

CREATING HAZARDS AND GOALS TO FISHPOD

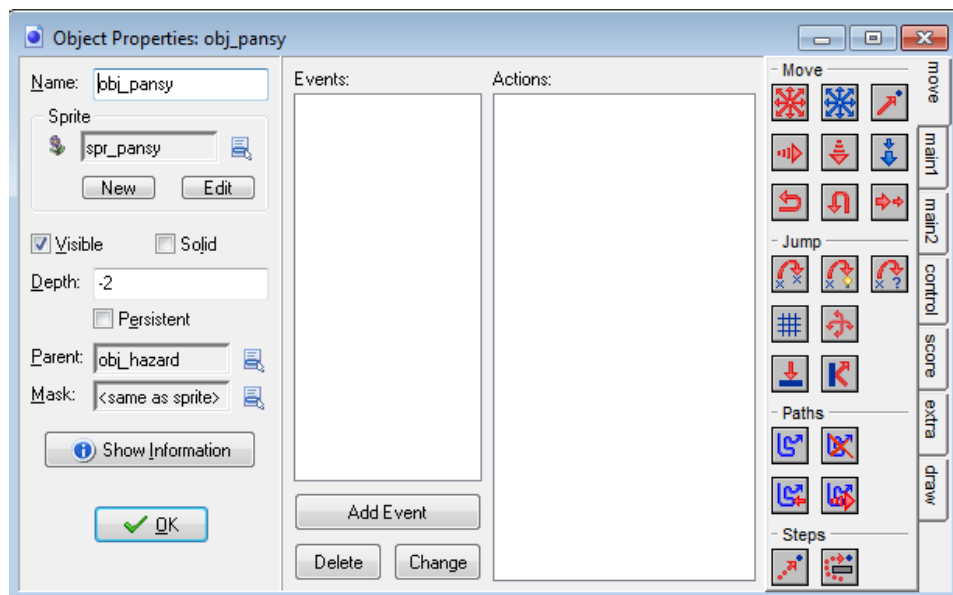
Now that we have the basic platform game mechanics in place, we should be able to turn this into a simple game by adding a few challenges. We'll do this by adding poisonous pansies and lava flows that send the player straight back to the start as soon as Fishpod collides with them.



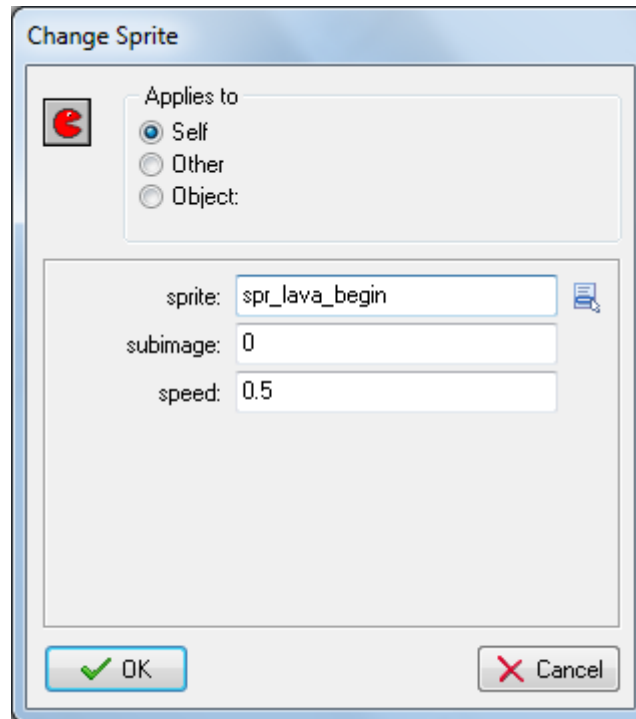
All of these objects need to have the same behavior, so we will use a parent object (**obj_hazard**) to make it possible to handle them all in the same collision event. It would also be nice if our lava flows finished themselves off automatically in the same way as the rock platforms do. So instances of the middle lava object will turn themselves into instances of the begin and end lava objects if there are no more hazards to either side of them. We will also make the begin and end lava objects add instances of platform objects to finish them off if there is nothing there already.

CREATING HAZARD OBJECTS AND THEIR CREATE EVENTS

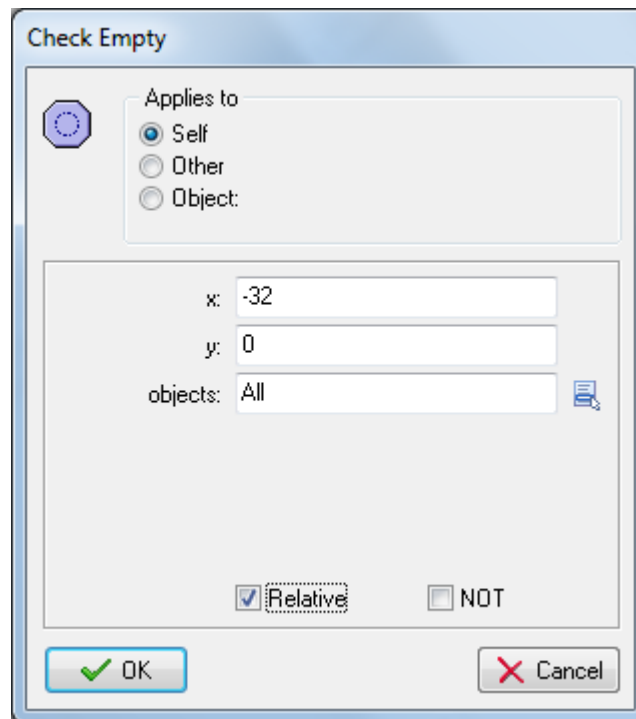
1. Create a new object called **obj_hazard** and close the properties form.
2. Create a new object called **obj_pansy**, select the **spr_pansy** sprite, set **obj_hazard** as the parent of this object and set its **Depth** to **-2** (in front of Fishpod, as this will ensure that the player can clearly see a hazard like this one when they collide with it).



