V 0.4

SPYDER

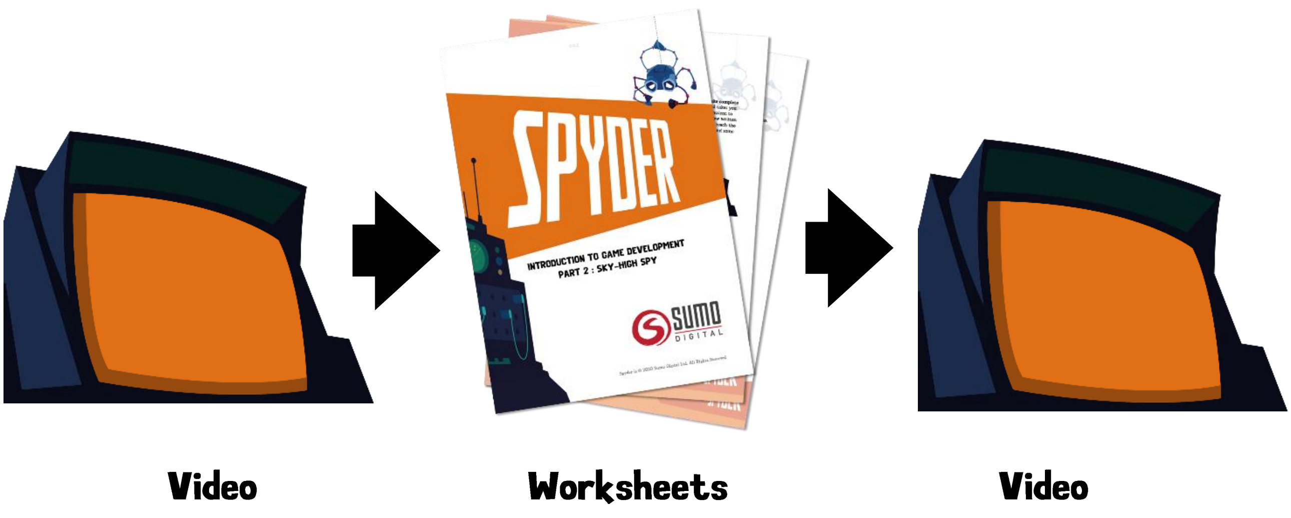
INTRODUCTION TO GAME DEVELOPMENT
PART 2 : SKY-HIGH SPY

SUMO DIGITAL

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 this document takes you step-by-step through making the game. 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.

STILL HAS SIX LEGS

Setting up the sprite resources for a new project:

1. Create a new “Drag and Drop” project in GameMaker Studio 2.

2. Right-click on the **Sprites** section of the **Resources** window and choose **Create Sprite from Image(s).**

2. Choose the file agent8_left_strip7.png (you’ll find it in the Resources folder of the downloaded archive) and name the Sprite spr_agent8_left.

3. Set the **Origin** to -10 x 64. Placing the **Origin** below the sprite in this way will make it rotate around the centre of the moons that Agent 8 will walk on in this game.

