This would destroy **ALL** laser instances- not just the one which has collided with Agent 8. It works for setting the score because there is only one Agent 8, but if there were many instances of `obj_agent8` then this would add 100 points to **ALL** of their scores.

UNLEASHING THE TOOLBOX

Adding a step event to the fan object:

1. Double-click on the fan object in the resource list to bring back its properties.
2. Click the **Add Event** button, select the **Step** event, and choose **Step** again from the pop-up menu.
3. Include the **Get Random Number** action from the **Random** section of the Toolbox. Set the **Type** to **Integer** (**whole numbers again**) and set the minimum to **1** and maximum to **100**. Check the **Temp** checkbox to show that this random number is just being used for a short time.
4. Immediately follow this with an **If Variable** action from the **Common** section of the Toolbox. Set **Is** to **Greater** and **Value** to **98**. This means the action will only trigger if the random number is 99 or 100 (a 2/100 or 2% chance).
5. Now include a **Create Instance** action in the **Actions** list for this event attached to the **If Variable** event. This involves dragging the action onto the red word **Empty** next to the bottom right corner of the **If Variable Action**. Don't worry if you miss first time, you can also drag the action there after it is in the workspace. Set **Object** to `obj_tools` and select both of the **Relative** options, so that the tool is created relative to the fan's position. The event should now look like Figure 1-15.

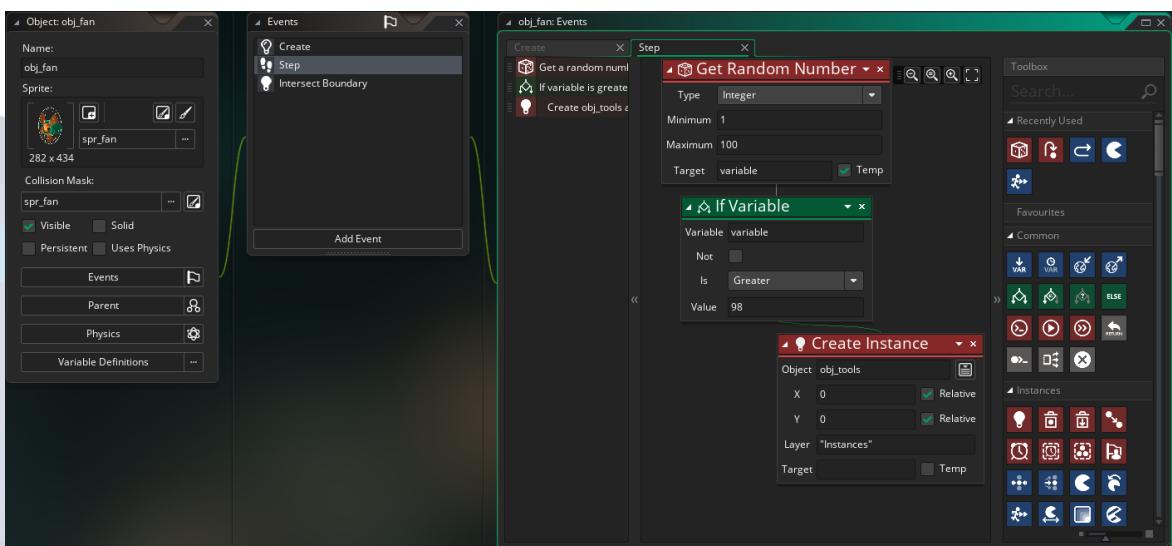


Figure 1-15. Tools are created at random in this Step event.

Displaying the thread and score in the Agent 8 object:

1. Double-click on the Agent 8 object in the resource list to bring back its properties.
2. Click the Add Event button, select the Draw event, and choose Draw again from the pop-up menu. This event controls the way the Agent 8 object is drawn.
3.  Include a **Draw Line** action with all values set to **0**, and all **Relative** options checked apart from the one for Y2. This will draw a vertical line from Agent 8's relative position (Relative X, Relative Y) to the position directly above him at the top of the screen (Relative X2, Absolute Y2).
4.  Include the **Draw Self** action from the Drawing section of the Toolbox. Normally GameMaker draws the object's sprite automatically, but as soon as we create a Draw event, we need to do it manually, so this just recreates the default drawing behaviour. This appears after the **Draw Line** action and so will be drawn on top of the line. Lists of actions like this are carried out one after another, starting from the top of the list and working down.
4.  Include the **Draw Instance Score** action also in the Drawing section. Set **X** to **640** and **Y** to **20** and DON'T check the **Relative** options. This means the score will be drawn at an absolute screen position, halfway across the screen and a little down from the top. The actions should now look like Figure 1-16.

