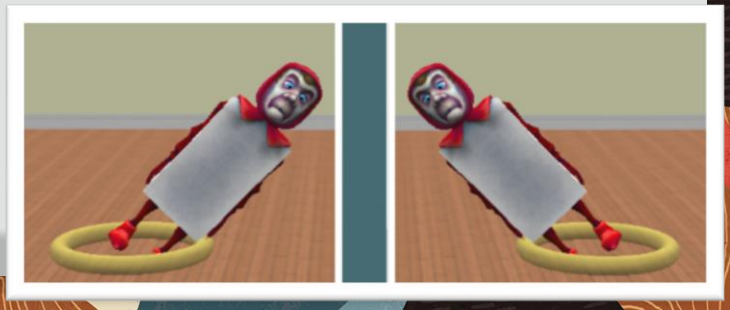
ORACLE

Academy

# Java Fundamentals

## 2-4

## Rotation and Randomization



ORACLE
Academy

# Objectives

- This lesson covers the following objectives:
  - Correlate storyboard statements with program execution tasks
  - Add a control statement to the Code editor
  - Use random numbers to randomize motion

# Program Development Using a Top-Down Approach

- Just as a homebuilder uses a set of blueprints and follows a series of steps, a programmer uses a plan and follows a series of steps to build a program
- Homebuilders work with a purpose and set specific goals as they build—the rooms are energy efficient, the home meets building codes, etc.

# Program Development Using a Top-Down Approach

- Programmer's also work with a purpose and set goals, because without them, success cannot be measured
- Another term for the goal or purpose of an animation is a high-level scenario—a story with a purpose

## Steps to Use a Top-Down Approach to Programming

- Define the high-level scenario (the purpose/goal of the program)
- Document the actions step-by-step in a textual storyboard
- This helps to gain a thorough understanding of the actions that need to be programmed

# Steps to Use a Top-Down Approach to Programming

- Create a table to align the storyboard steps to the programming instructions
- Review the table during the animation's development to ensure the animation meets specifications
- Revise the table as necessary

# Textual Storyboard Example

- Program the following actions in order:
  - Bunny walks in
  - Alice turns to look at the Bunny
  - Bunny says, "**Is this the party!**"
  - Alice says, "**Oh no!**"
  - Bunny turns to left
- Program the following actions together:
  - Bunny walks away quickly
  - Alice shakes her head


ORACLE Academy





The more exact you make your storyboard the easier coding will be. You should make sure that you have detailed each individual step of the scenario.

# Align Storyboard to Programming Instructions

| Order of Instructions | Storyboard Action | Programming Instructions |
|---|---|---|
| Do in order | Bunny walks in | Bunny moves forward 2 meters |
| | Alice turns to look at the Bunny | Alice turns to face the bunny |
| | Bunny says, "**Is this the party!**" | Bunny says "**Is this the party!**" |
| | Alice says, "**Oh no!**" | Alice says, "**Oh no!**" |
| | Bunny turns to left | Bunny turns left 025 |
| Do together | Bunny walks away quickly<br>Alice shakes her head | Bunny moves forward 4meters in 1 second<br>Alice turns her head left right and left again |





This is an example of how the actions defined in the storyboard relate directly to programming instructions.

# Object Movement

- Object movement is egocentric:
  - Objects move based on the direction they face
- An object can move in six directions:
  - Up
  - Down
  - Forward
  - Backward
  - Right
  - Left

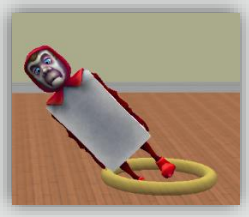
# Examples of Rotation Procedures

| Procedure | Description |
|---|---|
| Turn | Rotates an object left, right, forward, or backward on its center point<br>Left and right turn on the object's vertical axis;  forward and backward turn on the object's horizontal axis |
| Roll | Rolls an object left or right on its center point using the object's horizontal axis |

# Comparison of Turn and Roll Procedures

The object turning left and right on its center point using a vertical axis

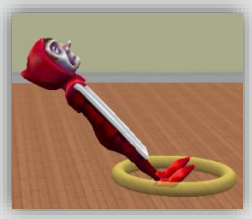
The object rolling left and right on its center point using a horizontal axis



The object turning forward on its center point using a horizontal axis

(The card was already facing left Turning forward caused it to lean forward)

The object turning backwards on its center point using a horizontal axis



(The card was facing left Turning backward caused it to lean backward)

ORACLE Academy







It would be useful at this point to create a simple animation containing a single character and test out each of the roll and turn procedures.

# Sub-Part Rotation Example

- Some objects have moveable sub-parts that can turn and roll
- The pocket watch has moveable hour and minute hands that can roll on the clock's center point
- The key to successful rotation is knowing the center point of an object







Having sub-parts of scenery objects move can make the animation much more realistic.

