USING VARIABLES IN FISHPOD

Our Fishpod object has a number of different states that affect the way it behaves: **standing**, **walking**, **jumping**, **falling**, and **dying**. Fishpod will appear differently in each of these states but its behaviour will be different in each of these states as well. Each of Fishpod's five states could be described as follows:

STANDING:	Stands motionless on a platform's surface.
WALKING:	Walks horizontally across a platform's surface.
JUMPING:	Jumps diagonally under momentum and the influence of gravity.
FALLING:	Falls under the influence of gravity.
DYING:	Dies under the influence of gravity and without collisions.

It's possible to represent different states of an object by creating separate objects for each state and using the **Change Instance** action to switch between them when required. **Change Instance** turns an instance of one type of object into another while keeping the same position, direction, and speed or any other individual properties of the instance.

What complicates things here is the different directions the character can be facing. Although we have five possible states that Fishpod will assume, there will actually be ten states in this game because each of the five states can either be facing **left** or **right**. We could easily go ahead and create different left and right objects for each state, but this would seem to complicate things. What we will do instead is use a **variable** to keep track of which direction the character is facing, using a value of 1 for facing left and a value of 2 for facing right.

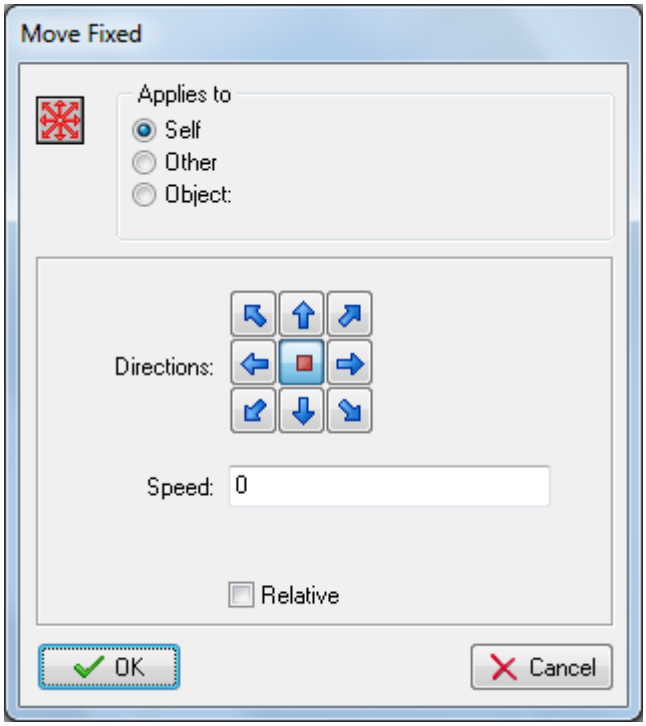
CREATING THE STANDING STATE OBJECT

1. Create a new object called **obj_pod_standing** and set the sprite to either **spr_pod_stand_left** or **spr_pod_stand_right**. It doesn't matter which sprite you assign because we will be selecting the correct sprite manually in the **Create** event.

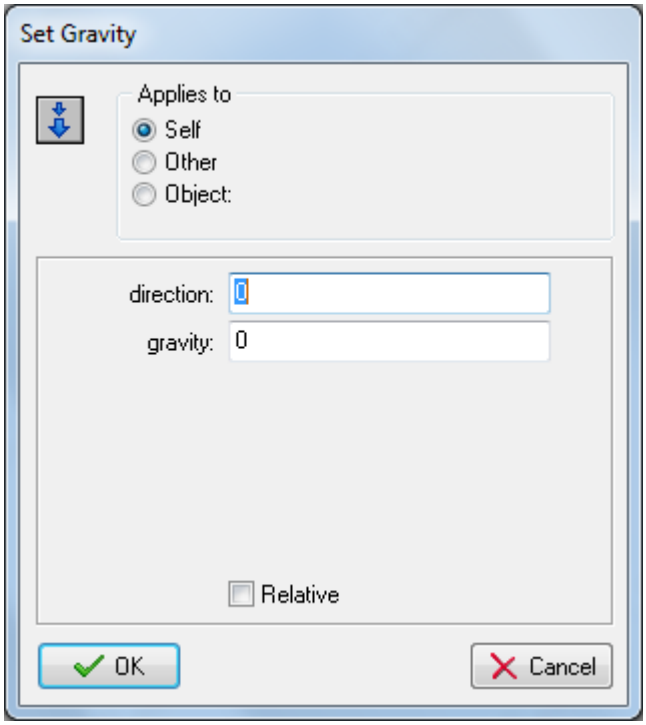
2. Set its **Depth** value to **-1** so that it appears in front of other objects.

