

# 1 Getting Started

To start, we'll create a basic project that demonstrates the features of Alice and explains the interface. It will show how to create a scene and make characters move.

The 'story' is that of an alien (the 'alien' object) on the moon who is startled by the alien-like robot (the 'alienRobot' object) and runs away from it.

1. Open 'Alice 3' and wait until the 'Select Project' screen appears (on the first run, it may take a few moments).

Templates provide a basic scene pre-made. They contain a 'ground' (e.g. the moon surface) and an 'atmosphere' colour (e.g. black). Both of these options can be changed later in the scene designer.

2. Select the 'MOON' template and click 'OK'. A new window will appear, like the one in Figure 1, showing the code view for your animation.

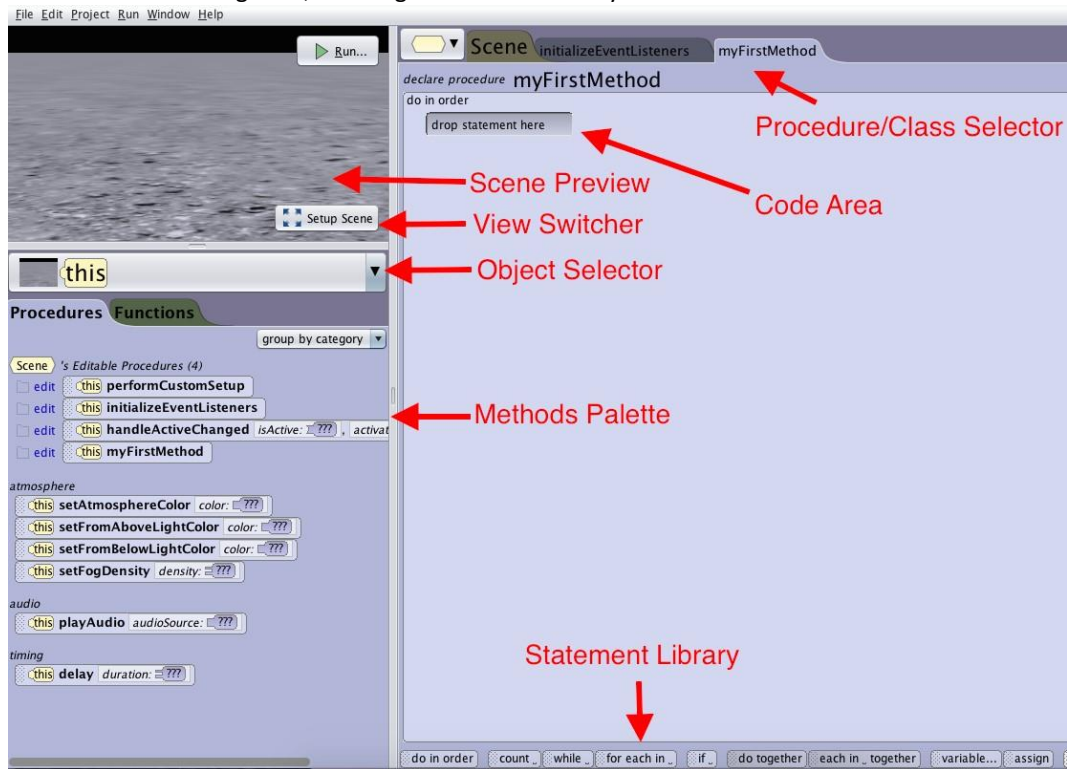


Figure 1: The code view of Alice 3

- Now, to start laying out your scene, click the view switcher button ('SetupScene') to switch to the scene view.