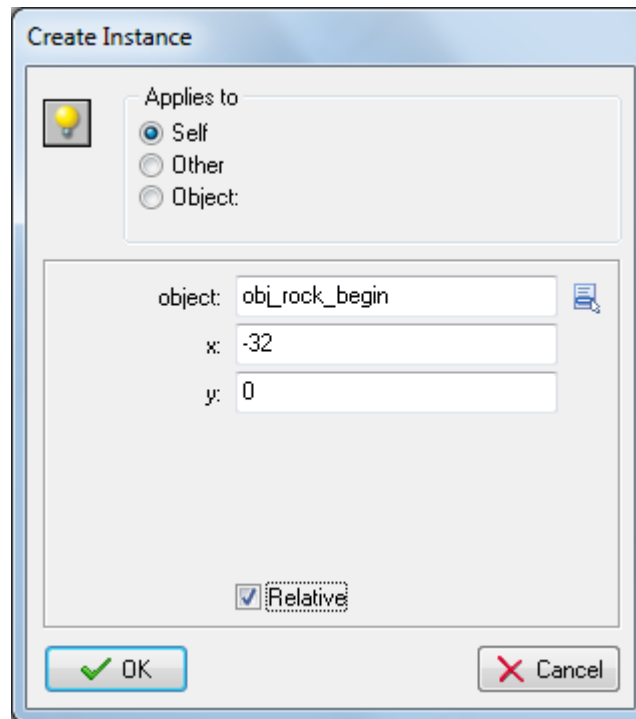
3. Create a new object called **obj_lava_begin**, select the **spr_lava_begin** sprite and set **obj_hazard** as the parent of this object. Unlike the pansy hazard, we will leave the **Depth** set to **0** for all the lava hazards because they are part of the platforms.
4. Add a **Create** event to **obj_lava_begin** and add a **Change Sprite** action with the sprite set to **spr_lava_begin**, **Subimage** to **0**, and **Speed** to **0.5**. This will slow the lava sprite's animation down by half.



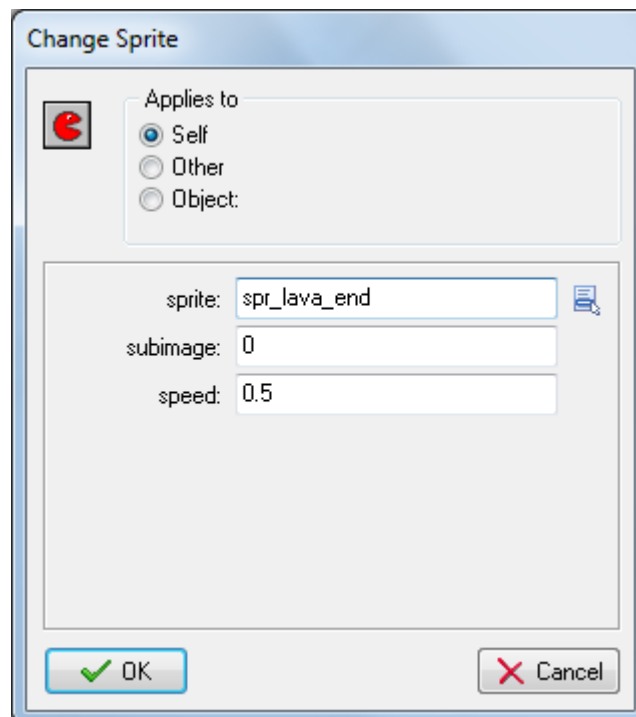
5. Add a **Check Empty** action that checks that there wouldn't be a collision if this instance was placed 32 pixels to the left of its current position (set **X** to **-32**, **Objects** to **all**, and check the **Relative** option).



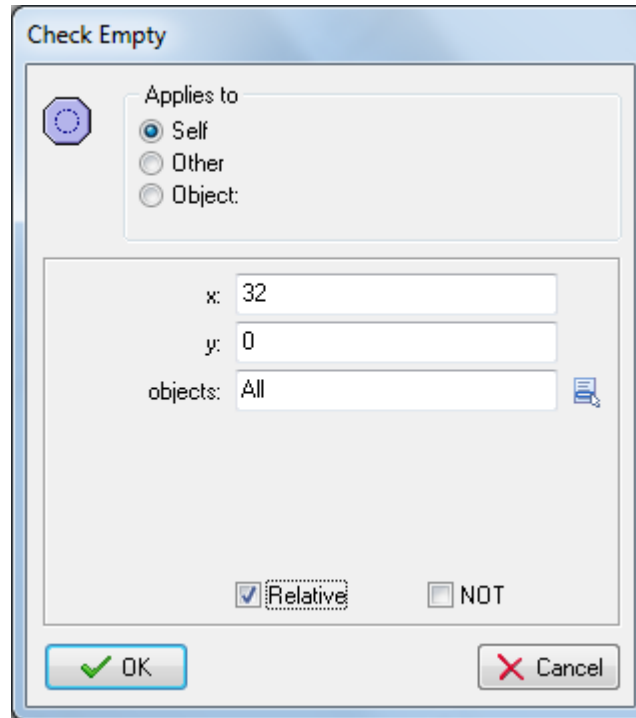
6. Add a **Create Instance** action that creates an instance of **obj_rock_begin** 32 pixels to the left of the current object (set **X** to **-32** and check the **Relative** option).



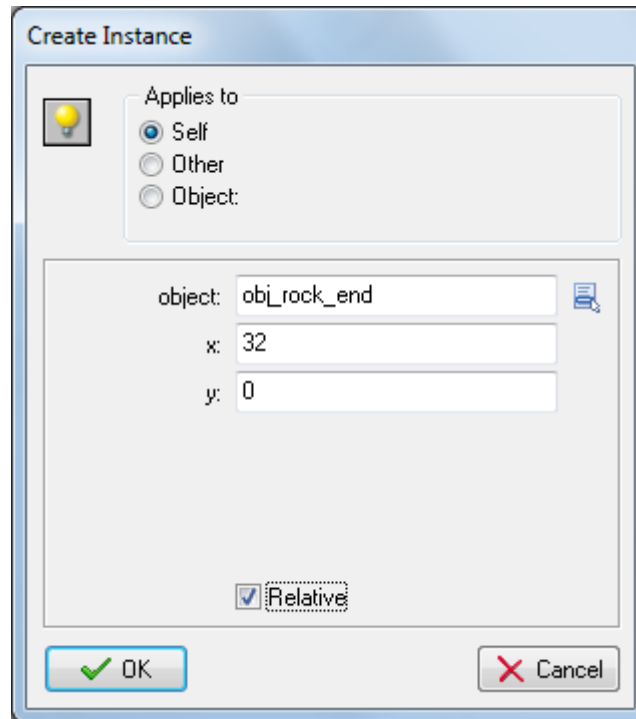
7. Create a new object called **obj_lava_end**, select the **spr_lava_end** sprite and set **obj_hazard** as the parent of this object.
8. Add a **Create** event to **obj_lava_end** and add a **Change Sprite** action with sprite set to **spr_lava_end**, **Subimage** to **0**, and **Speed** to **0.5**.