3. Click on the **Add Event** button and select the **Create** event.

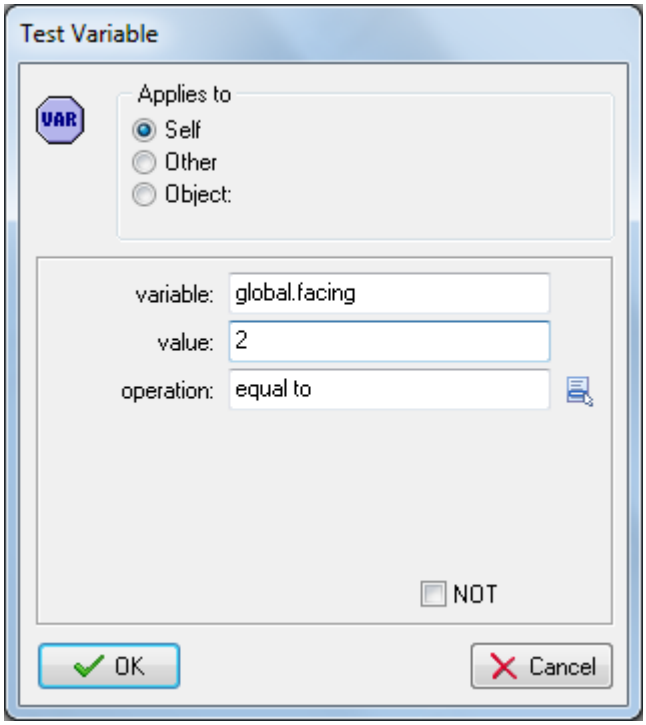
4. Add a **Move Fixed** action, select the middle direction button to indicate no direction of movement and leave the **Speed** set to **0**. This ensures that the character stops moving when it enters the standing state.



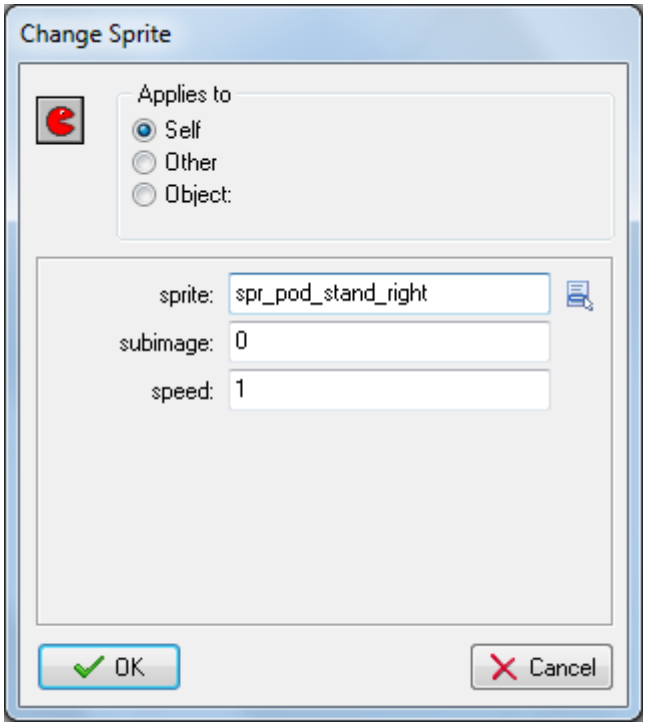
5. Add the **Set Gravity** action and leave **Direction** and **Gravity** set to 0. This ensures that gravity is turned off for the character when it enters the standing state; otherwise, gravity would start it moving again.



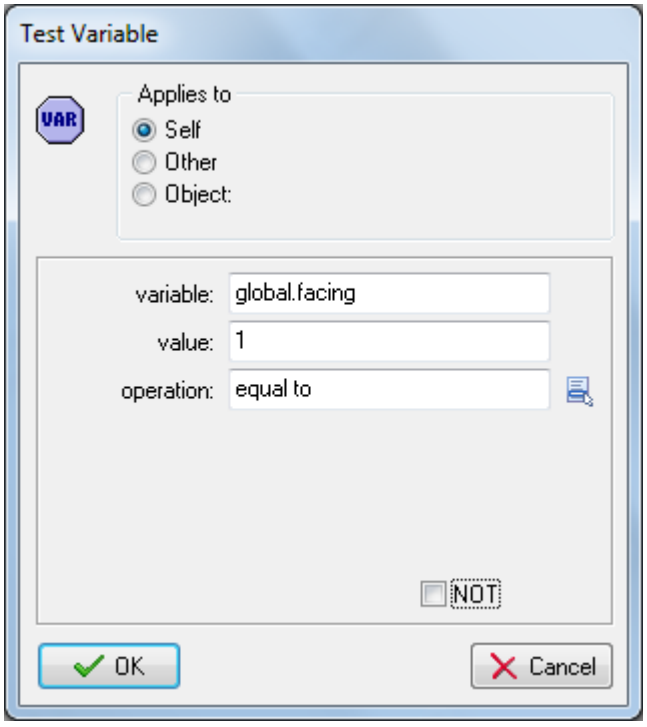
6. Add the **Test Variable** action. We're going to use a variable called **global.facing** to record whether the character is facing left or right, and use a numerical value. Type **global.facing** into **Variable**, 2 into **Value**, and leave **Operation** set to **equal to**. This checks to see whether the character is facing right and only performs the next action if this is true.