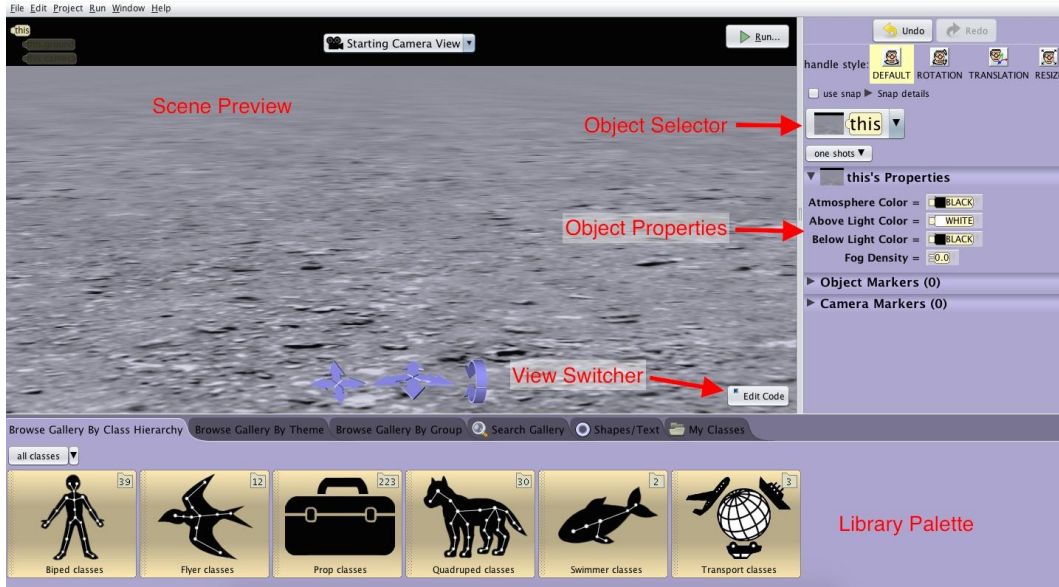


Figure 2: The scene view of Alice 3

- In the library palette, select 'Biped classes' and click on the 'new Alien()' icon, then press OK on the dialog that will open. This will place an alien character at the center of your scene with the name 'alien'.
- Now click on the arrow next to the 'all classes' button to return to the opening menu.



Figure 3: Click to return to the full library palette

- From the library, choose 'Quadruped classes' and then click on 'new AlienRobot()'. In the dialog that appears, select 'OK' to use the default name ('alienRobot') and configuration.

- With the 'alienRobot' character selected, change its x-position to 5.0 and then press ENTER (if the text goes gray, the change has not been applied - reselect the field and press ENTER). This will move the character to the left of the camera.

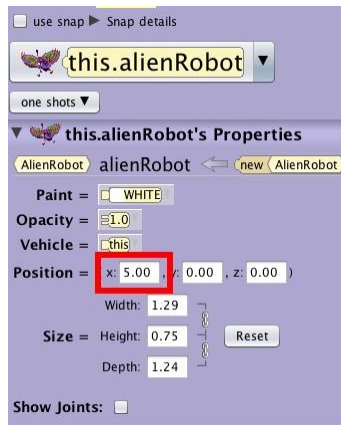


Figure 4: The properties of a character; note the change in the positioning

Perhaps counterintuitively, the positioning is from the character's perspective, rather than the camera's position. Therefore, a positive shift in the x-axis will move an object left and a negative shift will move it right. This also applies when talking about turning the character the 'left' and 'right' are relative to the character, not the camera.

- Switch back to the code view by selecting the view switching button ('EditCode').

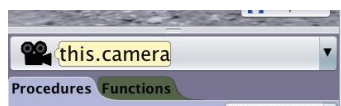


Figure 5: The 'Object Selector' drop down, with 'this.camera' selected

- From the 'Object Selector' drop down, select 'this.camera' and drag a 'moveAndOrientToAGoodVantagePointOf' procedure to the code area inside the 'do in this order' block. In the menu that appears, select the 'this.alienRobot' object to target the camera on.



Figure 6: The 'moveAndOrientToAGoodVantagePointOf' procedure

### 1.1 Making the alien see the robot

10. From the 'Object Selector' drop down, select 'this.alien' and drag a 'turnToFace' procedure to the code area underneath the 'moveAndOrientToAGoodVantagePointOf'. In the menu that appears, select the 'this.alienRobot' object to turn towards.
11. Add a 'think' procedure for 'this.alien' to the code area and select 'CustomTextString' in the pop-up menu. In the textbox, enter 'Oh no...' and press 'OK'. Click on 'add detail' and set the duration to 0.25 seconds.
12. Add a 'delay' procedure for 'this.alien' to the code area and set the duration to 0.25 seconds.
13. Add a 'turn' procedure for 'this.alien' to the code area and set the direction to 'LEFT' and amount to 0.5 rotations.



Figure 7: The code so far

At this stage, the code should look like that in Figure 7. Remember to save your work regularly (File > Save). You can also use the 'Run' button to see what the animation is doing already.

### 1.2 Being chased away

14. From the statement library on the bottom of the screen, drag a 'do together' block into the code area underneath the 'turn' procedure.

