ORACLE

Academy

Java Fundamentals

2-6

Control Statements

ORACLE
Academy

Objectives

- This lesson covers the following objectives:
  - Define multiple control statements to control animation timing
  - Create an animation that uses a control statement to control animation timing
  - Recognize programming constructs to invoke simultaneous movement

Arguments

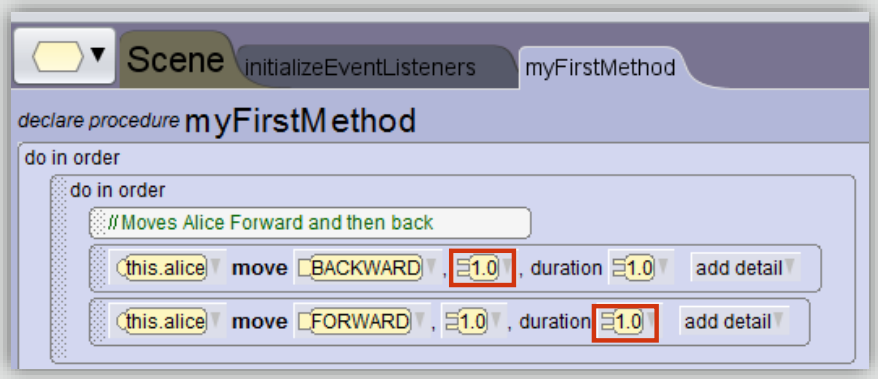
- The arguments of a procedure may be edited or further defined to control object movement and timing
- Examples of Alice 3 arguments include:
  - Object
  - Direction
  - Direction amount
  - Time duration

A computer program requires arguments to tell it how to implement the procedure.

Arguments are information that we pass to a procedure in order to specify how the procedure will execute.

Arguments Display

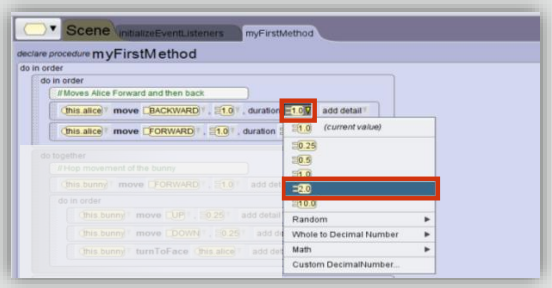
- Below are examples of arguments in a procedure



Arguments are the values provided to the statement to control things like duration and distance.

Steps to Edit Arguments

- Next to the argument's value, click the arrow to display the menu of values
- Select a new value
- The menu indicates the current value, followed by pre-set values to choose from, followed by additional menu options to specify randomization, math calculations, or a custom decimal number



Selecting a Placeholder Argument

- When adding a procedure to the code, you are required to select a value for each argument in the procedure
- Often times you may choose a pre-set value as a placeholder (a temporary value) which is later changed during an edit cycle
- Using a placeholder value is a common approach to creating and refining animation performance

Simultaneous Movements

- To create simultaneous movements for an object, use the Do together control statement

Control Statement	Description
Do In Order	• Default control statement in Code editor • Executes procedures in sequential order
Do Together	• Executes procedures simultaneously • Used for simultaneous movements such as walking and sitting motions

The do in order and do together statements are often nested inside each other to control specific actions on objects.

Simultaneous Movements Example 1

- For example, a movement executed together could be as simple as simultaneously raising both arms of a biped object from a hanging position to a straight arm overhead position

ORACLE Academy

You will discover when working with object movement that there is much more involved in moving objects sub-parts than you first thought.

Simultaneous Movements Example 2

- Another example is a walking motion, which requires simultaneous movement of the hips and shoulders
- To create the walking motion for a biped, use:
  - A series of move, roll, and turn procedures
  - Do together control statements
- Different programming may be required for different objects because no two objects walk the same way