9. Add a **Check Empty** action that checks that there wouldn't be a collision if this instance was placed 32 pixels to the right of its current position (set **X** to **32**, **Objects** to **all**, and check the **Relative** option).

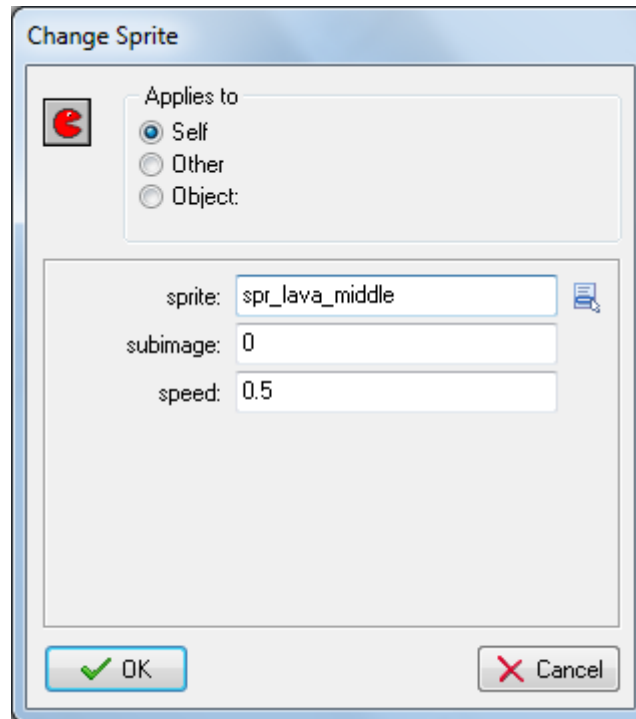


10. Add a **Create Instance** action that creates an instance of **obj_rock_end** 32 pixels to the right of the current object (set **X** to **32** and check the **Relative** option).

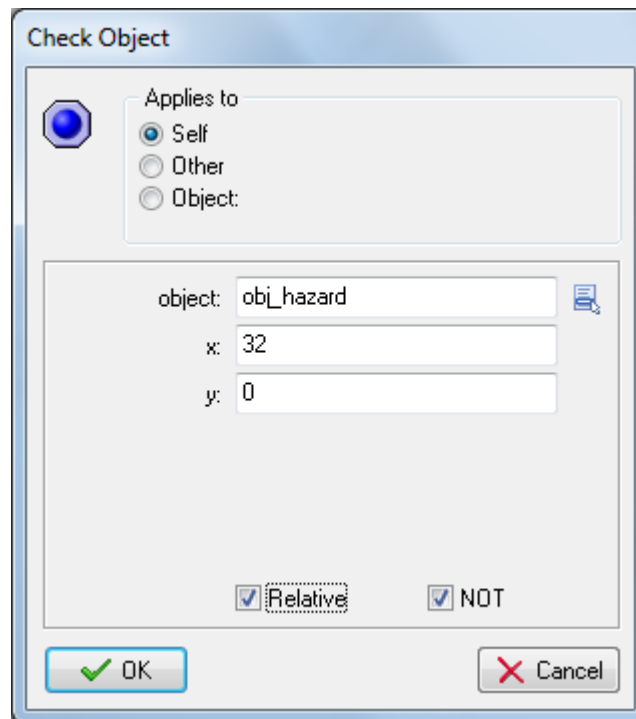


CREATING THE MIDDLE LAVA OBJECT AND ITS CREATE EVENT

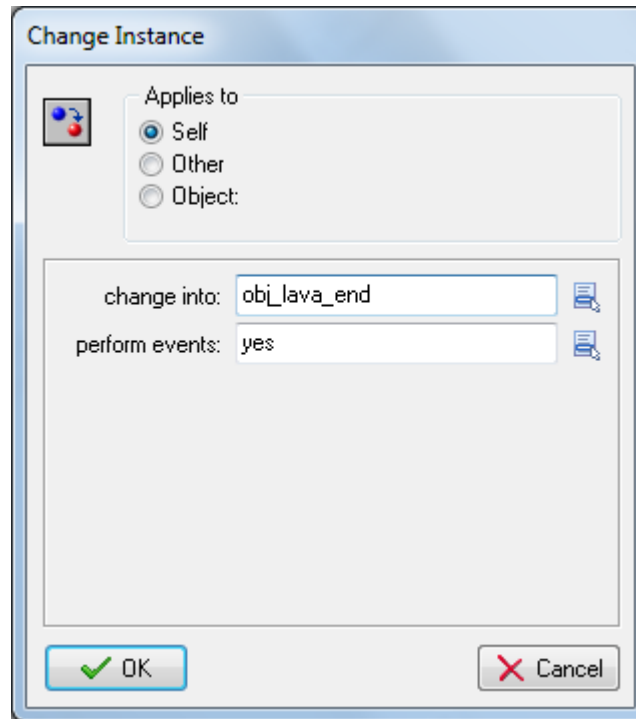
1. Create a new object called **obj_lava_middle**, select the **spr_lava_middle** sprite and set **obj_hazard** as the parent of this object.
2. Add a **Create** event to **obj_lava_middle** and add a **Change Sprite** action with sprite set to **spr_lava_middle**, **Subimage** to **0**, and **Speed** to **0.5**.