Up to this point, all of the procedures have been inside a 'do in order' block (as the block is there by default in new projects). Inside this block, the procedures run sequentially.

The 'do together' block will run each statement in parallel with the next. This means the actions will happen at the same time, however it will wait until the last statement is finished before continuing.

*If two procedures conflict with each other (e.g. two 'move' procedures on the same object), then unexpected behaviour may occur, including only one procedure running or neither running at all.*

15. With the 'this.alien' object selected, drag a 'move' procedure into the 'do together' block. Set the direction to forward and the amount to 'Custom DecimalNumber'. In the pop-up window, enter '7.0' and press 'OK'
16. In the object selector dropdown, change to 'this.alienRobot' and drag a 'turnToFace' procedure into the 'do together' and set the target to be 'this.alien'. Set the duration to 0.25 by clicking the 'add detail' button and selecting duration.
17. With the 'this.alienRobot' object selected, drag a 'moveToward' procedure into the 'do together' block and set the target to be 'this.alien'. Set the duration to 'Custom DecimalNumber' and in the pop-up window, enter '11.0' and press 'OK'.

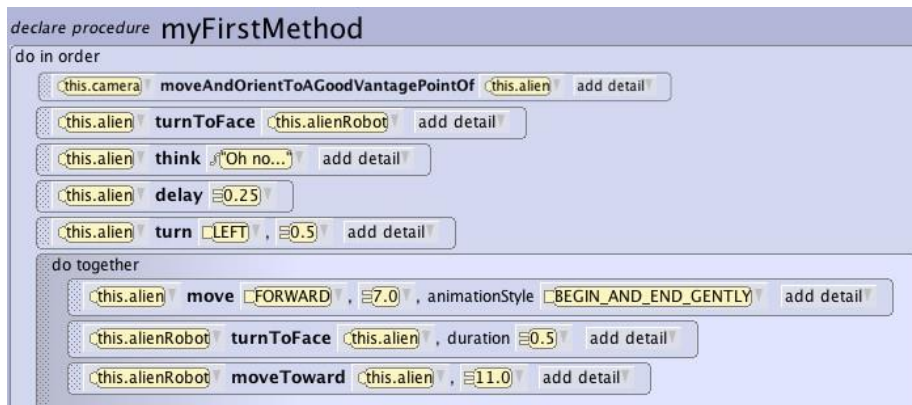


Figure 8: The code with the robot chasing the alien

18. Now save the project and press 'Run' to see the animation so far.

## 2 Adding Props and Camera Markers

To make the scene more realistic, a building will be added as a background prop that the characters chase around. To make the robot chasing the alien around the right side of the building visible, two different camera angles will be required.

1. Click on the view switcher button ('Setup Scene').
2. In the library palette, select 'Prop classes'. This category contains a selection of inanimate objects that can be used as scenery. They also have the same

standard procedures as the character classes - 'turn', 'move', and 'roll' are all available.

3. Select the 'new MarsOutpostBunker()' and drag it into the scene, placing it behind the alien and the alienRobot objects. In the dialog window that pops up, click 'OK' to select the default object name of 'marsOutpostBunker'.

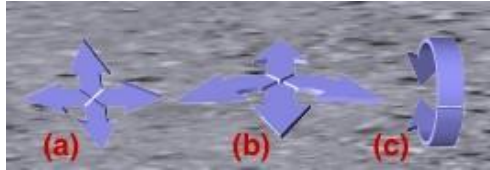


Figure 9: Controlling the camera - (a) repositions the camera, (b) reorientates (pans) the camera, (c) rotates the camera on the X-axis

4. Use the three sets of buttons to adjust the view of the camera so that the 'alienRobot' and 'alien' objects are centered in the view and the top of the bunker is visible.

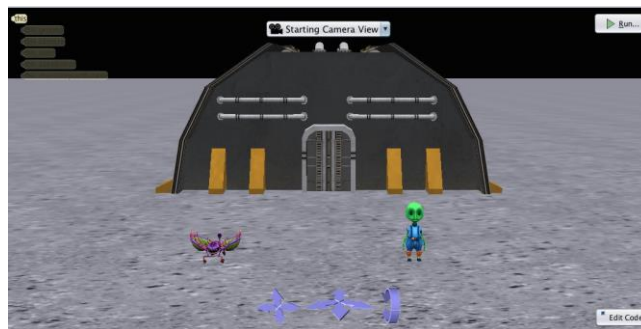


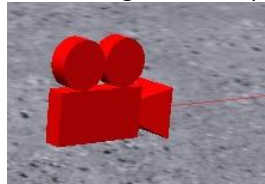
Figure 10: An example camera position for the front perspective

5. Once all three objects are in the frame, go to the properties panel on the right and expand the 'Camera Markers' section.
6. Click 'Add Camera Marker...'. In the window that pops up, change the name to 'bunkerFront' and click 'OK'.