# Steps to Program an Object to Rotate

- Select the instance of the object to rotate
- Drag a turn or roll procedure to the Code editor
- Set the direction argument
- Set the amount argument (1.0 = one full turn or roll)

Note: A green position indicator line will appear to help you align the programming statement in the desired position


ORACLE Academy







When rotating an object think about how the sub-parts would move as part of the rotation.

# Sub-Part Rotation

- Some objects have moveable sub-parts
- For example, the clock's hands can turn or roll
- Rotation can be applied to an entire object, or select sub-parts of the object
- An object's sub-part displays rings that show its range of motion

ORACLE Academy

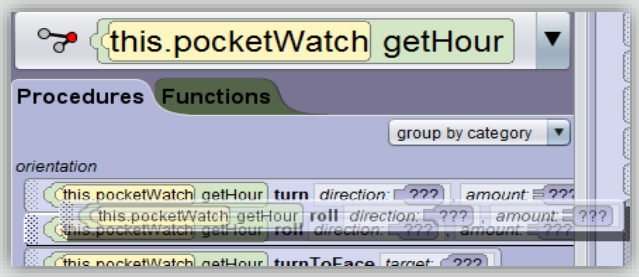




You can use the rings to manually move the sub-parts to change its initial state.

# Steps to Program an Object's Sub-Part to Rotate

- Select the instance of the object sub-part to rotate
- Drag a turn or roll procedure to the Code editor


- Select the direction value for the direction argument
- Select a rotation value for the amount argument (1.0 = one full turn or roll)

# Control Statements

- Control statements define how programming statements are executed in the program
- myFirstMethod is created with a default Do in order control statement
- Within it, all programming statements execute sequentially by default





These are probably the Alice 3 control statements that you will use the most.

# Control Statements

| Procedure | Description |
|---|---|
| Do in order | Execute the statements contained within the control statement in sequential order |
| Do together | Execute the statements contained within the control statement simultaneously |
| Count | Execute the statements contained within the control statement a specific number of times |
| While | Execute the statements contained within the control statement while a specified condition is true |





These are probably the Alice 3 control statements that you will use the most.

# Control Statements in the Code Editor

- Control statements are located at the bottom of the Code editor
- They represent blocks of code into which individual statements are grouped
- Drag control tiles into the Code editor, and then add statements to them

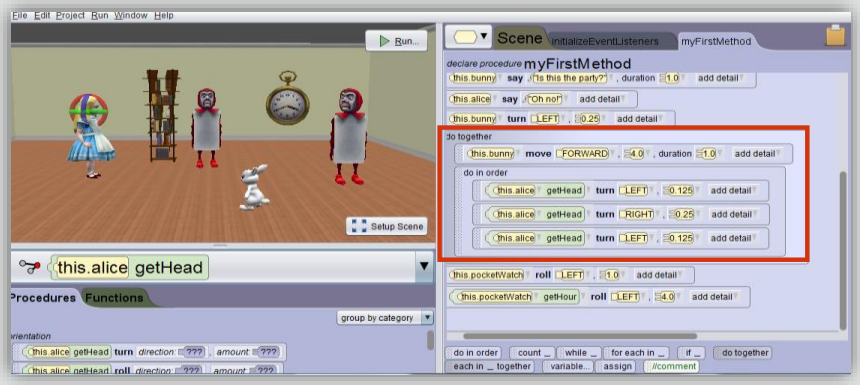





Control statements can be added before or after the code that will be placed within them.

# Add Procedures to a Control Statement

- Drag new or existing programming statements into the Control statement
- In the example illustrated below, the Alice object will move forward, turn left, and roll to the right in order




ORACLE Academy







It is good to add comments explaining the purpose of the code block as well as for the individual lines of code.

# Tip: Create Programming Blocks

- Even though myFirstMethod contains a default Do in order control statement, you can always add other Do in order tiles to the programming area
- This allows you to keep your programming instructions together in organized blocks
- Blocks can be easily copied to the clipboard or moved around in a large program
- Nested Do in order control statements do not impact the performance of the animation; they make program-editing easier for the programmer

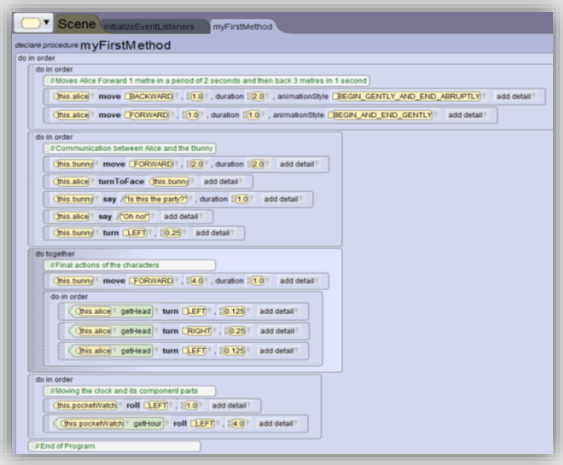




Code blocks should be created around lines of code that carry out a common purpose. If you were making a cat jump you could place all the commands for this action in a code block. This makes it easier to disable code when you are testing, as you only have to disable the code block.