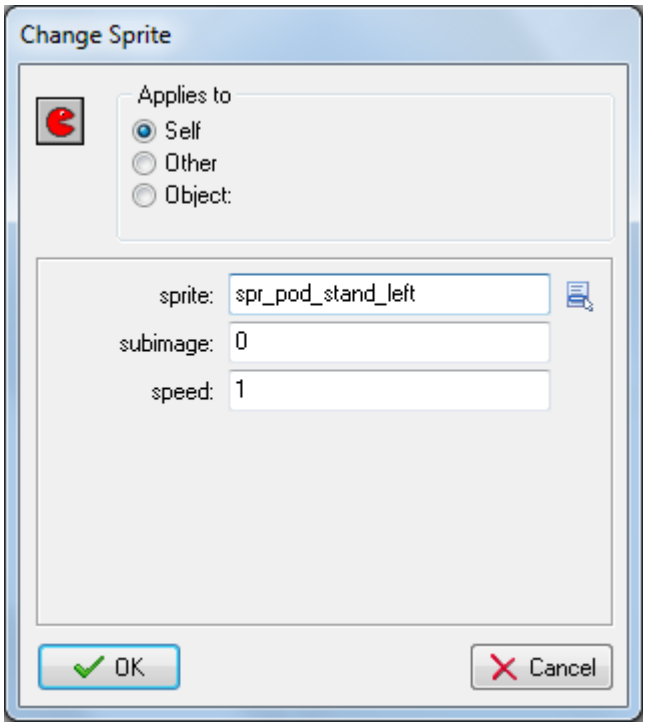
7. Add a **Change Sprite** action, select the **spr_pod_stand_right** sprite and leave the other settings as they are.



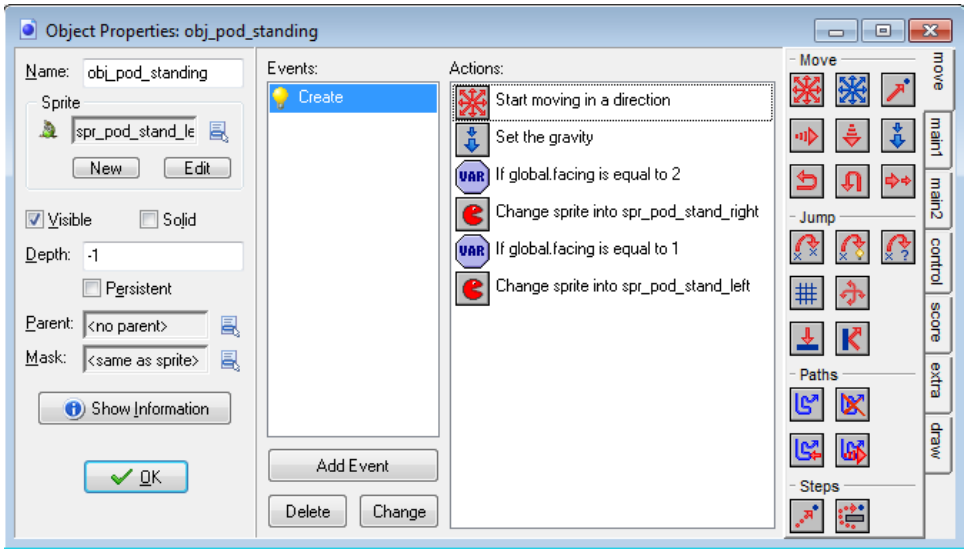
8. Add another **Test Variable** action, type **global.facing** into **Variable**, 1 into **Value**, and leave **Operation** set to **equal to**. This checks to see whether the character is facing left and only performs the next action if this is true.



9. Add a **Change Sprite** action, select the **spr_pod_stand_left** sprite and leave the other settings as they are.



You should now have a list of six actions for your **Create** event.



CREATING THE REMAINING STATE OBJECTS

1. Create a new object called **obj_pod_walking** and set its sprite to be either **spr_pod_walk_left** or **spr_pod_walk_right**, as we will be selecting the correct sprite manually in the **Create** event.

2. Set its **Depth** value to **-1** so that it appears in front of other objects.

3. Create a new object called **obj_pod_jumping,** set its sprite to be either **spr_pod_jump_left** or **spr_pod_jump_right** and set its **Depth** to **-1**.