A camera marker saves the current position and orientation of the camera so that it can be accessed programmatically from within the code. They can also be utilised from within the scene editor by selecting the marker in the 'Camera Markers' section of the properties panel and pressing the button below. This will reposition the camera to the saved position.



After setting a camera marker, when moving around the scene, a floating camera icon will be visible indicating the saved position of the camera.



7. Use the camera control buttons to reposition the camera and orientate it towards the right side of the bunker, as shown in Figure 11. Dragging the cursor around the scene will also reposition the camera.



Figure 11: An example camera position for the right-side perspective

8. When the camera is positioned and oriented, click the 'Add Camera Marker...' button in the properties panel. In the window, change the name to 'bunkerRight' and click 'OK'.
9. Switch back to the code editor.

10. Right-click the 'moveAndOrientToAGoodVantagePointOf' block in the code area and select 'Delete'.
11. In the object selector dropdown, select 'this.camera'.
12. Drag a 'moveAndOrientTo' procedure to the top of the code area above the 'turnToFace' procedure and select 'this.bunkerFront' as the target. This will make the scene start with the camera focussing on the front of the bunker, where the characters are initially.
13. From the statements palette on the bottom of the screen, drag a 'do together' block to the end of the code (below the previous 'do together' block).
14. Drag a 'moveAndOrientTo' procedure inside the new 'do together' block and select 'this.bunkerRight' as the target.



Figure 12: The code with two different camera angles utilised

15. Press 'Run' to preview the scene and see the camera transitions.
16. In the object selector dropdown, change to 'this.alien'.
17. Add a 'turn' block to the 'do together' block at the bottom of the code area and set the direction to 'LEFT' and amount to 0.25. Click on 'add detail' and set the duration to 0.25 seconds.
18. From the statements palette on the bottom of the screen, drag another 'do together' block to the end of the code (below the previous 'do together' block).



19. With 'this.alien' selected, add a 'move' block to the new 'do together' block at the bottom of the code area and set the direction to 'FORWARD' and amount to 1.0. Click on 'add detail' and set the duration to 0.5 seconds.
20. In the object selector dropdown, select 'this.alienRobot' and drag a 'turn' block into the 'do together' block. Set the direction to 'LEFT' and amount to 0.25 rotations. Click 'add detail' and set the duration to 0.5 seconds.
21. With 'this.alienRobot' selected, drag a 'move' block into the 'do together' block. Set the direction to 'RIGHT' and amount to 1. Click 'add detail' and set the duration to 0.5 seconds. This block, combined with the 'turn' block above, will make the robot arc towards the alien as it turns the corner.
22. From the statements palette on the bottom of the screen, drag another 'do together' block to the end of the code (below the three previous 'do together' blocks).
23. With 'this.alien' selected, add a 'move' block to the new 'do together' block at the bottom of the code area and set the direction to 'FORWARD' and amount to 'Custom DecimalNumber'. In the popup input, enter 22.0. Click on 'add detail' and set the duration to 2 seconds.
24. In the object selector dropdown, select 'this.alienRobot' and drag a 'move' block into the 'do together' block. Set the direction to 'FORWARD' and amount to 'Custom DecimalNumber'. In the popup input, enter 22.0. Click on 'add detail' and set the duration to 2 seconds.
25. Now press 'Run' to preview the scene and see the alien and robot chase along the right side of the building.

```

declare procedure myFirstMethod
do in order
  this.camera moveAndOrientTo this.bunkerFront add detail
  this.alien turnToFace this.alienRobot add detail
  this.alien think "Oh no..." add detail
  this.alien delay 0.25
  this.alien turn LEFT, 0.5 add detail
do together
  this.alien move FORWARD, 7.0, animationStyle BEGIN_AND_END_GENTLY add detail
  this.alienRobot turnToFace this.alien, duration 0.5 add detail
  this.alienRobot moveToward this.alien, 11.0 add detail
do together
  this.camera moveAndOrientTo this.bunkerRight add detail
  this.alien turn LEFT, 0.25, duration 0.25 add detail
do together
  this.alienRobot move RIGHT, 1.0, duration 0.5 add detail
  this.alienRobot turn LEFT, 0.25, duration 0.5 add detail
  this.alien move FORWARD, 1.0, duration 0.5 add detail
do together
  this.alien move FORWARD, 22.0, duration 2.0 add detail
  this.alienRobot move FORWARD, 22.0, duration 2.0 add detail