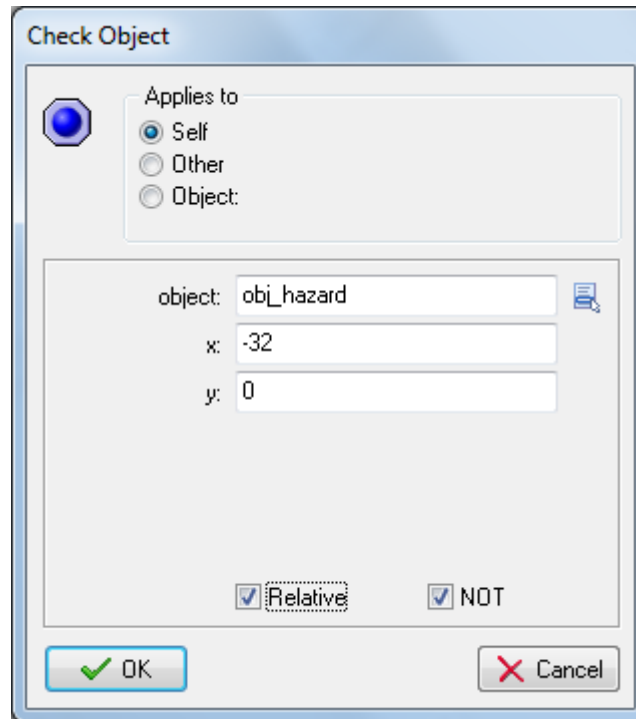
3. Add a **Check Object** action that checks if there would not be a collision with an instance of **obj_hazard** if the lava instance was placed 32 pixels to the right of its current position (set **X** to **32** and check both the **Relative** and **NOT** options).



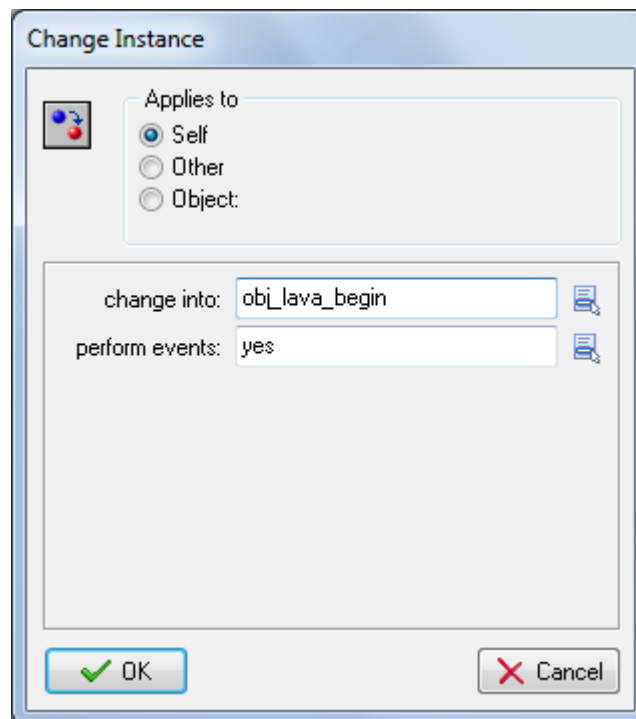
4. Add a **Start Block** action and add a **Change Instance** action that changes into **obj_lava_end** with **Perform Events** set to **yes**.



5. Add an **Exit Event** action to stop the remaining actions in the event from being executed.
6. Add an **End Block** action to mark the end of the grouped actions.
7. Add a **Check Object** action that checks that there would not be a collision with an instance of **obj_hazard** if the lava instance was placed 32 pixels to the left of its current position (set **X** to **-32** and check both the **Relative** and **NOT** options).

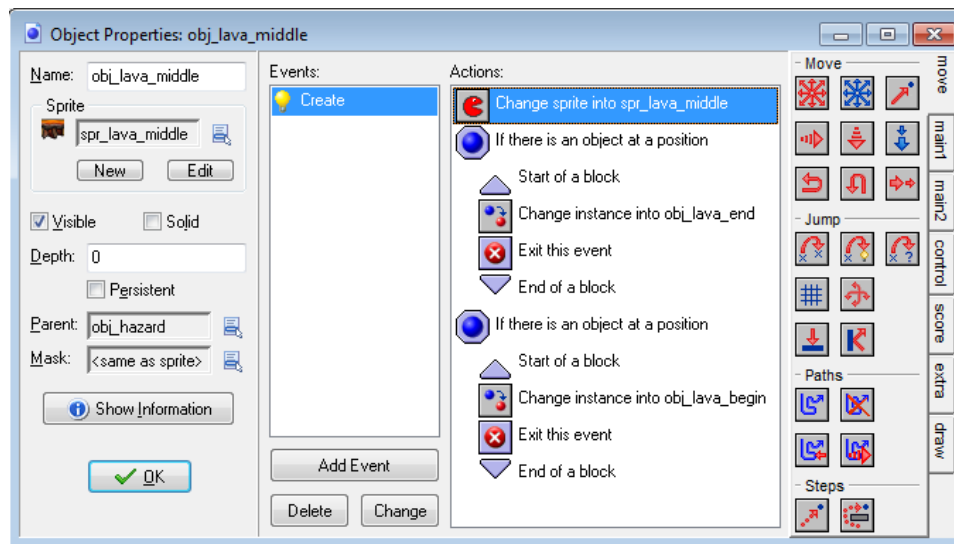


8. Add a **Start Block** action and add a **Change Instance** action that changes into **obj_lava_begin** with **Perform Events** set to **yes**.