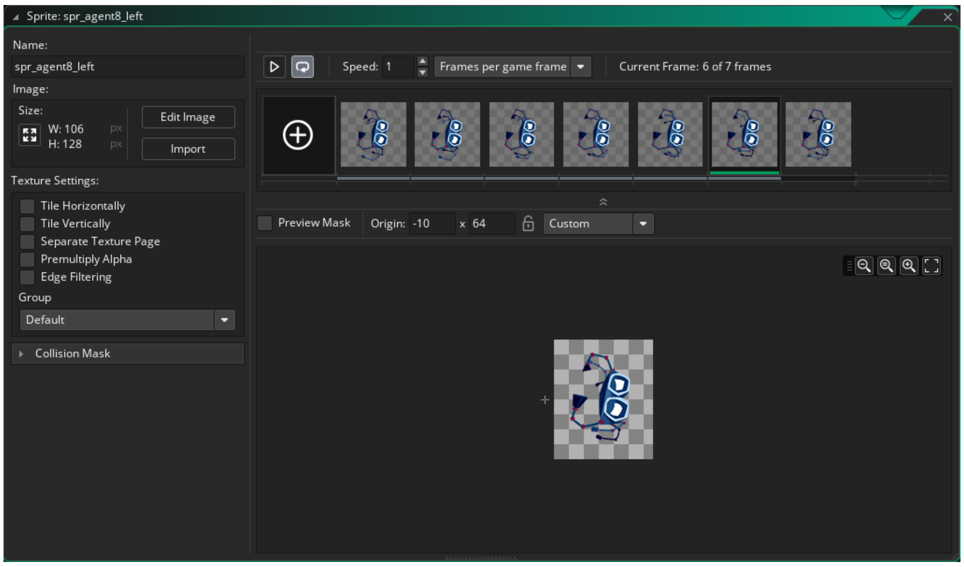


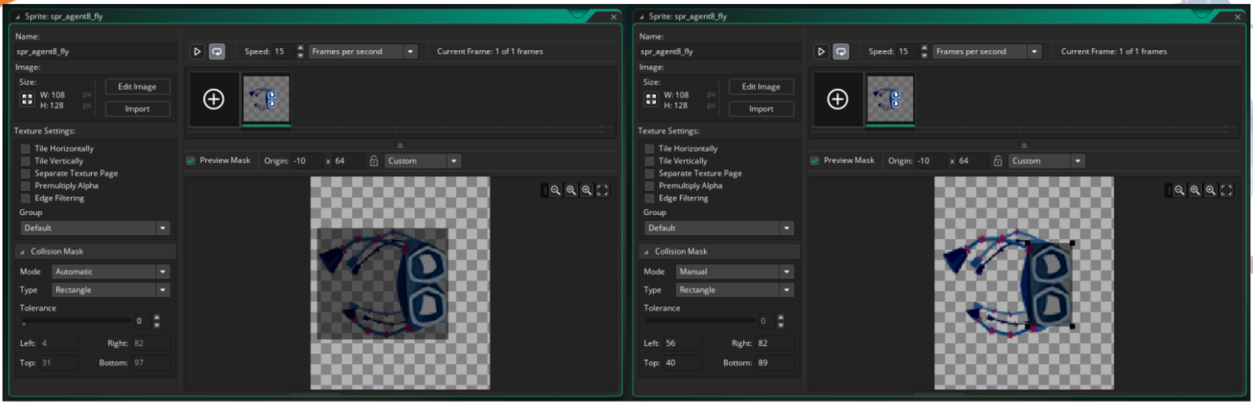
Figure 2-1: The Sprite Properties form after making the left facing Agent 8 sprite

4. Click on the small lock icon to prevent you from accidently changing the origin with a stray mouse click (you can press it again to make the origin editable again).

5. Finally set the sprites’ running speed to 1 Frames per game frame. Your Sprite Properties form should now look like Figure 2-1. Close the form.

6. Create a new Sprite called spr_agent8_right using the file agent8_left_right7.png. Like the left sprite, it should also have an **Origin** of -10 x 64.

7. Create a new Sprite called spr_agent8_fly using the file agent8_fly.png. This sprite needs the same **Origin** settings, but you should also open the **Collision Mask** settings and set **Mode** to Manual. Notice how 4 black “handles” appear on the sprite image which allow you to change the size of the collision area (you can also type values into the **Left**, **Right**, **Top** and **Bottom** values). Reduce the size of the rectangle down so that it is similar to that shown in Figure 2.2 (right). This makes the collision tests more “forgiving” for this sprite and will make the game feel “fairer” to the player.


Sprite: spr_agent8_fly
Name: spr_agent8_fly
Size: W: 108 px H: 128 px
Speed: 15 Frames per second
Current Frame: 1 of 1 frames
Preview Mask ✓ Origin: -10 x 64 Custom
Group: Default
Collision Mask
Mode: Automatic
Type: Rectangle
Tolerance: 0
Left: 4 Right: 82
Top: 31 Bottom: 97

Sprite: spr_agent8_fly
Name: spr_agent8_fly
Size: W: 108 px H: 128 px
Speed: 15 Frames per second
Current Frame: 1 of 1 frames
Preview Mask ✓ Origin: -10 x 64 Custom
Group: Default
Collision Mask
Mode: Manual
Type: Rectangle
Tolerance: 0
Left: 56 Right: 82
Top: 40 Bottom: 89

Figure 2-2: Reduce the size of the automatic collision mask (left) to make it smaller (right)

Creating the remaining game sprites:

1. To create the remaining sprites for the game load the appropriate file and set the **Origins** and **Speed** as indicated below.
2. spr_agent8_dead has an Origin of 94 x 64 (to match the distance of the **Origin** below Agent 8's feet on spr_agent9_fly). Set the **Speed** to 0.2 Frames per game frame.
3. spr_asteroid has an **Origin** of 105 x 75 (to place it at the centre of the moon). Set the **Speed** to 0.2 Frames per game frame.
4. spr_asteroid_pieces need to have their **Origin** set to Middle Centre and their **Speed** set to 0 Frames per game frame (i.e. no animation).
5. spr_meteor has an **Origin** of 162 x 64 (at the centre of the meteorite). Set the **Speed** to 0.2 Frames per game frame.
6. spr_gem has its **Origin** set to Middle Centre and its **Speed** can be left as the default (there is only a single frame so it makes no difference).
7. spr_background has its **Origin** and **Speed** left as the default.

SOUND PLANNING

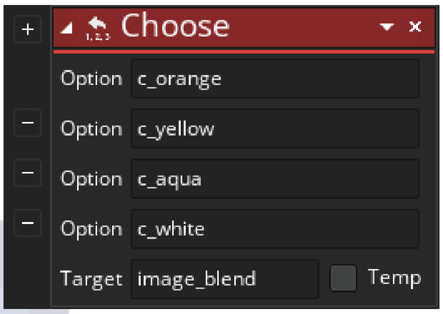
Creating the game's Sound resources:

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 snd_music.wav file from the Resources folder.
3. Close the Sound Properties form.
4. Repeat the process to create sound resources for snd_explode, snd_combust, and snd_reward.