```

Figure 13: The code to animate the chase around two sides of the building

### 3 Custom Procedures

To make the scene look more realistic, the movement of the alien needs to be refined to make it look like it is walking, not gliding, along the surface. To do this, two new procedures are needed: one to animate the character stepping, and one to repeat this procedure for as many steps as are needed.

#### 3.1 Creating the 'step' procedure

1. In the code editor, click the hexagonal class selector button in the top left. In the drop down that appears, select 'Biped' > 'Add Biped Procedure' and enter the name as 'step'.



Figure 14: This button opens a menu of all classes, procedures and functions

- Click the 'Add Parameter...' at the top and in the pop-up window, set the value type to 'DecimalNumber' and the name to 'quarterDuration', then press 'OK'. This will be the duration for each statement and will eventually be equal to one quarter of the total duration of the procedure.
- From the statements library, drag a 'do together' block to the code area.

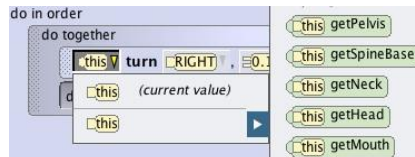


Figure 15: Click the left-side button on the block to change the 'calling object'

- From the procedures palette, add a 'turn' block into the 'do together' block. Set the direction to 'RIGHT' and the amount to 'Custom DecimalNumber'. In the pop-up, enter 0.125. Click on the 'this' button on the left side of the block and select 'this' > 'this.getRightShoulder' from the dropdown menu that appears. This will change which object calls the 'turn' method. Click 'add detail' and set the duration to the 'quarterDuration' parameter (the bottom option in the dropdown).

Each of the characters is made of a number of a 'joints' that are actually separate objects that form a larger object (the character itself).

These objects can be accessed and controlled through a number of functions such as 'this.getHead'. Although these methods are visible from the functions tab of the methods palette, the easiest way to utilise them in Alice is to change the object that a method (such as 'turn') is to be called on (the 'calling object') after the block has been placed in the code area.

- Add another 'turn' block to the 'do together' block and set the direction to 'RIGHT' and the amount to 0.125. Change the object calling the method to 'this.getLeftShoulder'. Click 'add detail' and set the duration to the 'quarterDuration' parameter.
- Add another 'turn' block to the 'do together' block and set the direction to 'BACKWARD' and the amount to 'Custom DecimalNumber'. In the pop-up window, set the value to 0.08 and then change the object calling the method to 'this.getRightHip'. Click 'add detail' and set the duration to the 'quarterDuration' parameter.
- Add another 'turn' block to the 'do together' block and set the direction to 'FORWARD' and the amount to 'Custom DecimalNumber'. In the pop-up window, set the value to 0.08 and then change the object calling the method to

- 'this.getLeftHip'. Click 'add detail' and set the duration to the 'quarterDuration' parameter.
8. Add another 'turn' block to the 'do together' block and set the direction to 'FORWARD' and the amount to 'Custom DecimalNumber'. In the pop-up window, set the value to 0.08 and then change the object calling the method to 'this.getRightKnee'. Click 'add detail' and set the duration to the 'quarterDuration' parameter.
  9. Add a 'move' block to the 'do together' block and set the direction to 'FORWARD' and the amount to 0.5 (keep the calling object as 'this'). Click 'add detail' and set the duration to the 'quarterDuration' parameter.
  10. From the statements library, add another 'do together' block after the existing one in the code area.
  11. Inside the new 'do together' block, add a 'move' block with the direction set to 'FORWARD' and the amount to 0.5. Click 'add detail' and set the duration to the 'quarterDuration' parameter.
  12. From the procedures palette, add a 'straightenOutJoints' block to the second 'do together' block. Click 'add detail' and set the duration to 'quarterDuration'.
  13. From the statements library, add another 'do together' block after the existing ones in the code area.