9. Add an **Exit Event** action and then an **End Block** action to mark the end of the grouped actions.

The **Create** event for your **obj_java_middle** object should look like this when you're done:

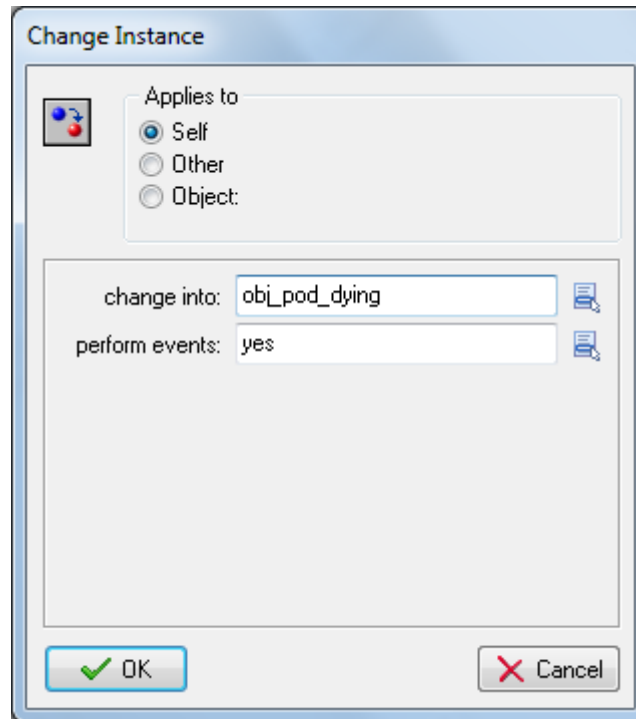


That takes care of the automatic creation of lava platforms.

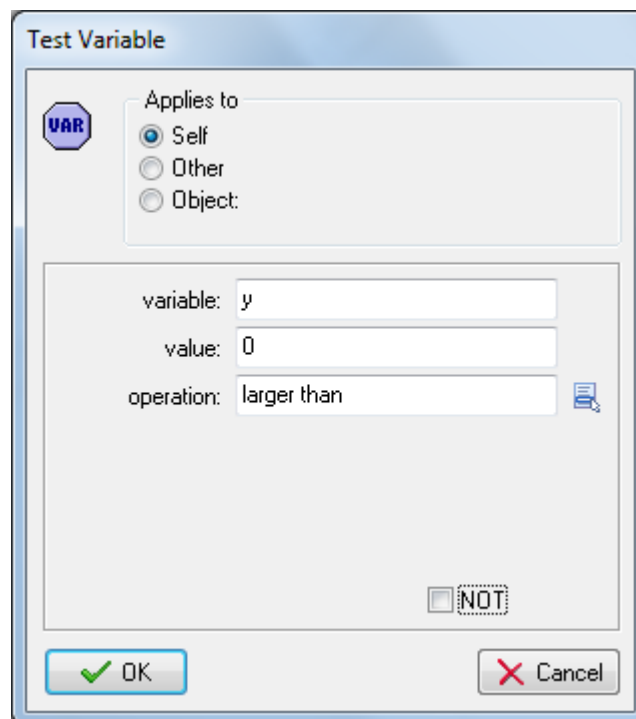
ADDING INHERITED EVENTS TO THE FISHPOD OBJECT

Next, we need to make sure that Fishpod restarts the level when he comes into contact with a hazard or goes off the edge of the screen. We already have a parent object for all hazards, which should make this easier, but things are a little more complicated for Fishpod. There are four different state objects that would require their own collision event with the hazard object (all of the states apart from the dying state object). So, rather than repeat ourselves four times, we will make **obj_pod** (the starting Fishpod object) a parent of these four states and then just handle the one collision event between **obj_pod** and **obj_hazard**.

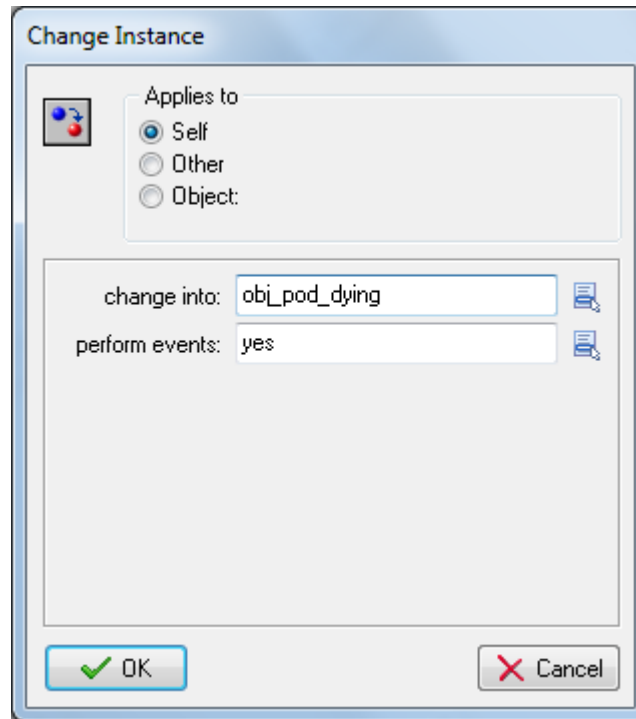
1. Open each of these state objects: **obj_pod_standing**, **obj_pod_walking**, **obj_pod_jumping**, and **obj_pod_falling** (but not **obj_pod_dying**) and set the parent of each one to be **obj_pod**.
2. Double-click **obj_pod** and add a **Collision** event with **obj_hazard**.
3. Add a **Change Instance** action that changes into **obj_pod_dying** with **Perform Events** set to **yes**.



4. Add an **Other > Outside Room** event and add a **Test Variable** action that checks if the **y** variable is **larger than 0**. This means that the next action will only occur if Fishpod is lower than the top of the screen (remember that y values start at 0 and increase down the screen). Gravity will bring Fishpod back down again if he jumps off the top of the screen, so there is no need to restart the level in this case.