SHOWER OF SPACE ROCKS

Creating an asteroid object:

1. Right-click on the **Objects** section of the **Resources** window and choose **Create Object.** An Object Properties form will appear.
2. In the **Name** field, give the object a name. You should call this one obj_asteroid.
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_asteroid sprite.
4. Add a **Create** event to the list of events for the asteroid.
5. Include a **Set Direction Variable** action (from the Movement section of the Toolbox). Where it says **Direction,** type irandom(360). This will randomly select a whole number between 0 and 359.
6. Include a **Set Speed** action (Movement section). Leave the **Type** set to Direction and set the **Speed** to 5.
7. Include the **Set Instance Rotation** action (Instances section) and set **Angle** to direction. This just makes sure that the direction of movement set in the first action is also the direction in which the sprite is rotated when it is drawn.
8. Include the **Choose** action (Random section) and click on the + icon three times so that four options are displayed in total (see Figure 2.3). Type c_orange in the first **Option,** c_yellow in the second, c_aqua in the third and c_white in the fourth **Option**. Set **Target** to image_blend to randomly assign a different colour to each asteroid.



Option c_orange
Option c_yellow
Option c_aqua
Option c_white
Target image_blend ☐ Temp

Figure 2-3. The Choose action allows you to randomly assign different values to a variable. Clicking on the + icon in the top left corner adds additional options.

9. Add a **Step, Step** event to the asteroid object and include a **Wrap Around Room** action. Set Margin to 128 and select both the **Horizontal** and **Vertical** options. This makes the asteroid reappear back on the opposite side of the room when it goes outside of the room's boundaries.

Creating an 'attached' asteroid object from the asteroid:

1. Right-click on obj_asteroid in the **Resources** and select **Duplicate. Name** the new object obj_asteroid_attached. This object represents the asteroid that Agent 8 is currently attached to. There can be many instances of obj_asteroid, but only one instance of obj_asteroid_attached at any point in time.
2. Right-click on the **Create** event and select **Change Event**. Change it to an **Other, Room Start** event instead. This is necessary because instances of obj_asteroid are going to change into instances of obj_asteroid_attached when Agent 8 lands on them. However, using the **Change Instance** action causes the new object's **Create** event to be called, which here would assign a new colour and direction to an existing asteroid. Putting these actions in a **Room Start** event makes sure that colour and direction are only assigned once at the very start of the level.
3. Include an **If Audio Is Playing** action (Audio section), set **Sound** to snd_music and select the **Not** option (so we're testing whether the music is not already playing).
4. Include a **Play Audio** action (Audio section) and make it depend on the previous action (drag it to the red word Empty). Set **Sound** to snd_music and select the **Loop** option.
5. Now add a **Create** event (one that we DO want to be called when the instance changes), but just include an **Assign Variable** action with **Name** set to depth and **Value** set to -1 (Common section). This brings the attached (most important) asteroid in front of the other asteroids, so it is drawn over the top.

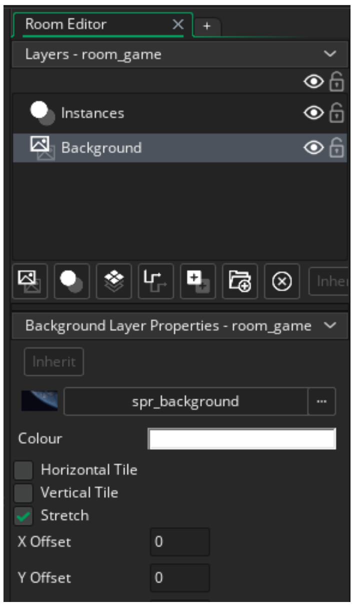
Creating a meteor object from the asteroid:

1. Right-click on obj_asteroid in the **Resources** and select **Duplicate. Name** the new object obj_meteor and assign it the spr_meteor sprite.
2. Edit the **Create** event and delete the **Choose** action (click on the x button in the top, right corner of the action). Change the **Speed** of the meteor to 12. Otherwise the object is the same as the asteroid so you can close the object, as we are done with it.

SPACEOUS ROOM

Editing the game room:

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.
2. Select the **Background** layer in the Room Editor window. This should reveal the Background Layer Properties in the pane below, with a familiar-looing Sprite selection tool. Click on the ellipsis icon and select the background sprite (see Figure 2-4). Also select the **Stretch** option in case you decide to make your room bigger later.