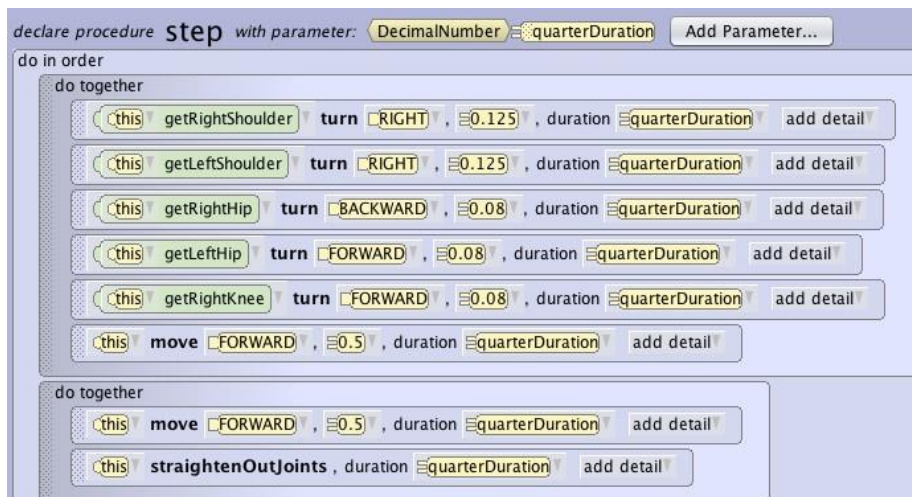


Figure 16: The code to animate the right leg when taking a step

14. Add a 'turn' block to the bottom 'do together' block and set the direction to 'LEFT' and the amount to 0.125. Change the object calling the method to

- 'this.getLeftShoulder'. Click 'add detail' and set the duration to the 'quarterDuration' parameter.
15. Add another 'turn' block to the bottom 'do together' block and set the direction to 'LEFT' and the amount to 0.125. Change the object calling the method to 'this.getLeftShoulder'. Click 'add detail' and set the duration to the 'quarterDuration' parameter.
  16. Add another 'turn' block to the bottom 'do together' block and set the direction to 'FORWARD' and the amount to 'Custom DecimalNumber'. In the pop-up window, set the value to 0.08 and then change the object calling the method to 'this.getRightHip'. Click 'add detail' and set the duration to the 'quarterDuration' parameter.
  17. Add another 'turn' block to the bottom 'do together' block and set the direction to 'BACKWARD' and the amount to 'Custom DecimalNumber'. In the pop-up window, set the value to 0.08 and then change the object calling the method to 'this.getLeftHip'. Click 'add detail' and set the duration to the 'quarterDuration' parameter.
  18. Add another 'turn' block to the bottom 'do together' block and set the direction to 'FORWARD' and the amount to 'Custom DecimalNumber'. In the pop-up window, set the value to 0.08 and then change the object calling the method to 'this.getLeftKnee'. Click 'add detail' and set the duration to the 'quarterDuration' parameter.
  19. Add a 'move' block to the bottom 'do together' block and set the direction to 'FORWARD' and the amount to 0.5 (keep the calling object as 'this'). Click 'add detail' and set the duration to the 'quarterDuration' parameter.
  20. Right-click on the second 'do together' block (containing only 2 blocks) and select 'Copy to Clipboard'. (*Note: The normal keyboard commands for copying and pasting are not available.*)
  21. Click and drag from the clipboard in the top right corner to paste the block at the bottom of the existing code. The final code for the 'step' procedure should look like Figure 17.

```

declare procedure step with parameter: DecimalNumber quarterDuration Add Parameter...
do in order
do together
  (this getRightShoulder turn RIGHT, 0.125, duration quarterDuration add detail
  (this getLeftShoulder turn RIGHT, 0.125, duration quarterDuration add detail
  (this getRightHip turn BACKWARD, 0.08, duration quarterDuration add detail
  (this getLeftHip turn FORWARD, 0.08, duration quarterDuration add detail
  (this getRightKnee turn FORWARD, 0.08, duration quarterDuration add detail
  (this move FORWARD, 0.5, duration quarterDuration add detail
do together
  (this move FORWARD, 0.5, duration quarterDuration add detail
  (this straightenOutJoints, duration quarterDuration add detail
do together
  (this getRightShoulder turn LEFT, 0.125, duration quarterDuration add detail
  (this getLeftShoulder turn LEFT, 0.125, duration quarterDuration add detail
  (this getRightHip turn FORWARD, 0.08, duration quarterDuration add detail
  (this getLeftHip turn BACKWARD, 0.08, duration quarterDuration add detail
  (this getLeftKnee turn FORWARD, 0.08, duration quarterDuration add detail
  (this move FORWARD, 0.5, duration quarterDuration add detail
do together
  (this move FORWARD, 0.5, duration quarterDuration add detail
  (this straightenOutJoints, duration quarterDuration add detail