5. Add a **Change Instance** action that changes into **obj_pod_dying** with **Perform Events** set to **yes**.

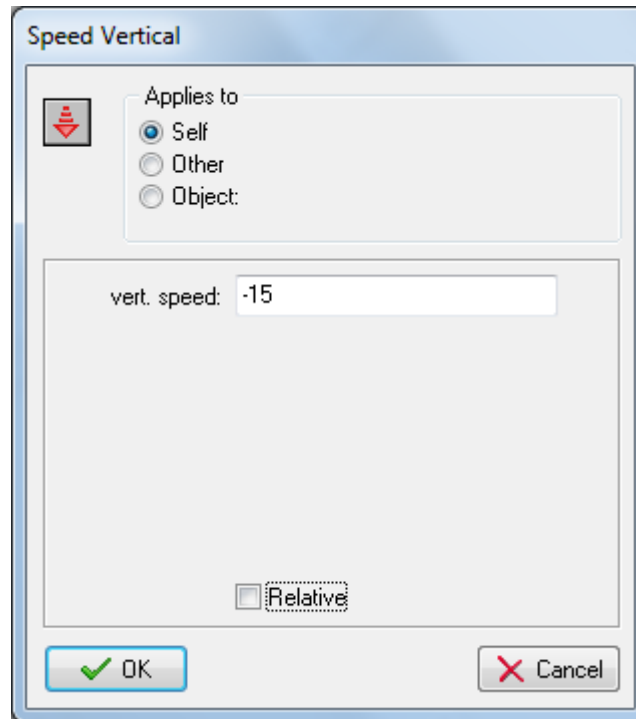


Even though we have only added these **Collision** and **Outside Room** events to **obj_pod**, they will also work for the four Fishpod state objects that have **obj_pod** as a parent. This is because objects inherit events from their parents. The Fishpod state objects have no **Collision** event defined with **obj_hazard** themselves, but their parent object (**obj_pod**) does, so they inherit this event and all its actions. Likewise for the **Outside Room** event. However, when both parent and child objects have the same event (like the Fishpod objects do for the **Create** event), the child's event overrides (replaces) the event of its parent.

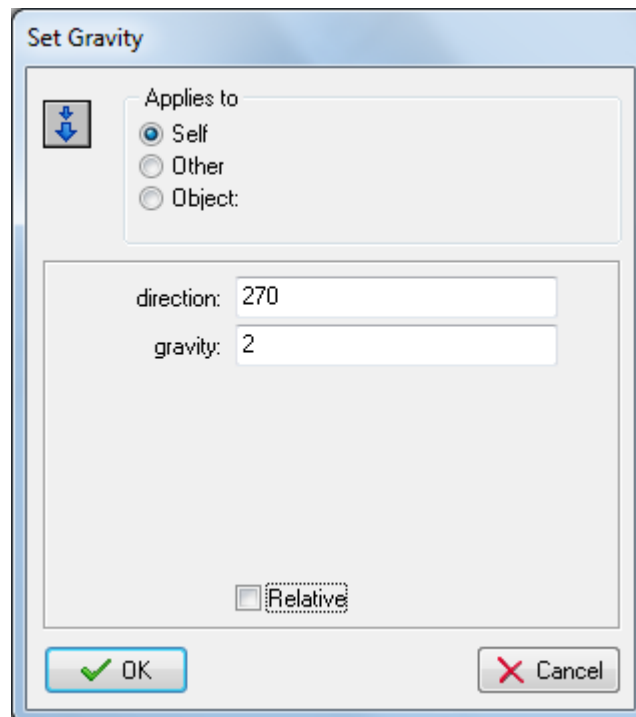
Now we need to create the correct behavior for the dying state object, as we haven't handled that yet. We'll make Fishpod fall off the screen and then restart the game when he's outside the room.

ADDING BEHAVIOURS FOR THE DYING STATE OBJECT

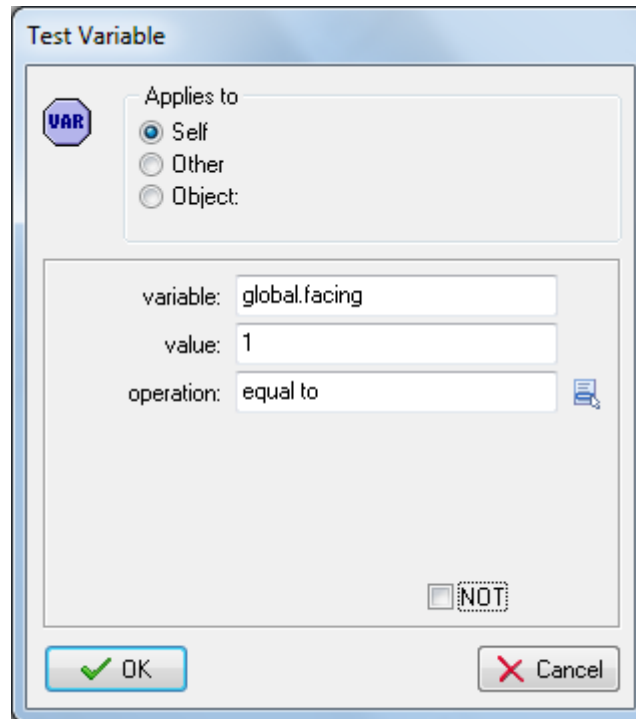
1. Double-click **obj_pod_dying** and add a **Create** event.
2. Add a **Speed Vertical** action and type **-15** into **Vert. Speed**. This will push Fishpod up into the air slightly before he gets pulled down by gravity.



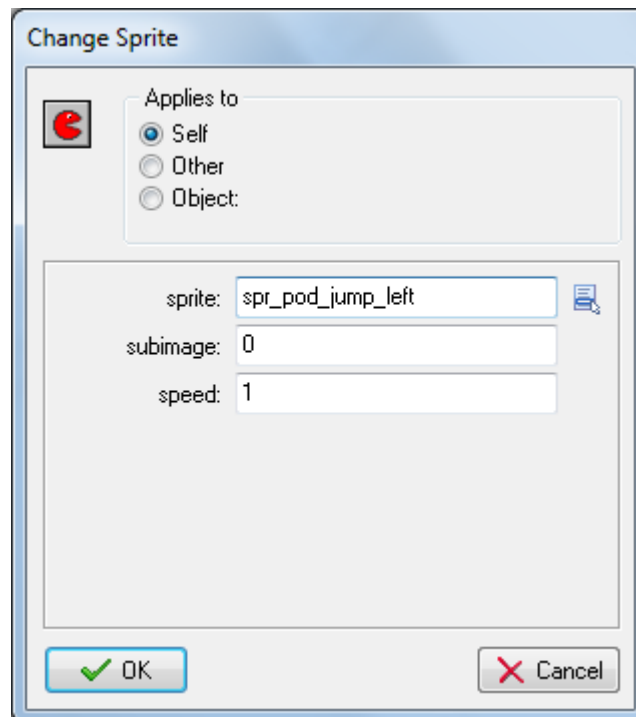
3. Add a **Set Gravity** action and type **270** into **Direction** (downward) and **2** into **Gravity**.