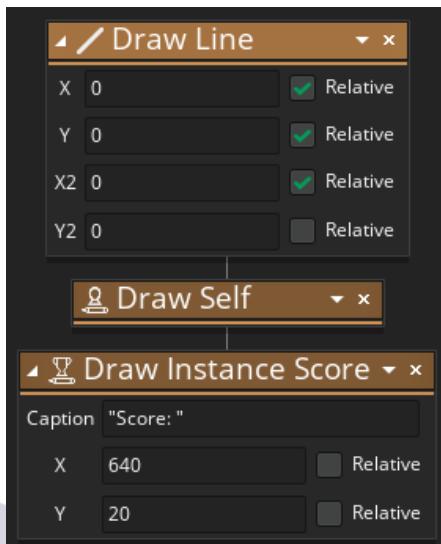


Figure 1-16. The Draw Event for Agent 8: note the Y2 Relative checkbox is unchecked.

THE AGENT RETIREMENT FUND

Creating a new coin object and its events:

1. Create a new object called **obj_coin** and give it the coin sprite.
2.  Add a **Create Event** for the object and include a **Set Direction Fixed** action with the left arrow selected.

3. Include a **Set Speed** action and set **Speed** to **8** (slower than tools to make life harder).
4. Add an **Outside room** event (in **Other events**) and include a **Destroy Instance** action from the **Instances** section.
5. Add a **Collision** event with the laser object and include a **Destroy Instance** action in that as well.
6. Also include a **Set Score** action in the collision event with a value of **-300** and the **Relative** property selected. Set this action so that it **Applies To** the Agent 8 object, so that it subtracts 300 from the player's score.
7. Add a **Collision** event with **obj_agent8** and include the **Destroy Instance** action.
8. Also include the **Set Score** action with a value of **500** and the **Relative** property selected. Set this action so that it **Applies To** the Agent 8 object and adds 500 to the player's score.
9. Add a **Step, Step** event and include a **Set Instance Rotation** action. Set **Angle** to **2** and check the **Relative** option. This will rotate the tool by 2 degrees in every frame. The coin object should now look like Figure 1-17.