```

Figure 17: The complete code for the 'step' procedure

### 3.2 Creating the 'walk' procedure

1. Click the hexagonal class selector button and select 'Biped' > 'Add BipedProcedure...' and name the procedure 'walk'.
2. With the 'walk' tab selected, click 'Add Parameter...' and in the pop-up window, set the value type to 'WholeNumber' and the name to 'steps', then press 'OK'. This will be the number of steps to take.
3. Click 'Add Parameter...' again and in the pop-up window, set the valuetype to 'DecimalNumber' and the name to 'stepDuration', then press 'OK'. This will be the duration per step.

4. From the statements library, drag a 'count' block into the code area and set the number to count up to 'steps'.
5. From the procedures palette, place a 'step' statement inside the 'count' block. Set the duration to the 'stepDuration' parameter (the bottom element of the dropdown menu).

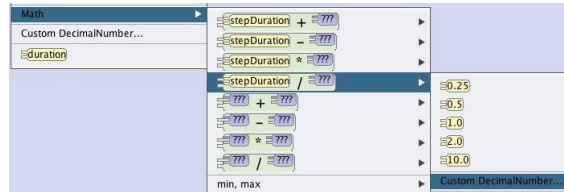


Figure 18: Divide the 'duration' by four to get the 'stepDuration'

6. Click on the duration parameter of the 'step' statement and select 'Math' > 'stepDuration / ???' > 'Custom DecimalNumber'. In the pop-up window, enter 4.0. This is required because the duration used in the 'step' block is one-fourth of the total actual duration of the 'step' block due to the four 'do together' blocks in the 'step' procedure.

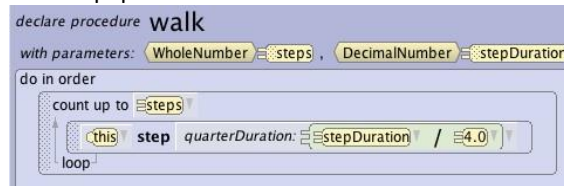


Figure 19: The complete code for the 'walk' procedure'

7. In the procedure tabs, select 'myFirstMethod'.
8. Right-click on the this.alien 'move' block inside the first 'do together' block and delete it.
9. With 'this.alien' selected in the object selector dropdown, drag a walk block from the procedure palette into the first 'do together' block where the 'move' block was. Set the number of steps to 3 and the duration to 'Custom DecimalNumber'. In the pop-up window, enter 0.14.
10. Right-click on the this.alien 'move' block inside the last 'do together' block and delete it.
11. With 'this.alien' selected in the object selector dropdown, drag a walk block from the procedure palette into the last 'do together' block where the 'move' block was. Set the number of steps to 11 and the duration to 'Custom DecimalNumber'. In the pop-up window, enter 0.1.



```

declare procedure myFirstMethod
do in order
  (this.camera) moveAndOrientTo (this.bunkerFront) add detail
  (this.alien) turnToFace (this.alienRobot) add detail
  (this.alien) think "Oh no..." add detail
  (this.alien) delay 0.25
  (this.alien) turn LEFT, 0.5 add detail
do together
  (this.alien) walk steps: 3, stepDuration: 0.14
  (this.alienRobot) turnToFace (this.alien), duration 0.5 add detail
  (this.alienRobot) moveToward (this.alien), 11.0 add detail
do together
  (this.camera) moveAndOrientTo (this.bunkerRight) add detail
  (this.alien) turn LEFT, 0.25, duration 0.25 add detail
do together
  (this.alienRobot) move RIGHT, 1.0, duration 0.5 add detail
  (this.alienRobot) turn LEFT, 0.25, duration 0.5 add detail
  (this.alien) move FORWARD, 1.0, duration 0.5 add detail
do together
  (this.alien) walk steps: 11, stepDuration: 0.1
  (this.alienRobot) move FORWARD, 22.0, duration 2.0 add detail

```

Figure 20: The final code for the 'myFirstMethod' procedure

- Run the scene and the alien should now be running away from the robot around the building.