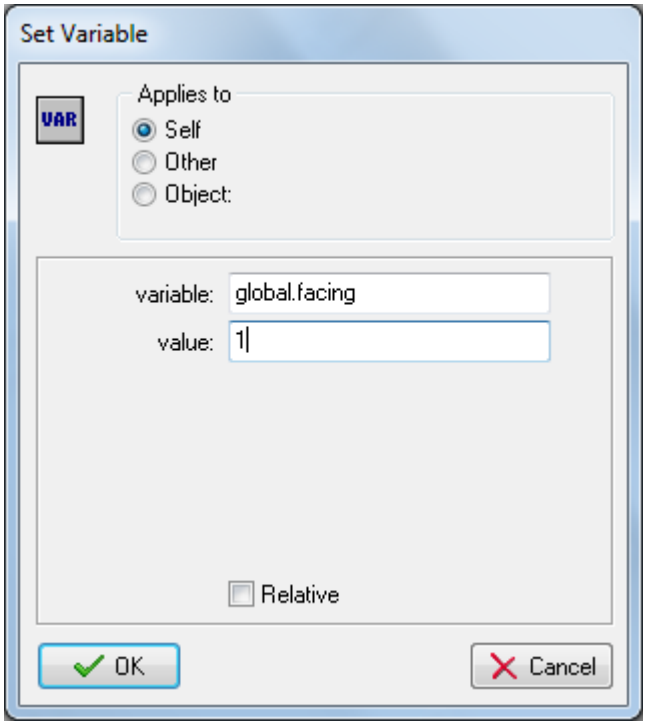
4. Create a new object called **obj_pod_falling**, set its sprite to be either **spr_pod_fall_left** or **spr_pod_fall_right** and set its **Depth** to **-1**.

5. Create a new object called **obj_pod_dying**, set its sprite to be either **spr_pod_jump_left** or **spr_pod_jump_right**, and set its **Depth** to **-1**. The jump sprite shows Fishpod spinning, so we will also use this as a dying animation played as he falls from the screen.

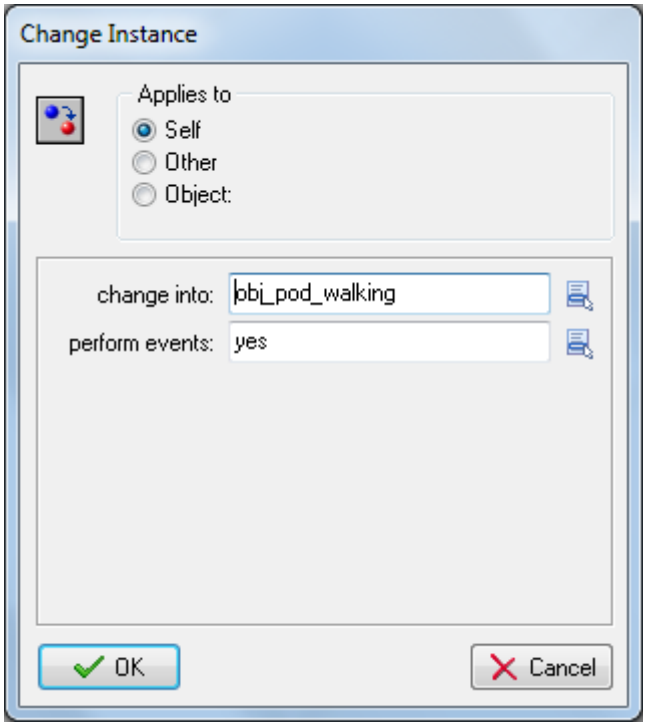
ADDING KEY EVENTS TO THE STANDING STATE OBJECT

1. Double-click the **obj_pod_standing** object, click on **Add Event** and select the **Keyboard <Left> event**. We are going to use a **Keyboard** event because we want Fishpod to keep walking when the player holds down the arrow keys.

2. Add a **Set Variable** action, type **global.facing** into **Variable** and **1** into **Value**. We will use this to make sure that Fishpod faces left when you press the left arrow key.



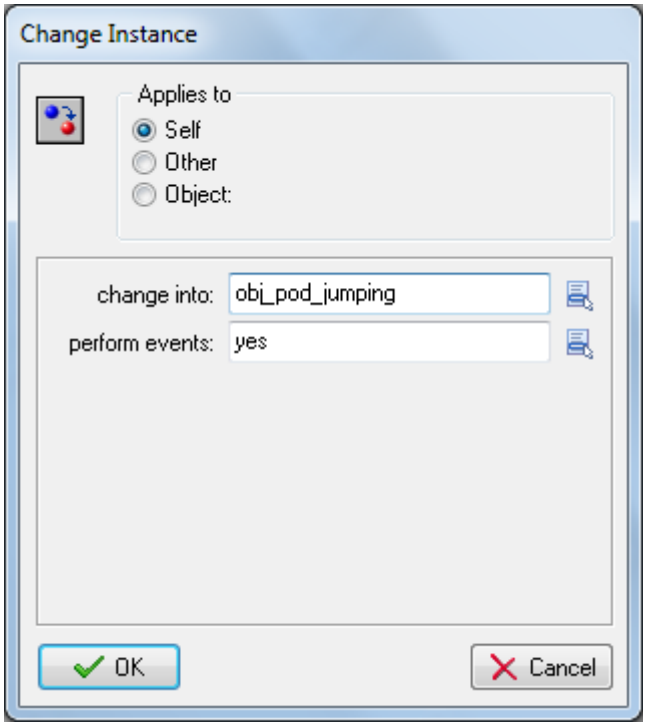
3. Add a **Change Instance** action, select **obj_pod_walking** from the **Change Into** menu and **yes** for **Perform Events**. This last setting means that the **Create** event of **obj_pod_walking** will be called as part of the change.