10. Close the object.

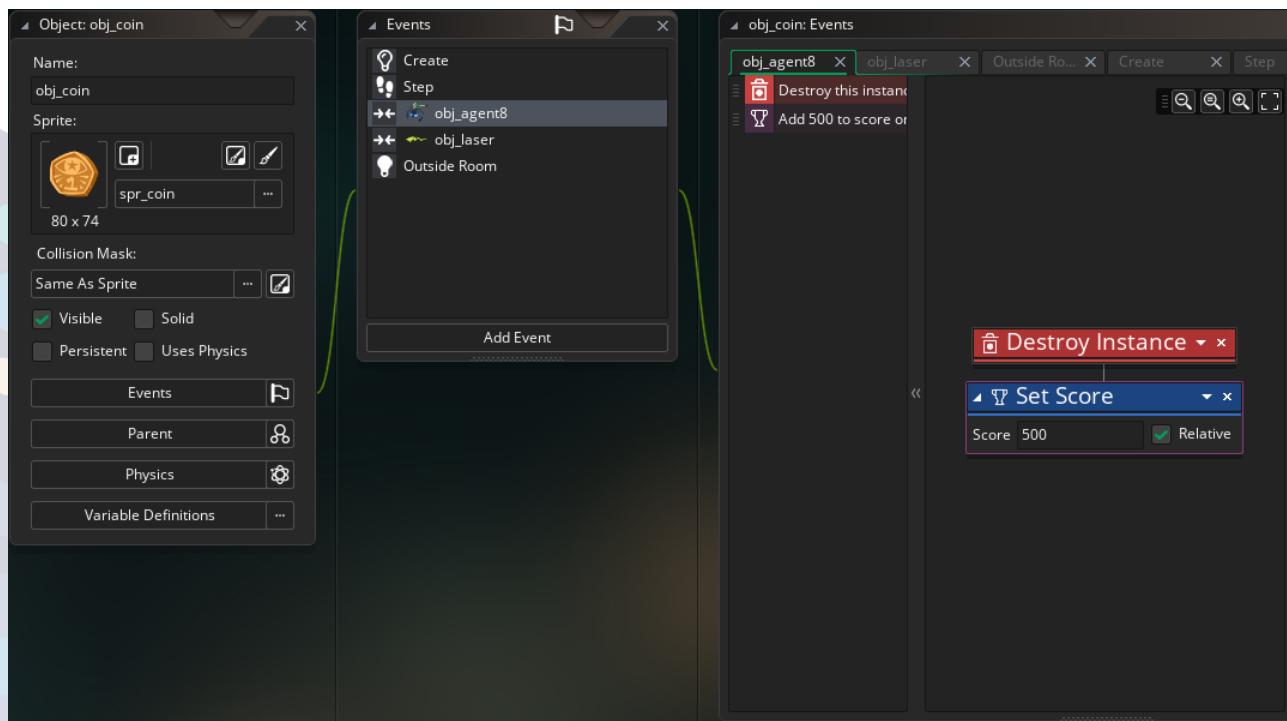


Figure 1-17. The Object Properties form for the coin object should now look like this.

Editing the fan object to randomly create coins:

1. Reopen the object properties for the fan object.
2. Double-click on the existing **Step** event to select it and view its actions.
3. Include a second **If Variable** action at the end of the **Step** event. Note that it will automatically assume you want to add this to the end of the last action and attach it to the **Create Instance** action. This is NOT what you want. Instead you will need to drag the new **If Variable** action onto the base of the old **If Variable** so that it is NOT dependent on the previous condition (see figure 1-18 below). Set **Is** to **Equal** and **Value** to **100**. This means there will be a 1/100 (1%) chance of creating a coin.
4. Now attach a **Create Instance** action to the **If Variable** action by dragging the new action onto the red word **Empty** next to the existing **If Variable Action**. Set the properties to create a coin object and select the **Relative** option, so that the coin is created relative to the fan's position.
5. Click the **Add Event** button, select the **KeyPressed** event, **Letters** and **R** from the pop-up menus.
6. Include the **Restart Room** action from the **Rooms** section of the Toolbox.