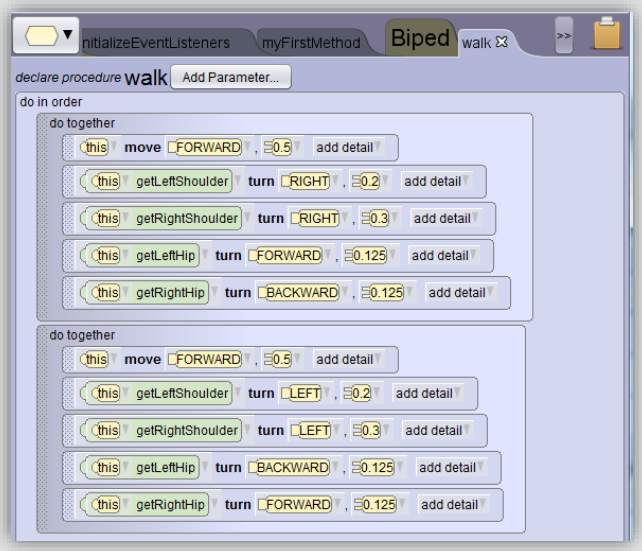
Walking naturally is an incredibly difficult animation to program. There are lots of joints involved as well as lots of different turn and roll statements to consider.

Walking Textual Storyboard Example

Order of Instructions	Programming Instructions
Do Together	Whole body moves forward
	Left shoulder turns right
	Right shoulder turns right
	Left hip turns forward
	Right hip turns backward
Do Together	Whole body moves forward
	Left shoulder turns left
	Right shoulder turns left
	Left hip turns backward
	Right hip turns forward

Walking Motion Example

- Examine this code for a simple walking motion



When Procedures Offset Each Other

- A common mistake is to include two procedures that cancel each other in a Do together construct
- For example, if you include a move up 1 meter procedure, followed by a move down 1 meter procedure in a Do together, nothing will happen
- The procedures cancel each other out

Remember you can't move in two directions at once. It may appear logically correct when you read it but when placed in a do together statement it becomes impossible to execute the command.

setVehicle Procedure

- The setVehicle procedure employs the concept of a rider object and a vehicle object
- The rider object is selected when the setVehicle procedure is used to specify the vehicle for the rider
- Then, when the vehicle object is programmed to move, the rider object will automatically move with it
- Examples:
  - Person rides a camel or horse
  - Camera follows a helicopter to shoot the scene from the helicopter's point of view

Setting one object's vehicle property to that of another allows you to move multiple objects at the same time, while only having to create the code for a single object.

setVehicle Procedure Example 1

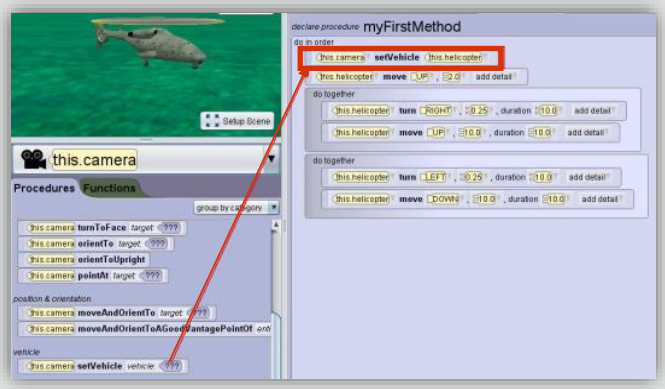
- The child is positioned on the camel in the Scene editor
- Then, the camel is set as the vehicle of the child in the Code editor
- When the camel moves, the child stays on top and moves with the camel

setVehicle Procedure Example 2

- The helicopter is set as the vehicle of the camera in the Code editor
- When the helicopter moves, the camera films the scene from the helicopter's perspective

Steps to Use setVehicle Procedure

- Determine the vehicle object and the rider object
- In the Code editor, select the rider object from the Instance menu
- From the Procedures tab, drag the setVehicle procedure into the Code editor
- In the procedure, select the vehicle object from the menu



Steps to Stop setVehicle Procedure

- If you want the rider object to get off the vehicle object, drag another setVehicle procedure into the Code editor at the point the rider should get off the vehicle
- Set the vehicle to this, which sets the vehicle of the rider back to the scene

It is important to remember to always reset the vehicle properties once the section of the animation is complete.

Terminology

- Key terms used in this lesson included:
  - Arguments
  - Do together control statement
  - Do in order control statement

Summary

- In this lesson, you should have learned how to:
  - Define multiple control statements to control animation timing
  - Create an animation that uses a control statement to control animation timing
  - Recognize programming constructs to invoke simultaneous movement


ORACLE
Academy