4. Repeat steps 2–3 using a **Keyboard <Right>** event and setting **global.facing** to **2** instead.

5. Click on the **Add Event** button and select a **Key Press <Space>** event. We are using a **Key Press** event here because we only want Fishpod to jump once for each press of the space bar.

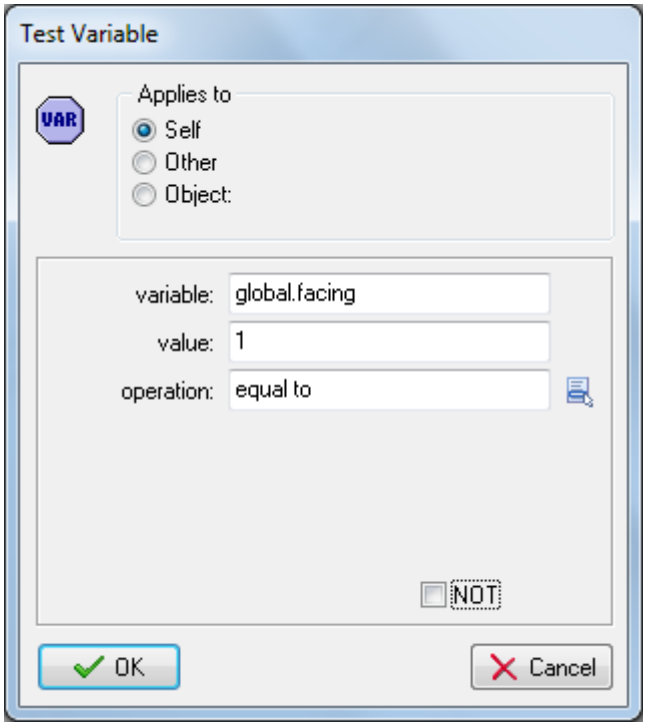
6. Add a **Change Instance** action, select **obj_pod_jumping** from the **Change Into** menu and **yes** for **Perform Events** so that the Create event of **obj_pod_jumping** is called as part of the change.



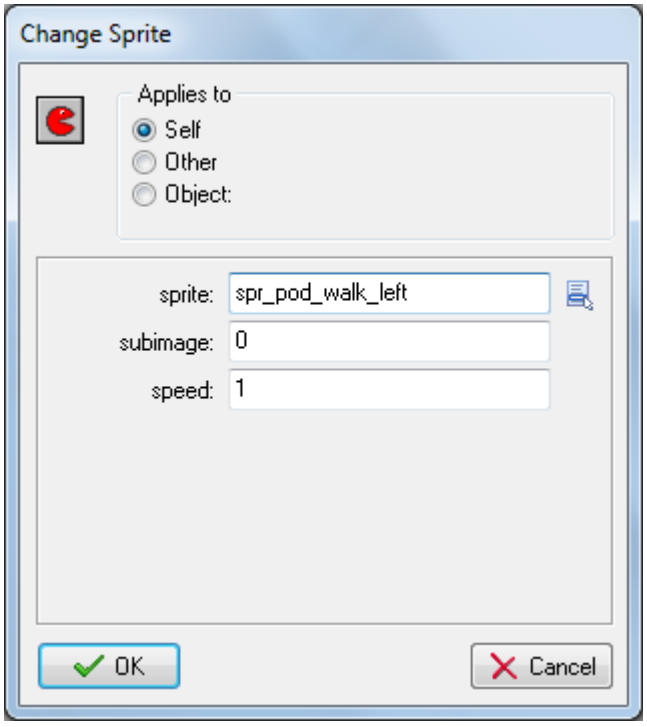
ADDING THE CREATE EVENT FOR THE WALKING STATE OBJECT

1. Double-click the **obj_pod_walking** object, click on the **Add Event** button and select the **Create** event.

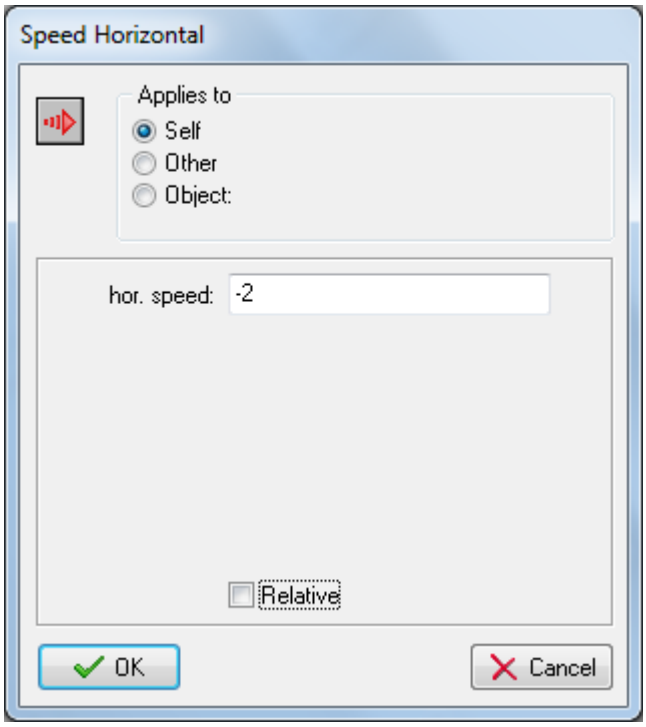
2. Add a **Test Variable** action, type **global.facing** into **Variable**, **1** into **Value**, and leave **Operation** set to **equal to**. This checks to see whether the character is facing left and only performs the next actions if this is true.



