ORACLE ACADEMY

Tutorial 3: Use Procedures and Arguments, Add Rotation and Randomization

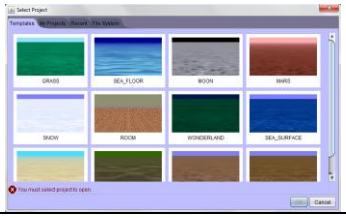
Overview	In this tutorial, you will practice using procedures and arguments to create an animation. You will practice rotating an object's subparts and invoking random movement to make an object's movements less predictable.
Key Concepts Learned	• Use procedures to move objects • Add Java programming procedures to the Code editor • Demonstrate how procedure values can be altered • Create programming comments • Reorder, edit, delete, copy, and disable programming statements • Test and debug an animation • Add a control statement to the Code editor • Use random numbers to randomize motion
Difficulty Level	Beginner: This tutorial is appropriate for someone who has used Alice 3 to: • Add multiple objects to a scene • Code a simple programming instruction • Use precise positioning and drag-and-drop positioning • Use a one-shot procedure to precisely position an object in a scene • Position the sub-parts of an object in the Scene editor
Duration	1 hour
Notes	This tutorial was built using Alice 3.1.81.

Part 1: Define the Scenario

Review the scenario and corresponding animation. In this tutorial, you are going to create an animation from the scenario defined below.

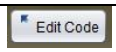
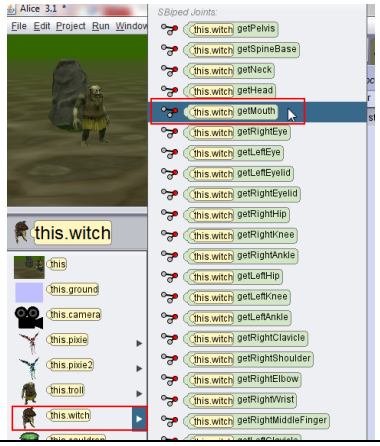
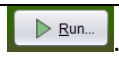
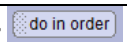
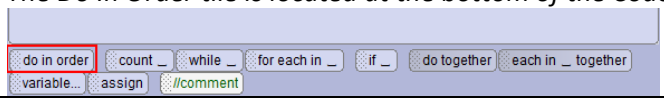
Scenario	**Animation**
A witch casts a spell on a troll.	A witch casts a spell on a troll using one of her potions. The troll doubles in size.

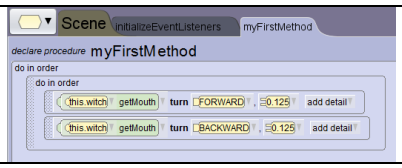
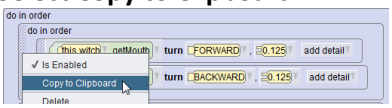
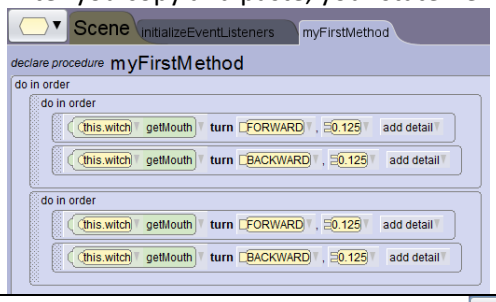
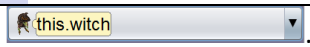
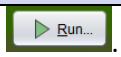
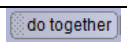
Part 2: Add Objects to the Scene

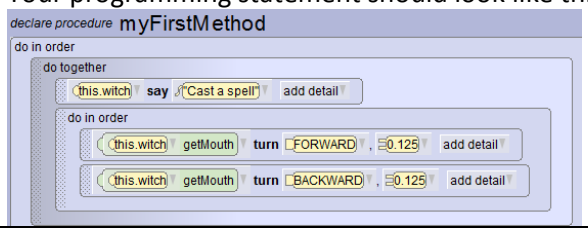
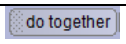
1.	Open the **SWAMP** template. Save the animation project with the file name **Tutorial 3**. REMEMBER: SAVE FREQUENTLY AS YOU WORK! 
2.	Add the following objects to the scene: • Troll • Witch • 2 Pixies • Cauldron • Spell book • Potion • Magic stone