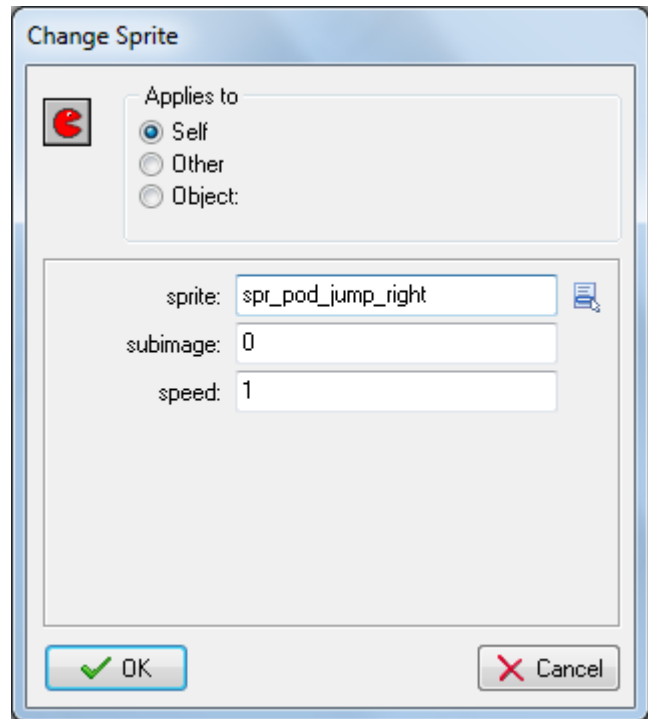
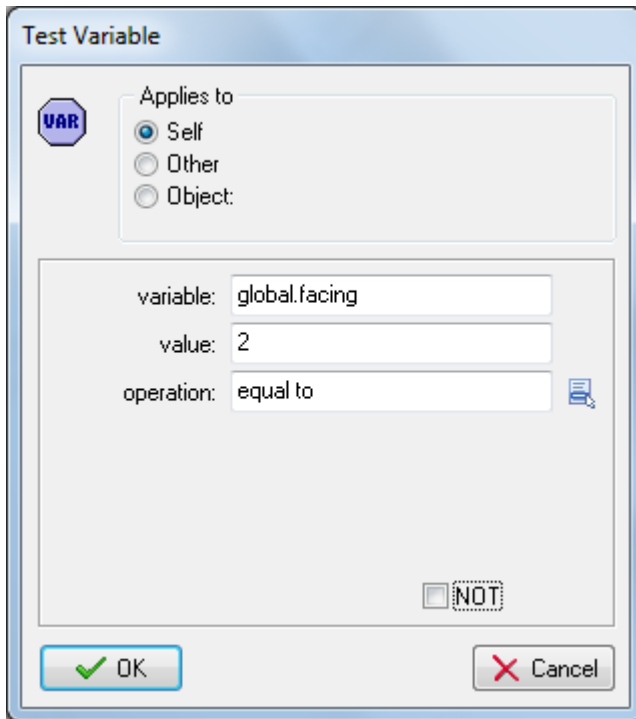
4. Add a **Test Variable** action, type **global.facing** into **Variable**, **1** into **Value**, and leave **Operation** set to **equal to**. This checks to see whether the character is facing left and only performs the next action(s) if this is true.



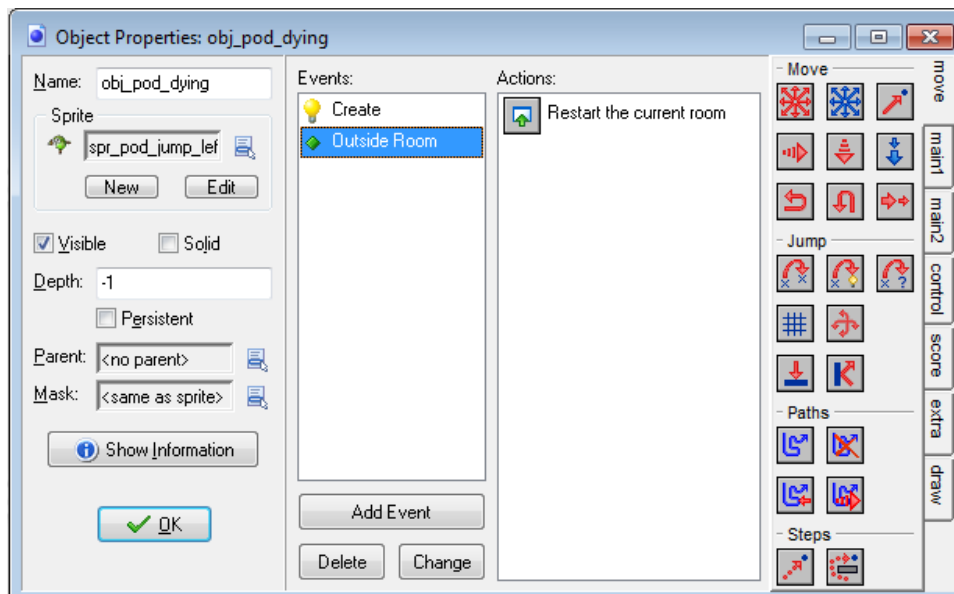
5. Add a **Change Sprite** action and choose the **spr_pod_jump_left** sprite.



- Repeat steps 4 and 5 testing for when it is facing right and change the sprite to **spr_pod_jump_right**.



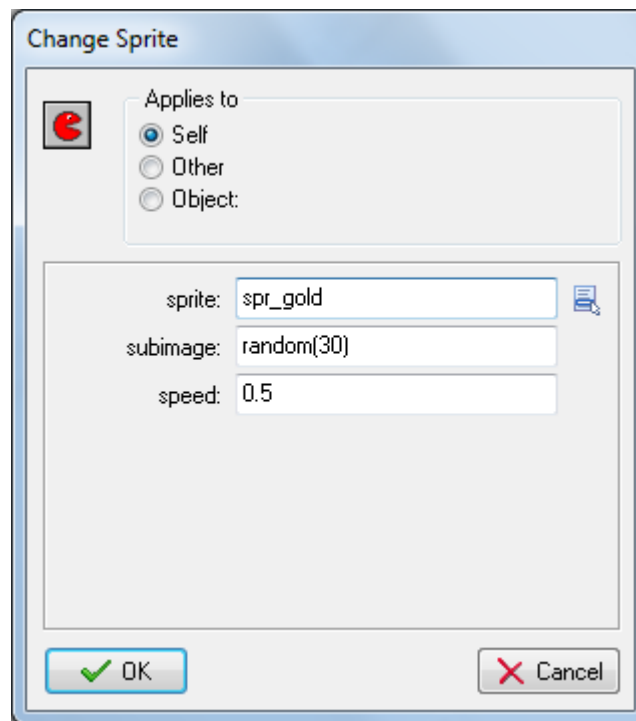
- Add an **Other > Outside Room** event and include a **Restart Room** action without setting any **Transition**.