# Nested Do In Order Control Statements

- Here, three Do in order statements and a do together statement are nested inside a fourth Do in order statement
- Nested statements add visual structure to a program, much like an outline brings structure to a report




Nesting is the process of putting one thing inside of another, like eggs lying next to each other in a nest

# Random Numbers

- Random numbers are numbers generated by the computer with no predictable pattern to their sequence
- Random numbers are generated within a given range of numbers
- Computers may require random numbers for:
  - Security:
    - for example, randomly generated passwords
  - Simulation:
    - for example, earth science modeling (i.e., erosion over time)





When using random numbers it is quite normal for the same number to be repeated. This is part of the random behavior.

# Examples of Random Numbers

- Random numbers can be generated within a wide range of numbers, both positive and negative, and may include whole numbers and fractional parts of a number
- Examples of random numbers:
  - 15674
  - -6934022
  - 0.371
  - -89.763

# Random Numbers in Animations

- Animals do not move in straight geometric lines
- They change direction slightly as they walk, swim, and fly
- Random numbers may be utilized in the distance argument of a method so that the object moves in a less predictable, more life-like manner
- Random numbers are often used when creating games that require unpredictable behavior

ORACLE Academy



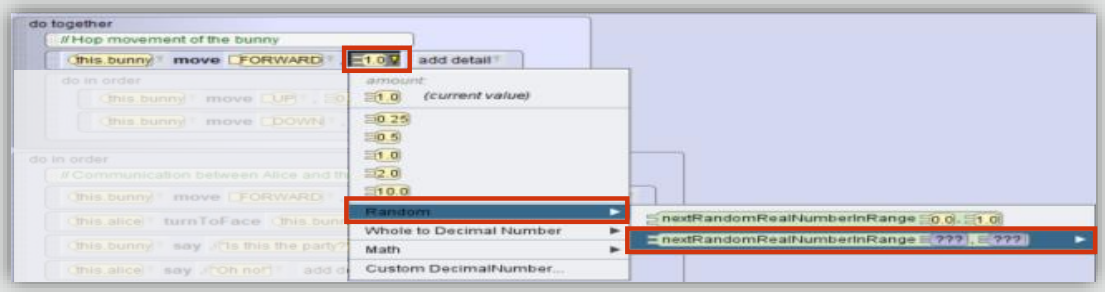




Random behavior adds interest for the user as it means the characters behave differently on every run of the animation.

# Steps to Use Random Numbers

- For any numerical argument, one of the options is a random number
- Select the argument from the drop-down list
  - Select Random
  - Select nextRandomRealNumberInRange
  - Select the low and high values

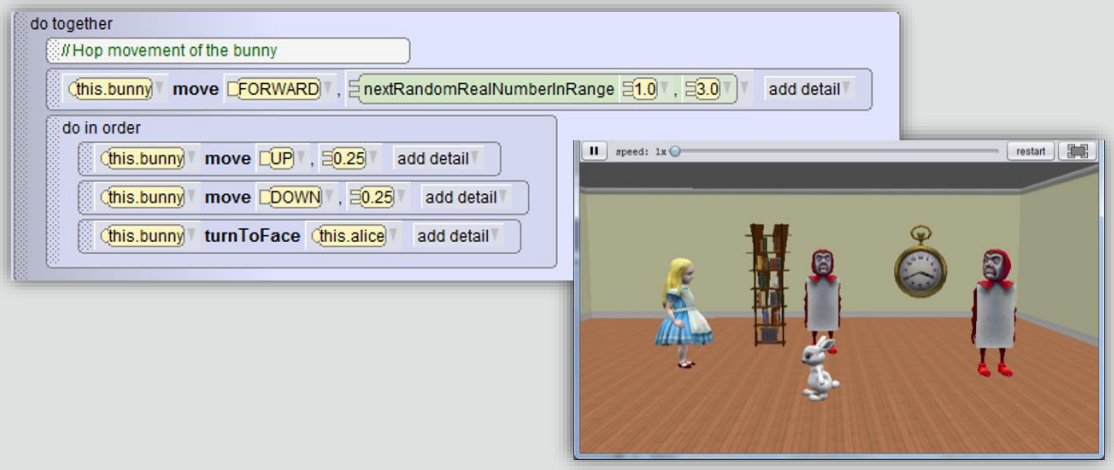


# Steps to Use Random Numbers

- Run the animation
- Alice 3 will randomly generate a value within the selected range when the statement is executed

# Random Number Animation Example

- The bunny moves forward a random distance moving up and down before turning towards Alice at run-time




ORACLE Academy







It can be useful to change the values of the random range and retest the program

# Terminology

- Key terms used in this lesson included:
  - Control statements
  - Nesting
  - Random numbers
  - Textual storyboard

# Summary

- In this lesson, you should have learned how to:
  - Correlate storyboard statements with program execution tasks
  - Add a control statement to the Code editor
  - Use random numbers to randomize motion

ORACLE Academy


ORACLE
Academy