Room Editor

Layers - room_game

Instances

Background

Background Layer Properties - room_game

Inherit

spr_background

Colour

- [ ] Horizontal Tile
- [ ] Vertical Tile
- [x] Stretch

X Offset 0

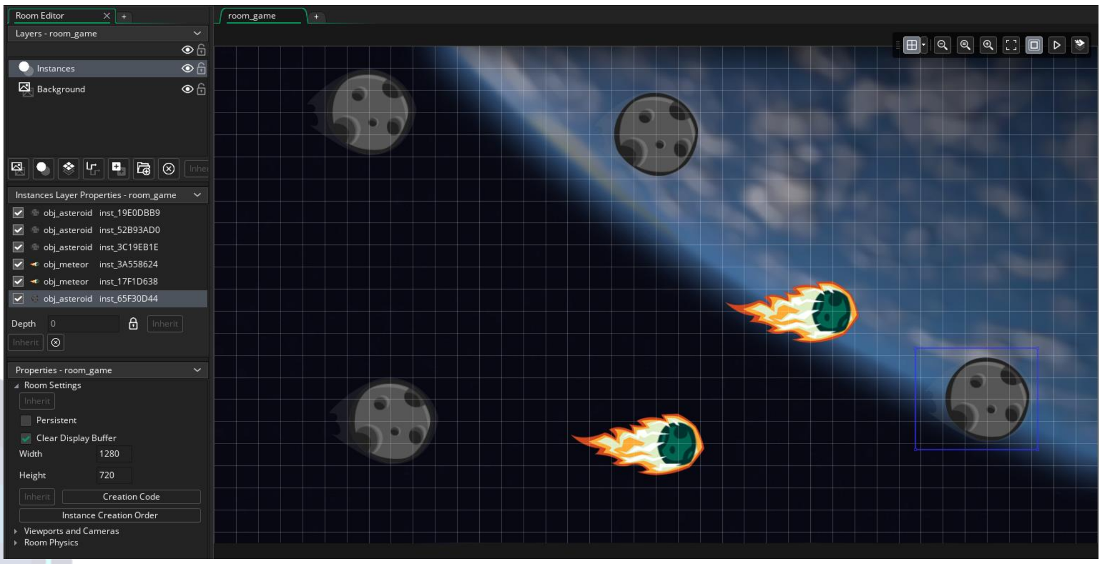
Y Offset 0

Figure 2-4. The Background Layer Properties allows you to assign background sprites.

3. In the bottom left, under Room Settings set the **Width** of the room to 1280 pixels wide and the **Height** to be 720 pixels.

4. Click the **Instances** layer in the top left of the Room Editor window.

5. Click and drag a few instances of obj_asteroid and obj_meteor into the room grid from the Resources. The room should now look something like Figure 2-5.


Room Editor | room_game

Layers - room_game

Instances

Background

Instances Layer Properties - room_game

- [x] obj_asteroid inst_19E0DBB9
- [x] obj_asteroid inst_52B93AD0
- [x] obj_asteroid inst_3C19EB1E
- [x] obj_meteor inst_3A558624
- [x] obj_meteor inst_17F1D638
- [x] obj_asteroid inst_65F30D44

Depth 0

Properties - room_game

Room Settings

- [ ] Persistent
- [x] Clear Display Buffer

Width 1280

Height 720

Creation Code

Instance Creation Order

Viewports and Cameras

Room Physics

Figure 2-5. The Room Editor with asteroids and meteors.

Note *Although we have based all the tutorial games around a room size of 1280 x 720 pixels (standard definition), this game works particularly well at 1920 x 1080 pixels (high definition). If your monitor can cope, then ramp up the size of the room to 1920 x 1080 and you'll get better gameplay as a result of the extra space to maneuver.*

Saving your work and running the game:

1. Choose **Save Project** from the **File** menu (or click the disk icon) and press F5 (or click the play icon) to run the game. 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 SkyHighSpy1.yyz in the Projects directory of the archive.

TAKING AGENT 8 FOR A SPIN

Creating an Agent 8 object:

1. Right-click on **Objects** in the **Resources** window and choose **Create Object.**
2. Give the object a name by entering obj_agent8_walk in the **Name** field.
3. Select the spr_agent8_left sprite from the sprite selection menu.
4. Add a **Key Down** event and select **No Key** from the pop-up menu.
5. Include a **Set Animation Speed** action (Instances section) and set the **Speed** to 0. This stops the walk animations from playing when Agent 8 is standing still.
6. Add a **Key Down** event and select **Left** from the pop-up menu.
7. Include the **Set Direction Variable** action in the **Actions** list and set **Direction** to image_index*3 with the **Relative** option selected. This changes the speed of rotation based on the current frame of animation (slow at the beginning, fast at the end). This gives Agent 8 a scampering kind of movement which fits well for a spider.

Note *The star character (*) is used as the multiplication symbol and can be found on the 8 key on most keyboards (press shift and 8).*