3. Add a **Start Block** action, add a **Change Sprite** action, and choose the **spr_pod_walk_left** sprite.

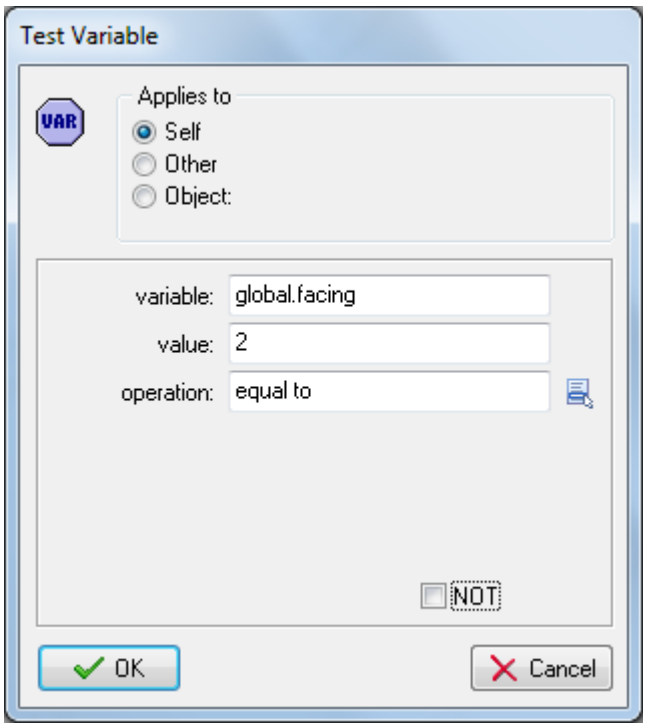


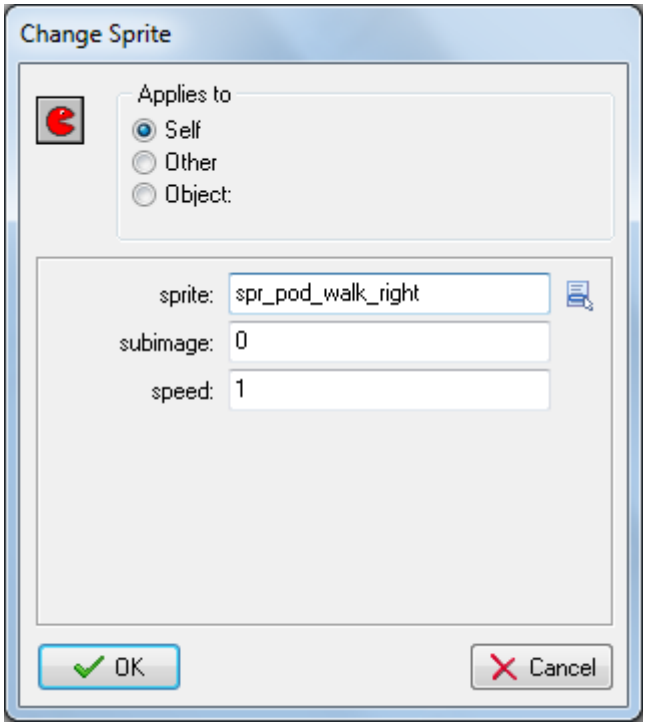
4. Add a **Speed Horizontal** action and set the **Hor. Speed** to **-2**.