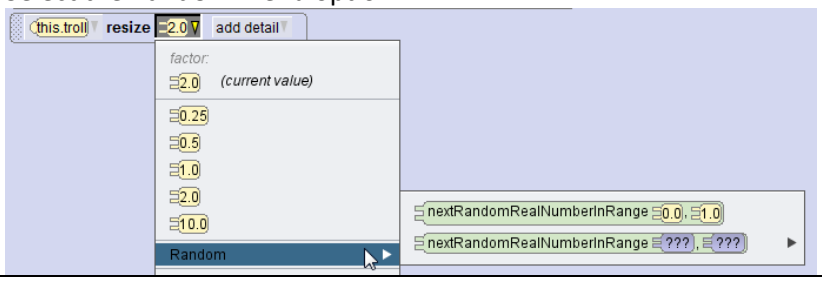
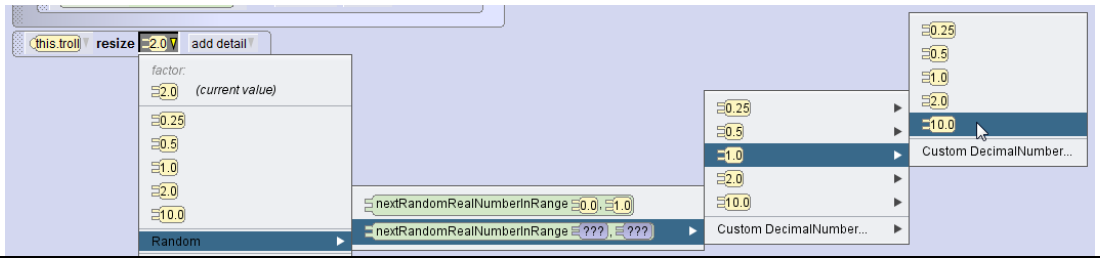
3.	Position the objects in the scene as follows. Notice that some objects were resized, and the spell book is in the witch's hand. When you are finished, your scene should look like this:  Feel free to add additional objects to the scene to make it unique!

Part 3: Program Objects to Move

1.	Click the **Edit Code** button  to go to the Code editor.
2.	Select the Witch's Mouth from the instance menu. 
3.	Drag the **turn** procedure into myFirstMethod. Select arguments: **FORWARD** → **0.125**.
4.	Drag the **turn** procedure into myFirstMethod. Select arguments: **BACKWARD** → **0.125**.
5.	Run the animation  . *Observe how the witch's mouth moves like she is talking.*
6.	Drag a **Do In Order**  control statement into myFirstMethod. *The Do In Order tile is located at the bottom of the Code editor*: 
7.	Drag the **turn** procedures inside of the Do In Order. *This will make it easier to copy this statement so your witch can talk a few times.* *Your programming statement should look like this:*

	
8.	Copy the Do In Order statement and paste it below. To do this: Right-click on the **Do In Order**. Select **Copy to Clipboard.**  Then click on the clipboard and drag the object from the clipboard to **myFirstMethod**. OR Click the **Control key (CTRL)** on your keyboard, then click on the **Do In Order** and drag it below. This will also create a copy. After you copy and paste, your statements should look like this: 
9.	In the Instance menu, select the witch  .
10.	Drag a **say** procedure into the first Do In Order, above the turn procedures. Select **Custom TextString…** Enter text: **Cast a spell**
11.	Drag a **say** procedure into the second Do In Order, above the turn procedures. Select **Custom TextString…** Enter text: **TROLL RESIZE!**
12.	Your programming statements should look like this: 
13.	**Run** the animation  . *You will notice that the witch talks first, then moves her mouth. You want her to move her mouth at the same time she is saying something.*
14.	Drag the first **say** procedure above the Do In Order.
15.	Drag a **Do Together**  to the top of myFirstMethod.
16.	Drag the first **say** procedure (Cast a spell) inside of the Do Together.

17.	Drag the first **Do In Order** inside of the Do Together. Your programming statement should look like this: 
18.	**Run** the animation. *You will notice that the witch talks while moving her mouth.*
19.	Repeat these steps to edit the second set of programming statements so that the witch says "TROLL RESIZE!" while moving her mouth. When you are done, your programming statements should look like this: 
20.	**Run** the animation. *Next, you will program the troll to move his arms, then grow larger.*
21.	Select the **Troll** from the instance menu.
22.	Drag the **say** procedure into myFirstMethod, below the other programming statements. Select **Custom TextString…** Enter text: **Oh no!**
23.	Drag a **Do Together**  into myFirstMethod.
24.	Select the **troll's left shoulder** from the instance menu.
25.	Drag a **roll** procedure into the Do Together. Select arguments: **LEFT** → **0.25**.
26.	Select the **Troll's right shoulder** from the instance menu.
27.	Drag a **roll** procedure into the Do Together. Select arguments: **RIGHT** → **0.25**.
28.	Select the **Troll** from the instance menu.
29.	Drag the **resize** procedure into myFirstMethod. Select argument: **2.0.**
30.	**Run** the animation.
31.	*Next, you will randomize the size of the troll, so that each time the animation is run, the troll becomes a different size.* In the resize procedure, click on the **factor argument** (currently set at 2.0).

	Select the **Random** menu option. 
32.	Select **nextRandomRealNumberInRange ???, ???** Select **1.0** for the first value, then **10.0** for the second value. *This will set the resize factor of the Troll to a random number between 1 and 10 each time the animation is run.* 
33.	**Run** the animation. Continue editing the animation to make it unique!

Part 4: Give the Animation a Unique Ending

Now that the animation is complete, you need to give the animation a unique ending. Here are some ideas to spur your thinking:

- The troll picks up the witch and tosses her around.
- The pixies cast a spell on the witch and make her very small.
- The troll, witch and pixies dance.