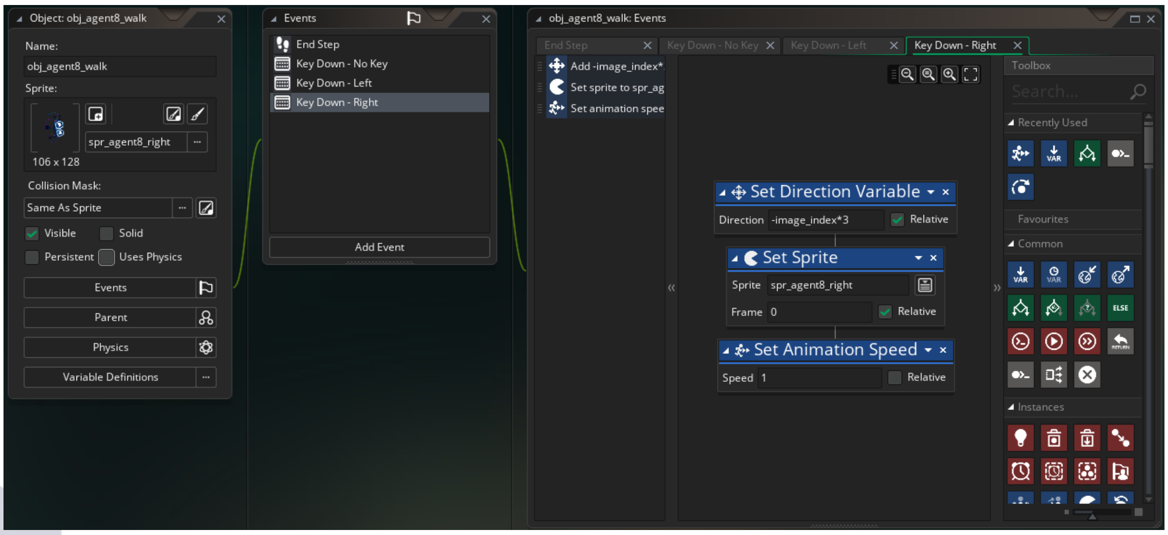
---

8. Include a **Set Sprite** action (Instances section). Set **Sprite** to spr_agent8_left, **Frame** to 0, and select the **Relative** option (i.e. don't change the current animation frame).
9. Include a **Set Animation Speed** action (Instances section) and set the **Speed** to 1.
10. Add a **Key Down, Right** event and include the **Set Direction Variable** action in the **Actions** list and set **Direction** to -image_index*3 (note the minus sign at the start) with the **Relative** option selected.
11. Include a **Set Sprite** action (Instances section). Set **Sprite** to spr_agent8_right, **Frame** to 0, and select the **Relative** option.
12. Include a **Set Animation Speed** action (Instances section) and set the **Speed** to 1.

11. Add a **Step, End Step** action and include a **Jump to Point** action (Movement section). Set **X** to obj_asteroid_attached.x and **Y** to obj_asteroid_attached.y. This will move the origin of the Agent 8 object to the origin of the (moving) asteroid so that the player does indeed appear to be 'attached' to that asteroid in the game.

12. Include a **Set Instance Rotation** action (Instances section) and set the **Angle** to direction. This makes sure that the direction of movement set in the keyboard events is also the direction in which the sprite is rotated when it is drawn.

The Agent 8 Walk object should now look like Figure 2-6. Try dragging a single instance of both obj_agent8_walk and obj_asteroid_attached into your room. Check that you follow the attached asteroid and can walk around it okay before continuing. You can also compare your version to SkyHighSpy2.yyz in the Projects directory of the archive.


Object: obj_agent8_walk
Name: obj_agent8_walk
Sprite: spr_agent8_right
106 x 128
Collision Mask: Same As Sprite
Visible ✓ Solid
Persistent Uses Physics

Events
End Step
Key Down - No Key
Key Down - Left
Key Down - Right

obj_agent8_walk: Events
End Step | Key Down - No Key | Key Down - Left | Key Down - Right

Add -image_index*
Set sprite to spr_ag
Set animation spee

Set Direction Variable
Direction -image_index*3 ✓ Relative

Set Sprite
Sprite spr_agent8_right
Frame 0 ✓ Relative

Set Animation Speed
Speed 1 Relative

Figure 2-6. The basic Agent 8 object with keyboard events.

YOU'RE FIRED!

Creating the dead Agent 8 object:

1. Create a new object called obj_agent8_dead and assign it the spr_agent8_dead sprite.
2. Add a **Create** event and include a **Set Animation Speed** action (Instances section). Set **Speed** to 1. Note that this just plays the sprite animation at its normal full speed (i.e. 0.2 **Frames per game frame** as previously set in the sprite settings).
3. Include a **Play Audio** action (Audio section) and set **Sound** to snd_combust.
4. Add an **Other, Outside Room** event and include a **Restart Room** action (Rooms section).

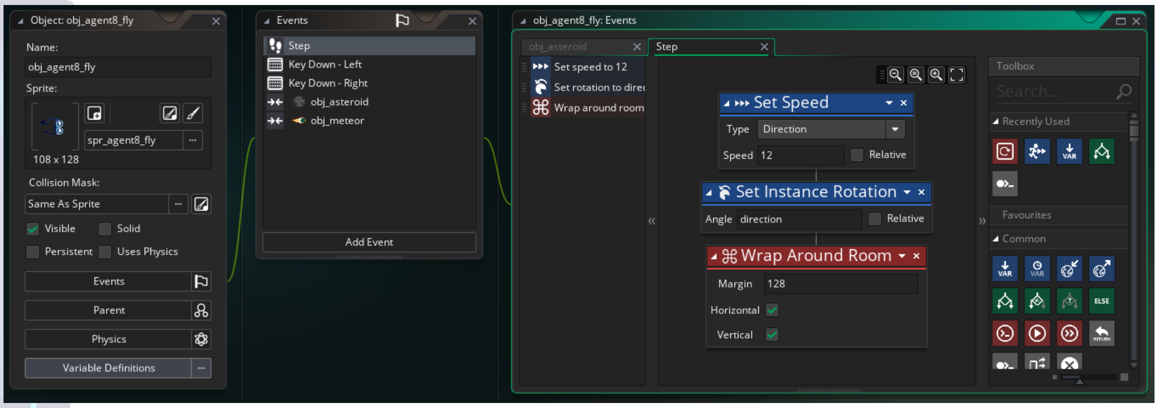
Creating the flying Agent 8 object:

1. Create a new object called obj_agent8_fly and assign it the spr_agent8_fly sprite.
2. Add a **Key Down, Left** event and include a **Set Direction Variable** action (Movement section). Set Direction to 1 and select the **Relative** option. The player will only be able to turn very slowly when flying through the air.
3. Add a **Key Down, Right** event and include another **Set Direction Variable** action. Set Direction to -1 and select the **Relative** option.
4. Add a **Step, Step** event and include a **Set Speed** action (Movement section). Set **Type** to Direction (which is changing as Agent 8 turns) and **Speed** to 12 (like the meteor).
5. Include a **Set Instance Rotation** (Instances section) and set **Angle** to direction.