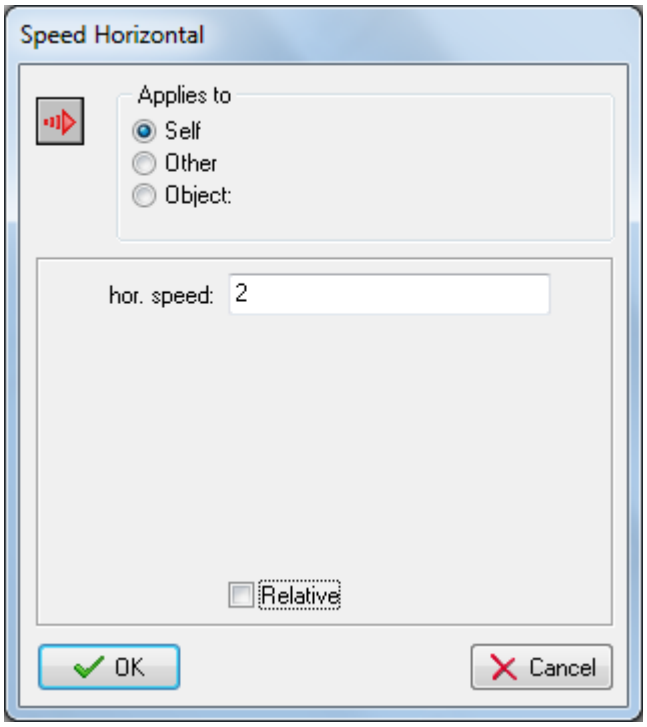


5. Add an **End Block** action to indicate the end of a block of actions.

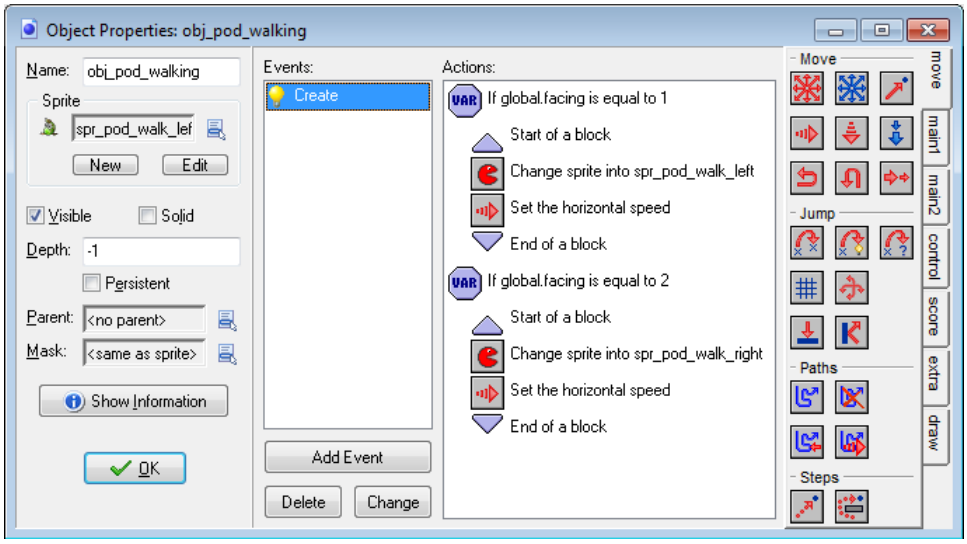
6. Repeat steps 2-4 testing for whether the object is facing right, changing the sprite to **spr_pod_walk_right**, and setting the **Hor. Speed** to **2**.







You should now have a set of ten actions for the **Create** event that looks like this:



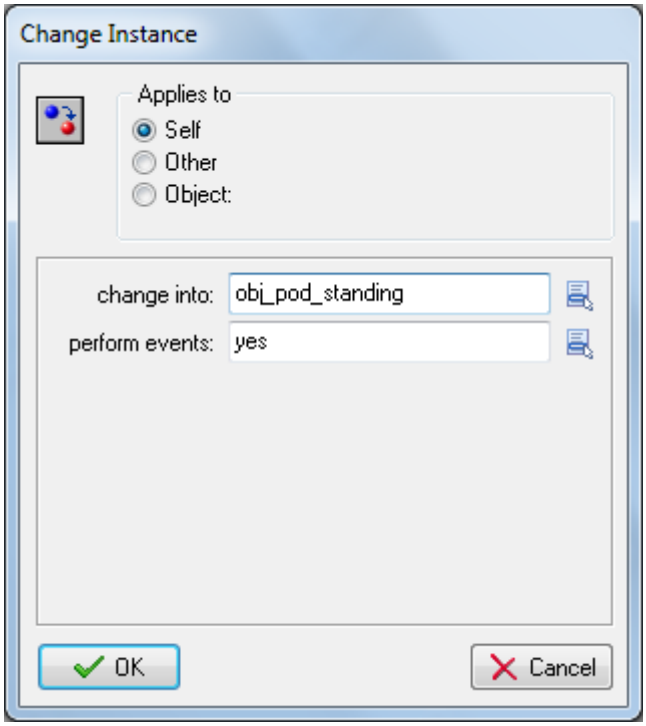
The next step is to handle the transition from the walking state back to the standing state. The animations for Fishpod have been designed to depict a single hop, which would look odd if it was interrupted mid-flight, so there are going to be no Keyboard events for the walking state. Instead, Fishpod will return back to the standing state at the end of each walking animation loop, so that the player can control him again between hops.

The **Animation End** event is triggered when the current sprite animation reaches its final frame, so this is the event we will use for the state transition.