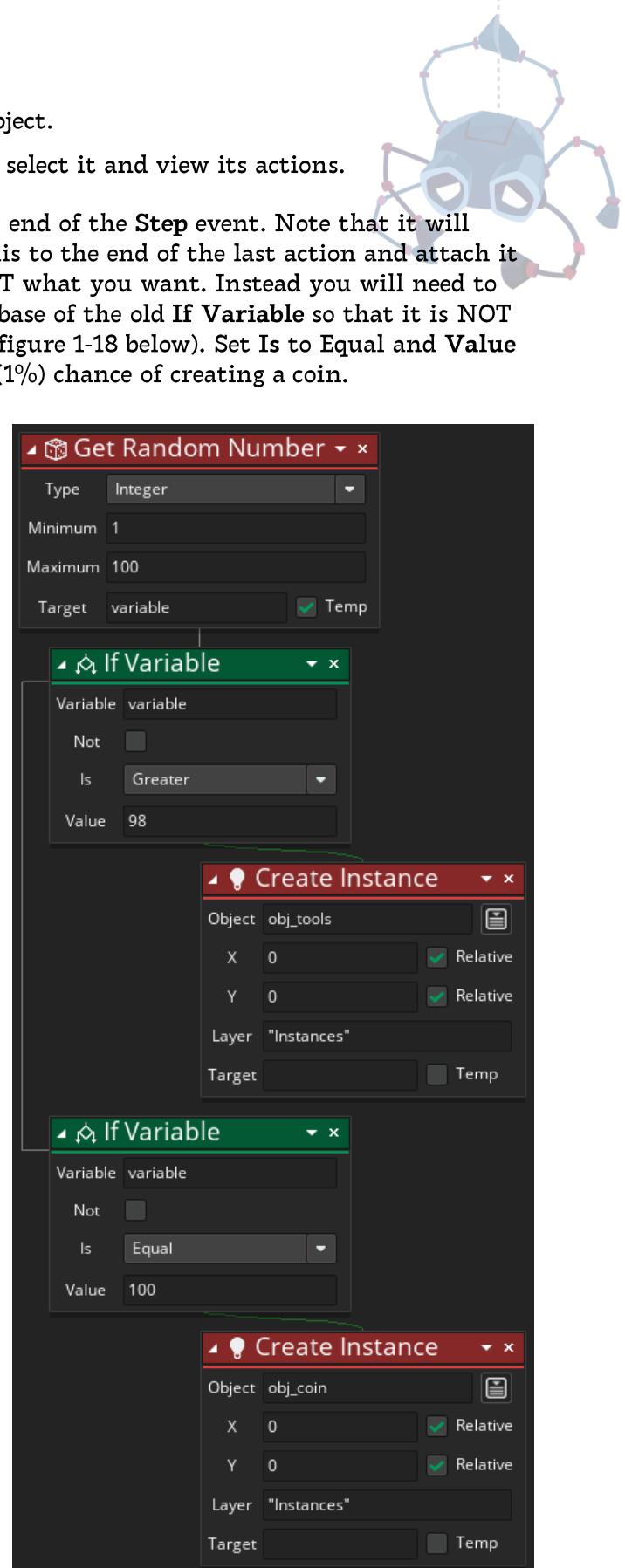


Figure 1-18 (right). The sequence of actions for the fan's **Step** event should look like this. If yours isn't working then double check that you've set the type to **Integer** on the **Get Random Number** action.

Creating a dead agent8 object:



1. Create a new object called `obj_agent8_dead` and give it the Agent 8 sprite.
2. Add a Create event and include a Set Gravity Force action with Force set to **3**.
3. Include a Set Direction Fixed action and select the top left direction arrow.
4. Include a Set Speed action with Type set to Direction and Speed set to **5**. This gives Agent 8 a little push upwards before gravity pulls him back down.
5. Add a Step, Step event and include a Set Instance Rotation action. Set Angle to **-10** and check the Relative option.
6. Add an Outside room event (in Other events) and include a Destroy Instance action from the Instances section. Close the `obj_agent8_dead` object.
7. Reopen the `obj_agent8` object and add a Collision event with `obj_tools`. Include a Change Instance action with Object set to `obj_agent8_dead`.

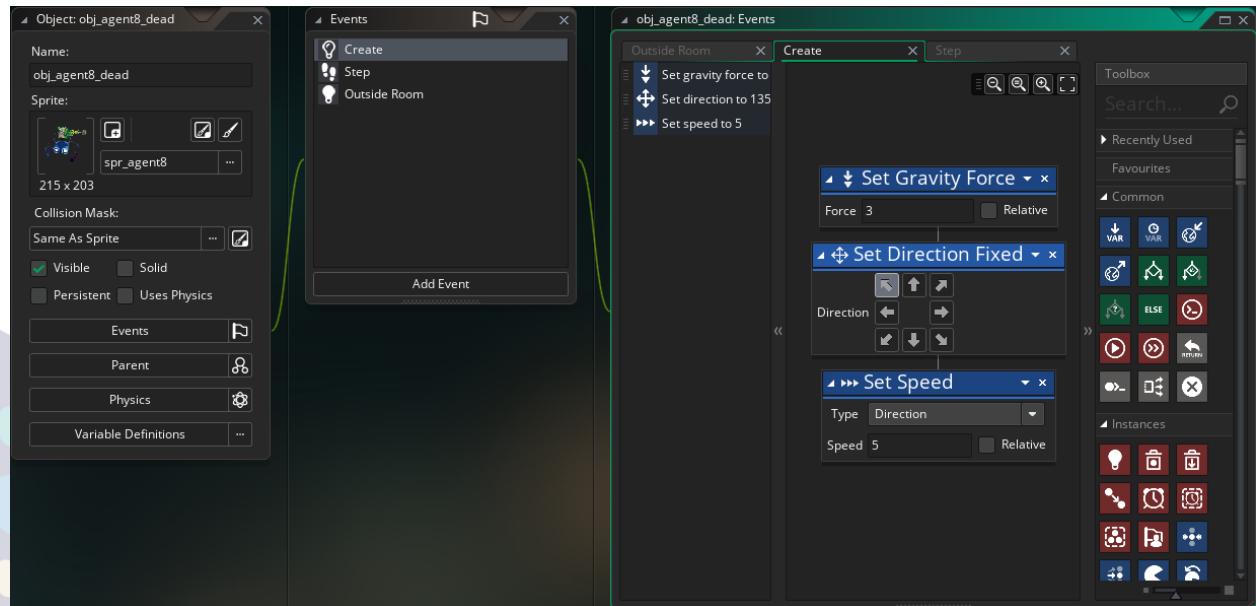


Figure 1-19. The Object Properties form for the dead Agent 8 object should now look like this.

Note

That completes the second phase of our game! All the gameplay elements are now in place. Save the game and carefully test it to make sure it works correctly. You'll also find the current version of the game in the project file Projects/SpyWire2.yyz

BACKGROUND NOISE

A Background Image

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



1. Double click on your room in the Resources menu.
2. Select the **Background** layer in the Room Editor window. This should reveal the Background Layer Properties in the pane below, with a familiar-looking Sprite selection tool. Click on the ellipsis icon and select the background sprite (see Figure 1-20).