Note *GameMaker's UI often uses uppercase letters (e.g. for the 'D' at the start of Direction in the **Set Direction Variable** dropdown menu), but all of GameMaker's built-in variables use lowercase letters when they have to be typed in using the keyboard (e.g. the direction variable). If you use uppercase, then it will not recognise it as the correct variable.*

---

6. Include a **Wrap Around Room** action (Movement section). Set **Margin** to 128 and select both the **Horizontal** and **Vertical** options. This will make Agent 8 jump to the opposite side of the room when he flies outside its boundaries. The margin is set to wider than the size of the largest wrapping sprite to prevent instances appearing to pop into existence in their new location.
7. Add a **Collision** event with obj_meteor. Include a **Change Instance** action (Instances section) and set **Object** to obj_agent8_dead.
8. Add a **Collision** event with obj_asteroid. Include a **Change Instance** action which **Applies To** the Other object (the asteroid) and set **Object** to obj_asteroid_attached.
9. Include a second **Change Instance** action and set **Object** to obj_agent8_walk. There's no need to change **Applies To** as it will default to Self (the obj_agent8_fly instance). The obj_agent8_fly properties form should now look like Figure 2-7.


Object: obj_agent8_fly
Name: obj_agent8_fly
Sprite: spr_agent8_fly
108 x 128
Collision Mask: Same As Sprite
Visible [✓] Solid [ ]
Persistent [ ] Uses Physics [ ]

Events:
Step
Key Down - Left
Key Down - Right
obj_asteroid
obj_meteor

obj_agent8_fly: Events — Step
Set speed to 12
Set rotation to direc
Wrap around room

Set Speed
Type: Direction
Speed: 12 Relative [ ]

Set Instance Rotation
Angle: direction Relative [ ]

Wrap Around Room
Margin: 128
Horizontal [✓]
Vertical [✓]

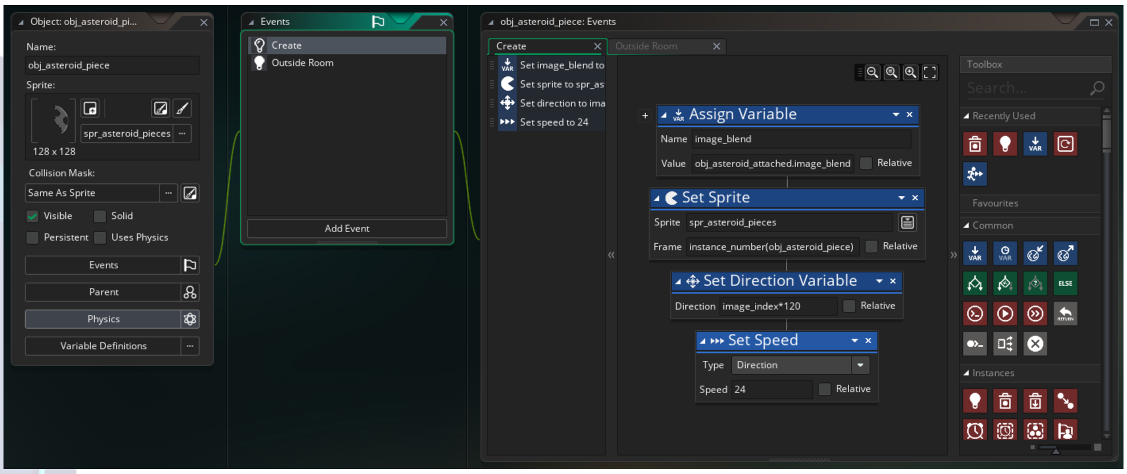
Figure 2-7. The flying Agent 9 object should look like this.

Creating the gem object:

1. Create a new object called obj_gem and give it the gem sprite.
2. Add a **Create** event and include an **Assign Variable** action (Common section) with **Name** set to depth and **Value** set to 10. This puts the gem behind all the asteroids.
3. Add an **Other, Outside Room** event and include an **Jump to Point** action (Movement section) with **X** set to irandom(room_width) and **Y** set to irandom(room_height). This makes sure the gem can't get left outside of the room where it can't be seen, as it will randomly teleport inside again if this should happen.

Creating the asteroid piece object:

1. Create a new object called obj_asteroid_piece and give it the spr_asteroid_pieces sprite.
2. Add a **Create** event and include an **Assign Variable** action (Common section) with **Name** set to image_blend and **Value** set to obj_asteroid_attached.image_blend. This copies the colour settings from the asteroid Agent 8 is currently walking on.
3. Include a **Set Sprite** action with **Sprite** set to spr_asteroid_pieces and **Frame** set to instance_number(obj_asteroid_piece). This will make sure that each of the three asteroid pieces are assigned a different sequential image from the sprite.
4. Include a **Set Direction Variable** action with **Direction** set to image_index*120. Note that image_index is the name of the variable that GameMaker uses to store the animation frame (the same one set using **Frame** in a **Set Sprite** action). This will align the object's direction of movement with the orientation implied by the images.
5. Finally include a **Set Speed** action (Movement section). Leave the **Type** set to Direction and set the **Speed** to 24. The properties form should now look like Figure 2-8.
6. Now add an **Other, Outside Room** event and include a **Destroy Instance** action.