ADDING THE ANIMATION END EVENT TO THE WALKING STATE OBJECT

1. Click on the **Add Event** button and select the **Other > Animation End** event.

2. Add a **Change Instance** action that changes into **obj_pod_standing** and select **yes** for **Perform Events**.



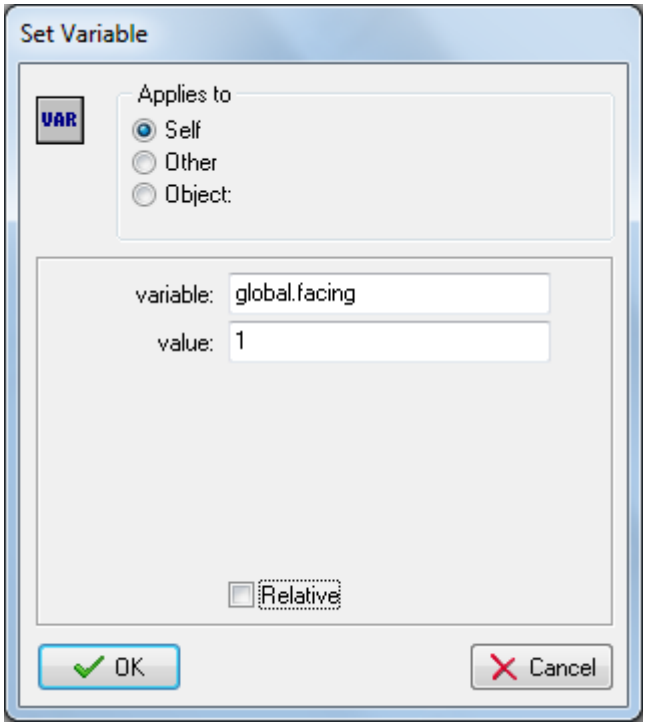
CREATING A NEW STARTING OBJECT AND ITS CREATE EVENT

We're now very close to being able to test our character events for the first time, but before we can do that, we're actually going to have to create another object for Fishpod. This isn't another state object but an object that sets up Fishpod's initial properties in the level. Normally, we would use the **Create** event of one of the state objects (probably **obj_pod_standing**) to provide any starting values (for things such as health and lives, for example). However, we are already using the **Create** events to set up the behavior of each state. If we put starting values in these **Create** events as well, then they would get reset between states (so you would go back to full health while walking, for instance). This is clearly not desirable, so instead we will create a new **obj_pod** object that sets up the initial properties for the character before changing itself into **obj_pod_standing**. We don't want to give our character health at this stage, but we do need to initialize the facing variable for the character. This will be done in the **Create** event of **obj_pod**, as it will only ever be created once at the very start of the level. It is essential that we initialize this value somewhere; otherwise, Game Maker will quite rightly give an error the first time we try and check what the value of facing is.

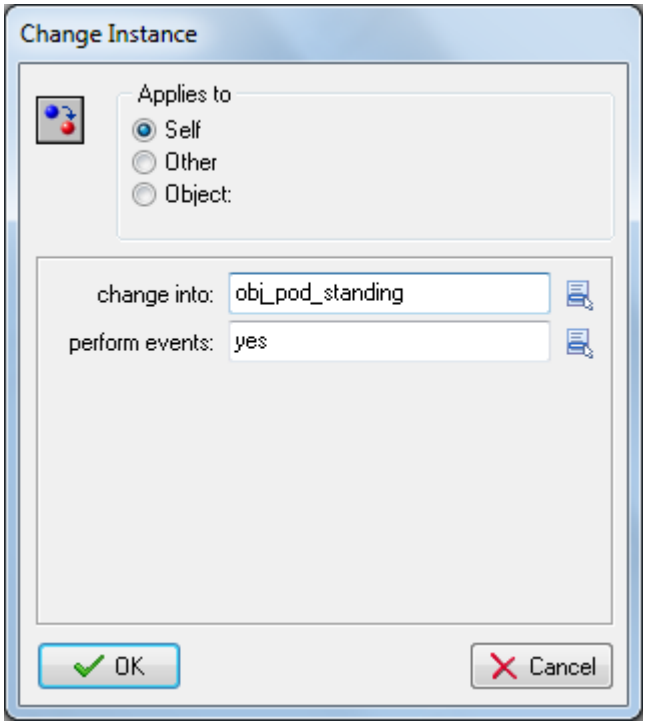
1. Create a new object called **obj_pod** and set its sprite to any one of the Fishpod sprites.

2. Click on the **Add Event** button and select the **Create** event.

3. Add a **Set Variable** action, type **global.facing** into **Variable** and **1** into **Value**. This sets the initial facing direction of Fishpod to the left.



4. Add a **Change Instance** action, select **obj_pod_standing** from the **Change Into** menu and **yes** for **Perform Events**. This will put the object into the standing state and it will never return to the starting object again.



We can now reopen our test room and place an instance of **obj_pod** somewhere within the level. Run the game and check that you can move Fishpod left and right using the arrow keys on the keyboard.

Note that the character will currently walk on thin air and through platforms as we've not handled those things yet. Pressing the space bar will also cause Fishpod to spin endlessly and you'll have to quit the game (this is because he is stuck in the jumping state and there is no way out of that state yet).

SOURCE: Habgood, Jacob, Nielsen, Nana, Rijks, Martin and Kevin Crossley. *The Game Maker's Companion: Game Development: The Journey Continues.* New York: Apress, 2010. Print.