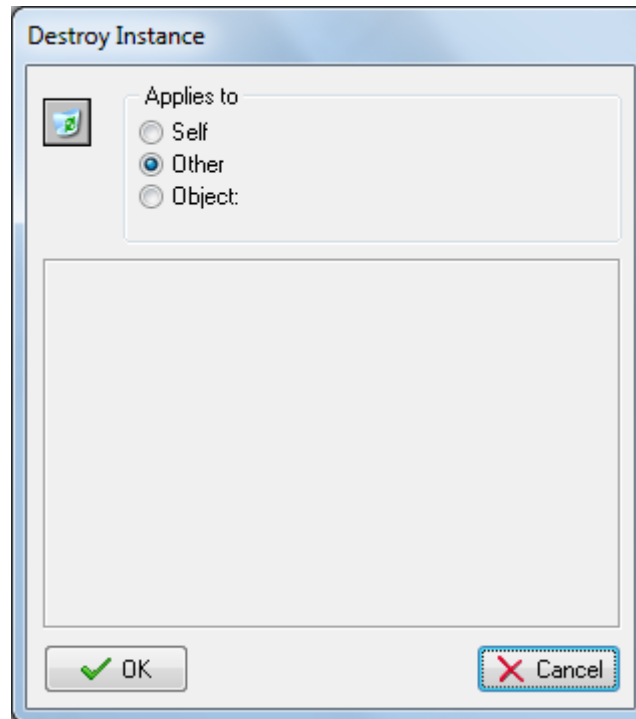
CREATING THE GOAL OBJECT

The final addition to our game will be to include some gold nuggets that Fishpod must collect in order to complete each level. Once there are no gold nuggets left on the current level, the game will automatically progress onto the next room.

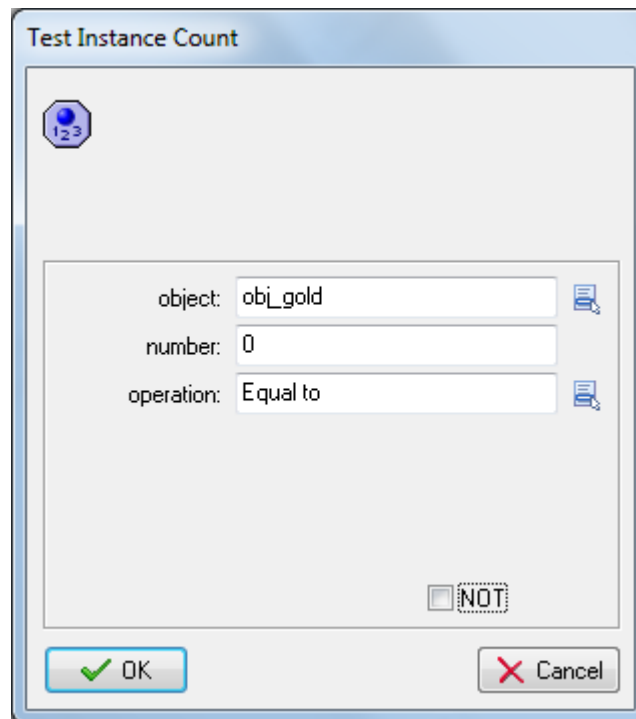
1. Create a new object called **obj_gold** and give it the **spr_gold** sprite.
2. Add a **Create** event, add a **Change Sprite** action, set the sprite to **spr_gold**, **Subimage** to **random(30)**, and **Speed** to **0.5**. This will set the gold sprite's start frame to a random subimage between 0 and 29 (as there are 30 frames in the sprite) and slow its animation down by half.



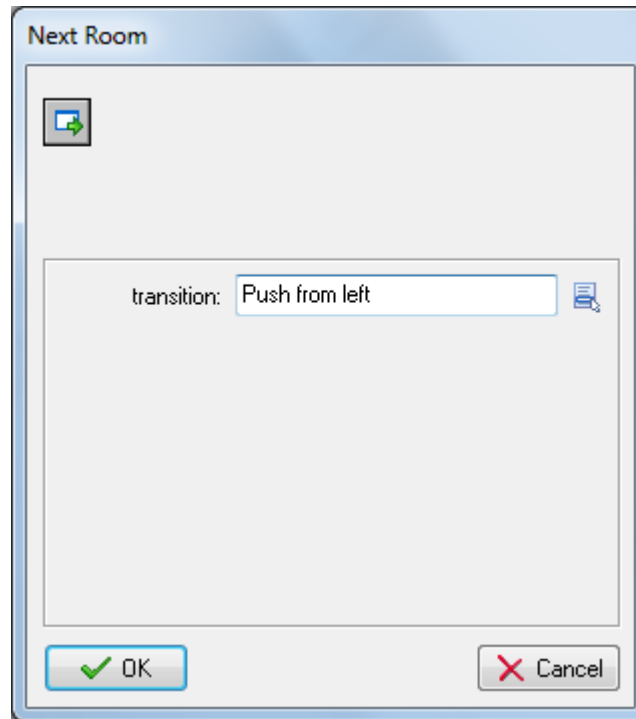
3. Double-click **obj_pod** and add a **Collision** event with **obj_gold**.
4. Add a **Destroy Instance** action and select **Other**. This will destroy the gold as it is the other instance in the collision.



5. Add a **Test Instance Count** action to test when the number of instances of **obj_gold** is equal to **0**. This means that the next action will only happen when the last instance of **obj_gold** has been collected.



6. Add a **Next Room** action and select your preferred **Transition**.



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.