Object: obj_asteroid_pi...
Name: obj_asteroid_piece
Sprite: spr_asteroid_pieces
128 x 128
Collision Mask: Same As Sprite
☑ Visible ☐ Solid
☐ Persistent ☐ Uses Physics

Events: Create, Outside Room

obj_asteroid_piece: Events — Create
- Set image_blend to
- Set sprite to spr_as
- Set direction to ima
- Set speed to 24

Assign Variable
Name: image_blend
Value: obj_asteroid_attached.image_blend ☐ Relative

Set Sprite
Sprite: spr_asteroid_pieces
Frame: instance_number(obj_asteroid_piece) ☐ Relative

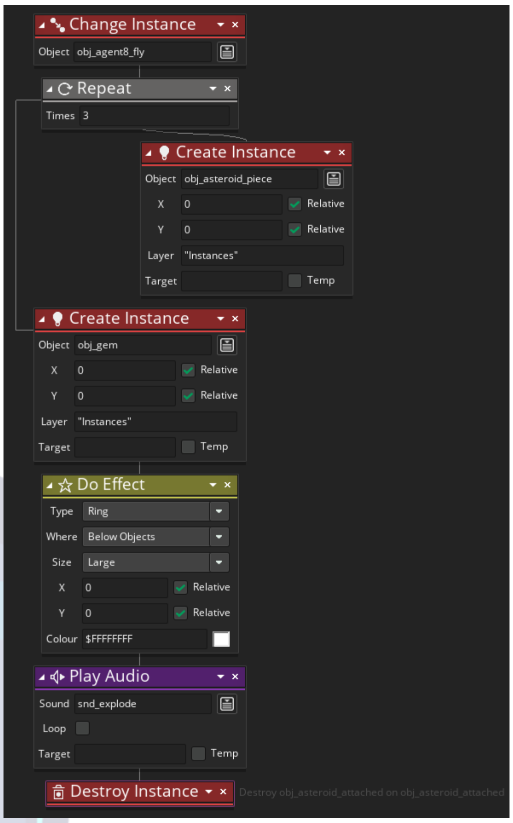
Set Direction Variable
Direction: image_index*120 ☐ Relative

Set Speed
Type: Direction
Speed: 24 ☐ Relative

Figure 2-8. The Asteroid Pieces object.

Editing the walking agent8 object:

1. Reopen the obj_agent8_walk object (not the sprite) in the resource list.
2. Add a **Key Pressed, Space** event and include a **Change Instance** action (Instances section) and set **Object** to obj_agent8_fly.
3. Include a **Repeat** event and set **Times** to 3 (there are three asteroid pieces).
4. Now attach a **Create Instance** action to the **Repeat** action by dragging the new action onto the red word Empty. Set **Object** to obj_asteroid_piece and select both **Relative** options, so that the three pieces are all created relative to the asteroid's position.
5. Include another **Create Instance** action which doesn't depend on the **Repeat** action (i.e. not connected to the previous **Create Instance** action). Set **Object** to obj_gem and select both **Relative** options again.
6. Include a **Do Effect** action with **Type** set to Ring, **Where** set to Below Objects, **Size** set to Large and both **Relative** options selected.
7. Include a **Play Audio** action with **Sound** set to snd_explode.
8. Include a **Destroy Instance** action which **Applies To** obj_asteroid_attached. The actions for this event should now look like Figure 2-9.



```
Change Instance
  Object  obj_agent8_fly
Repeat
  Times  3
    Create Instance
      Object  obj_asteroid_piece
      X  0        [✓] Relative
      Y  0        [✓] Relative
      Layer  "Instances"
      Target      [ ] Temp
Create Instance
  Object  obj_gem
  X  0        [✓] Relative
  Y  0        [✓] Relative
  Layer  "Instances"
  Target      [ ] Temp
Do Effect
  Type  Ring
  Where  Below Objects
  Size  Large
  X  0        [✓] Relative
  Y  0        [✓] Relative
  Colour  $FFFFFFFF
Play Audio
  Sound  snd_explode
  Loop  [ ]
  Target      [ ] Temp
Destroy Instance   Destroy obj_asteroid_attached on obj_asteroid_attached
```

Figure 2-9. The actions for the Key Press, Space event of the walking Agent 8 object. Note that only the first Create Instance action is dependant on the Repeat action.

9. Now add a **Create** event. Note that this object hasn't needed one up to now, but we want to influence Agent 8's behaviour when he collides with an asteroid. At the moment he often appears to teleport to the opposite side of the asteroid because that's the direction he was travelling in. We can fix that as follows.

10. Include an **Assign Variable** action (Common section) with **Name** set to depth and **Value** set to -2. This moves Agent 8 in front of the asteroid.

11. Include a **Set Point Direction** action (Movement section) with **X** set to obj_asteroid_attached.x and **Y** set to obj_asteroid_attached.y. This sets Agent 8's direction to point him at the centre of the asteroid, as he won't always collide head-on.

12. Include a **Set Direction Variable** action with **Direction** set to 180 and select the **Relative** option. This will rotate him 180 degrees so that he lands on his feet with his underside facing the asteroid.

