Changing Camera Views!

Part 2: Simple Scene Change & Lighting Fixes

By Bella Onwumbiko
under the direction of
Professor Susan Rodger
Duke University
July 2013
In this tutorial, we will set up a penguin dance party scene that will occur inside of the igloo. If you remember, in Part 1, the Rockette walked up to the empty igloo, looked inside, and then turned around. We want to add some code to create a flexible scene change method, so that the next thing the user sees in the party scene! At the end of this tutorial, I’ll also show you how to fix some lighting issues that you may come across!

You will become more comfortable:
• Using a black square to create a scene change
• Creating methods with parameters
• Using the is Showing property
• Dropping dummy cameras to create interesting views
• Fix lighting issues
Section 1a: Setting up the World

Let’s add a scene that will occur in the igloo! When the rockette walks up to the igloo she won’t see anything... but the next time she looks, there will be a small dance party with penguins inside!

1) Click on the add objects button.

2) Use the camera keys to move the camera so that you can get a look inside of the igloo.

3) Drop a dummy camera at that position. Rename the ‘dummy’ to ‘iglooView’.
Now we need to add three penguins to the world! If you are using a PC, then you can simply drag the penguins into the igloo. If not, follow these instructions:

1) Open the Animals folder, and add a Penguin to the world.

2) Right click on Penguin in the object tree → set point of view to camera → move forward 5 meters

3) Use the buttons on the top right-hand corner of the screen to move the Penguin so that it is touching the ground.

4) Repeat these steps for 2 more Penguins, so that there are 3 Penguins in total.
The next thing we need to do is create a simple scene change. To do this, we need to add a black square to the world and move it to the front of the camera, so that the square acts as a lens cap.

Go back to the Local Gallery. Open the Shapes folder. Add a Square to the world.

Click on Square in the object tree. Go to the properties tab, and change the color of the Square to black.
To move the square to be right in front of the camera:

1) Right click on square in the object tree ➔ set point of view to camera

2) Right click on square in object tree ➔ move forward 1 meter

3) Right click on square in object tree ➔ move up 1 meter

4) Click on square in the object tree. Go to the properties tab. Set the vehicle of the square to be the camera.

Click done!
The next thing we want to do is set the camera’s point of view back to the originalView.

Click on camera in the object tree. Click and drag set point of view to originalView so that it is the last line of code in the method editor. Set the duration to 3 seconds.
Now we want to create a scene change method. This method will have 3 parts:

1) Square **move** down 1 meter
2) Camera **set point of view to** (camera view)
3) Square **move** up 1 meter

Click on **world** in the object tree. Click on the **methods** tab. Click on ‘create new method.’ Name the method **sceneChange**.

Drag in the code to make your method look like this:
Cool! The `sceneChange` method is almost complete. As of now, it will only change the camera’s POV to the iglooView. What if we want to change it to the frontView or the originalView? To make the method more flexible, we will add a parameter!

In the `world.sceneChange` method, click on ‘create new parameter.’ Call it ‘cameraView.’ Change the type to `Object`.
Next, we will drag and drop the ‘cameraView’ parameter on top of where it says ‘iglooView.’

Now, this method is done!
Section 2: Writing the Code!

Next, we want to set up a new scene! Click on **world** in the object tree. Click on the methods tab. Click on ‘create new method.’ Call it ‘scene2’.

This method is going to have the code for the penguins’ dance. You can use your creativity here! Check out all of the built in methods that come with the penguin!

**For each of the three penguins**, click and drag different commands into the method editor.

To test out what your **scene2** will look like, change the event in the events editor so that it reads: When the world starts, do **world.scene2**. Then, press play!
Section 2: Writing the Code!

Here is what my some of my code for scene2 looks like:

- Do together
  - `penguin.jump times = 2`
  - `penguin2.jump times = 2`
  - `penguin3.jump times = 2`

- Do together
  - `penguin.turn_head_left`
  - `penguin3.turn_head_right`

- Do together
  - `penguin2 turn right` 3 revolutions `duration = 3 seconds`
  - More...

- Do in order
  - `penguin resize 2` more...
  - `penguin3 resize 2` more...

- Do together
  - `penguin resize 0.5` more...
  - `penguin3 resize 0.5` more...

PS – there is more code on the next slide!
This is the rest of my code in scene2:

Do together

- `penguin2.glide`
- `penguin.roll left 3 revolutions more...`
- `penguin3.roll left 3 revolutions more...`

Do together

- `penguin.rightWing turn backward 0.5 revolutions more...`
- `penguin.leftWing turn backward 0.5 revolutions more...`
- `penguin3.rightWing turn backward 0.5 revolutions more...`
- `penguin3.leftWing turn backward 0.5 revolutions more...`

You don’t need to copy mine exactly. This is just to give you an idea of how to make the penguins move around in the igloo!
The next thing we have to do is set the `isShowing` property of the `penguins` to `false`. Remember, when the `rockette` first looks into the igloo, she shouldn’t see anything!