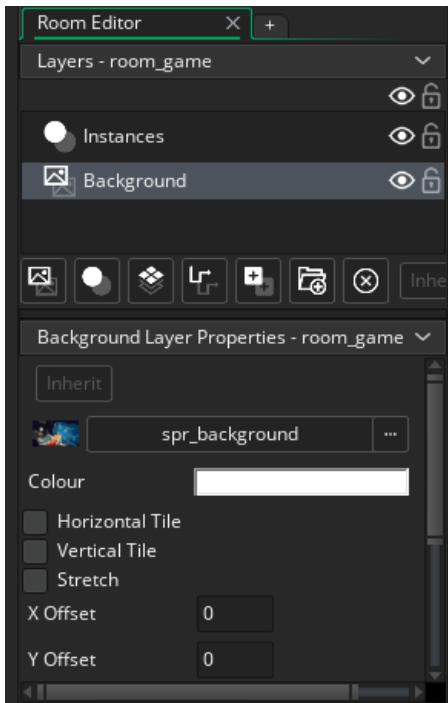


Figure 1-20. The Background Layer Properties allows you to assign background sprites.

Background Music

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

1. Right click on Sounds in the Resources window and select **Create Sound**. Name it **snd_music**.
2. In the properties form that appears, click on the ellipsis and select the **music.mp3** file from the **Resources** folder. The Sound Properties should now look like Figure 1-21.
3. Close the Sound Properties form by clicking on the cross button.
4. Reopen the fan object.
5. Add a new **Other, Game Start** event to the fan object.

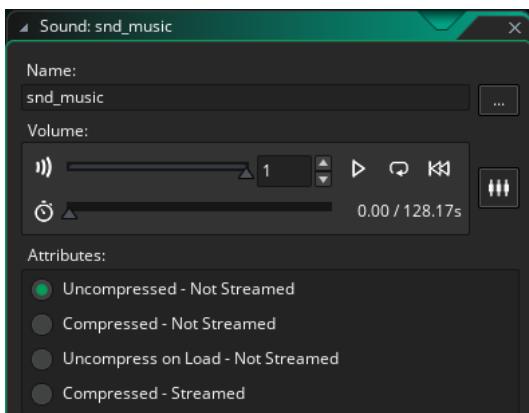


Figure 1-21. The Sound Properties form allows you to load, preview and save sound files.

6. Include a **Play Audio** action (Audio section) in the **Game Start** event. Select the music sound and check the **Loop** option. This makes the music loop back to the start when it finishes. The sound action form should then look like Figure 1-22.
7. Click **X** to close the action, and click **X** again to close the fan object.

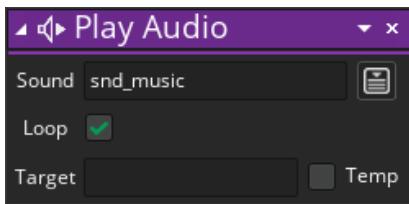


Figure 1-22. This sound action loops the background music.

Sound Effects

Creating and playing sound effects for shooting coins and tools:

1. Create a new sound resource called **snd_tool**.
2. Load the **tool.wav** file from the **Resources** folder of the archive.
3. Close the Sound Properties form.
4. Reopen the Object Properties form for the tool object and select the existing **Collision** event with the laser object.
5. Include a **Play Sound** action in the **Collision** event and select the new sound. Leave the **Loop** option unchecked.
6. Repeat the previous steps to create a sound resource for **error.wav**. Include an action to play it in the coin's collision event with the laser.
7. There are several other sound effects in this folder for you to use and experiment with.

CONGRATULATIONS

Fantastic Work!

You'll find a finished version of this tutorial in the project file `Projects/SpyWire3.yyz` in the archive. You can now return to the video to celebrate what you've achieved and check out a few additional bells and whistles which can be added to the game to give it a more professional feel. Well done – you're amazing! We'll have to make things a bit harder for you next time...



ACKNOWLEDGEMENTS

We would like to thank the whole Spyder™ team for allowing us to use assets from their amazing game in this tutorial. This tutorial would be approximately 15 years less exciting without piggy-backing on all their hard work. If you'd like to see what Spyder's real game developers did with their game then head on over to the Apple Arcade and check it out!



www.spyderthegame.com