PARENTAL INFLUENCE

Creating an Agent 8 parent object:

1. Create a new object called obj_agent8 and assign it any of the agent8 sprites (it doesn't really matter which as it is never used – we'll explain why in the video).
2. Add a **Collision** event with obj_gem and include a **Set Score** action with **Score** set to 100 and the **Relative** option selected.
3. Include the **Destroy Instance** action with **Appies To** set to Other (the gem).
4. Include a **Play Audio** action with Sound set to snd_reward.
5. Add a **Draw, Draw** event and include a **Draw Instance Score** action (Drawing section) with **Caption** set to "Score:", **X** set to room_width/2 and **Y** set to 50.
6. Include a **Draw Self** action (Drawing section).
7. Add a **Step, Begin Step** event and include a **Get Instance Count** action (Instances section) with **Object** set to obj_asteroid and **Target** set to asteroids. This counts the number of remaining asteroids and puts the answer in a new variable called asteroids.
8. Include a second **Get Instance Count** action (Instances section) with **Object** set to obj_asteroid_attached and **Target** set to attached.
9. Include a third **Get Instance Count** action (Instances section) with **Object** set to obj_gem and **Target** set to gems.
10. Include an **If Variable** action (Common section) with **Variable** set to asteroids+attached+gems, **Is** set to Equal and **Value** set to 0. This now checks to see if all the asteroids and gems have been collected or destroyed.

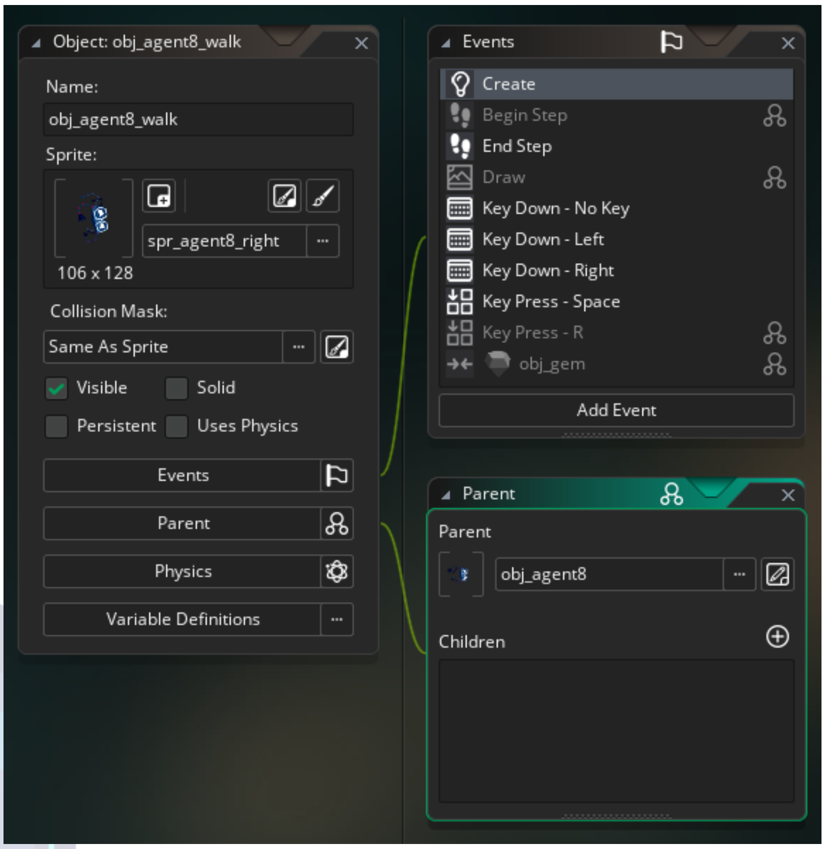
11. Attach a **Destroy Instance** action to the **If Variable** action by dragging the new action onto the red word Empty. Set **Applies To** to obj_meteor to destroy all the remaining meteors and provide the player with some time to gloat over their score.

12. Add a **Key Press, R** event and include a **Restart Room** action (Rooms section) so that the player can reset the game when they get board of gloating.

13. Finally a little bit of magic. Reopen obj_agent8_walk and set its parent to be obj_agent8 (click on the icon with three circles, next to where it says Parent). Note how the events and actions from obj_agent8 are added to the existing object, but appear greyed-out. The greyed-out events are the ones which have been "inherited" from the parent object (see Figure 2-10). We'll learn more about this in the next tutorial.

14. Reopen obj_agent8_fly and set its parent to be obj_agent8.

15. Reopen obj_agent8_dead and set its parent to be obj_agent8. Consider how we've added multiple behaviours to three objects using just one set of actions in obj_agent8. Feels good doesn't it? Best of all if we want to change the behaviour (for example increasing the number of points for a gem) then we only need to change it in one place.


Object: obj_agent8_walk
Name: obj_agent8_walk
Sprite: spr_agent8_right
106 x 128
Collision Mask: Same As Sprite
Visible [✓]
Solid [ ]
Persistent [ ]
Uses Physics [ ]

Events
- Create
- Begin Step
- End Step
- Draw
- Key Down - No Key
- Key Down - Left
- Key Down - Right
- Key Press - Space
- Key Press - R
- obj_gem

Parent
Parent: obj_agent8
Children

Figure 2-10. The Agent8 walk object showing its own events and those inherited from its parents (in grey).

CONGRATULATIONS

Fantastic Work!

You'll find a finished version of this tutorial in the project file Projects/SkyHighSpy3.yyz in the archive. As before, you can now return to the video to celebrate what you've achieved and check out a few features which you might want to add to the game. You're learning fast, so stick with it and you'll have earned your game development stripes in no 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