Click on `penguin` in the object tree. Click on the properties tab. Set the `isShowing` to `false`.

Repeat this for `penguin2` and for `penguin3`.

Now, you shouldn’t be able to see the penguins in the object viewer.
However, when `scene2` is called, we want the penguins to become visible again. This time we want to drag and drop the `is Showing` property for each penguin into the `scene2` method. We want to set `is Showing` to `true`.

At the very top of the `scene2` method, drag up a `Do together`.
Click on penguin in the object tree. Click on the properties tab. Drag and drop the is Showing property into the first Do together. Set it to true. Set the duration to 0 seconds.

Repeat this for penguin2 and penguin3 as well.
Final Code for scene2:

Do in order

- penguin set isShowing to true \( duration = 0 \) seconds
- penguin2 set isShowing to true \( duration = 0 \) seconds
- penguin3 set isShowing to true \( duration = 0 \) seconds

Do together

- penguin.jump \( times = 2 \)
- penguin2.jump \( times = 2 \)
- penguin3.jump \( times = 2 \)

Do together

- penguin.turn_head_left
- penguin3.turn_head_right

PS – there is more code on the next slide!
Section 2: Writing the Code!

Final Code for scene2:

Do together
- penguin2 -- turn right -- 3 revolutions -- duration = 3 seconds -- more...

Do in order
- penguin -- resize 2 -- more...
- penguin3 -- resize 2 -- more...

Do together
- penguin -- resize 0.5 -- more...
- penguin3 -- resize 0.5 -- more...

Do together
- penguin2.glide
- penguin -- roll left -- 3 revolutions -- more...
- penguin3 -- roll left -- 3 revolutions -- more...

PS – there is more code on the next slide!
## Final Code for *scene2*:

<table>
<thead>
<tr>
<th></th>
<th>Action</th>
<th>Value</th>
<th>More?</th>
</tr>
</thead>
<tbody>
<tr>
<td>penguin.rightWing</td>
<td>turn backward</td>
<td>0.5 revolutions</td>
<td>more...</td>
</tr>
<tr>
<td>penguin.leftWing</td>
<td>turn backward</td>
<td>0.5 revolutions</td>
<td>more...</td>
</tr>
<tr>
<td>penguin3.rightWing</td>
<td>turn backward</td>
<td>0.5 revolutions</td>
<td>more...</td>
</tr>
<tr>
<td>penguin3.leftWing</td>
<td>turn backward</td>
<td>0.5 revolutions</td>
<td>more...</td>
</tr>
</tbody>
</table>
Great! Now that we have our `scene2` set up. We can change the event in the event editor back to ‘When the world starts, do world.my first method.’

Next, we need to do is add the `sceneChange` method to the bottom of world.my first method. Click on the `world.my first method` tab to open it. Next, click on `world` in the object tree. Click on the methods tab. Drag and drop the `sceneChange method → cameraViews → iglooView` so that it is the last line of code in `myfirstmethod`. 
Section 2: Writing the Code!

Next, drag in the `scene2` method to be the last line of code in `world.myfirstmethod`.

Lastly, when the penguins finish their dance (i.e. when scene2 ends) we want the black square to move down 1 meter to act as a lens cap over the camera…signaling the end of the Alice world.

Click on square in the object tree. Drag the move command and drop it below the `world.scene2` method.
NOTE: The position that your camera is in in the object viewer is the same position that the camera will be in when the world starts.

We want this world to start in the originalView. In the object tree, right click on the camera:

Camera → set point of view to → cameraViews → originalView

NOW, Play the world to see all the magic unfold!
Section 2: Writing the Code!

Final Code for `world.myfirstmethod`:

```plaintext
Do together
- Loop 20 times times show complicated version
  rockete.walk howFar = 1

Do in order
- camera set point of view to originalView duration = 3 seconds more...
- camera set point of view to rightSideView duration = 3 seconds more...
- camera set point of view to frontView duration = 3 seconds more...
- camera set point of view to leftSideView duration = 3 seconds more...
- camera set point of view to rockete.hips.stomach.torso.collar.head duration = 3 seconds more...

rockete turn left 0.5 revolutions duration = 2 seconds more...

world.sceneChange cameraView = iglooView

world.scene2
square move down 1 meter more...
```
Got a problem with the lighting in the world? Here are some simple solutions:

1. Set the vehicle of the light to be the camera. Whenever the camera moves, the light will move with it.

   Light $\rightarrow$ properties tab $\rightarrow$ Seldom Used Properties $\rightarrow$ vehicle $\rightarrow$ camera

- Light’s vehicle = world
- Light’s vehicle = camera
Got a problem with the lighting in the world? Here are some simple solutions:

2. You can make an object look brighter by changing its emissive color to light grey:

- emissiveColor property of penguins and igloo is set to black
- emissiveColor property of penguins and igloo is set to light grey