Getting Started with Java Using Alice

Overview

This workshop is designed for students with little or no programming experience and teaches basic Java programming concepts through developing 3-D Animations in Alice 3.1. Alice is a free, educational, introductory Java development environment created at Carnegie Mellon University. While it is suitable for all beginning learners, it can be especially engaging for girls. Students will have fun creating animated stories and games using objects and characters from a rich gallery of 3-D models.

The Alice platform is currently supported in 24 languages, including English, Arabic, Russian, Spanish, and Turkish.

Duration

- Recommended classroom instruction time: up to 8 hours
- Professional education credit hours for educators who complete Oracle Academy training: 8

Target Audiences

Educators

- Secondary school teachers who teach computer programming
- Secondary school teachers who seek to incorporate computing into their curriculum and awaken students' interest in computer science
- May also be suitable for upper-primary school teachers, club leaders, and other general educators who wish to incorporate computer science into their curriculum

Students

- Students who wish to begin to learn about object-oriented programming and Java and to learn how to create animations using fun and engaging tools
- While Alice is suitable for beginning learners of any age, our experience shows it is most engaging for upper primary and early secondary school students

Prerequisites

- None

Suggested Next Courses

- Oracle Academy Workshop – Creating Java Programs Using Greenfoot
- Oracle Academy Course – Java Fundamentals

Lesson-by-Lesson Objectives

Prepare for this Course

- Describe the purpose for using Alice 3 to learn Java
- Check system requirements needed for the course
- Install necessary software for the course

Get Started with Alice 3

- Identify scene components
- Create and save a new project
- Add an object to a scene
- Communicate the value of saving multiple versions of a scene
- Code a simple programming instruction
- Use the copy and undo command
- Understand the value of testing and debugging

Add and Position Objects

- Open a saved version of a project
- Add multiple objects to a scene

- Describe the difference between precise positioning and drag-and-drop positioning
- Use a one-shot procedure to precisely position an object in a scene
- Edit properties of an object in the Scene editor
- Describe three-dimensional positioning axes
- Position the sub-parts of an object in the Scene editor

Use Procedures and Arguments

- Toggle and describe the visual difference between the Scene editor and Code editor
- Locate and describe the purpose of the methods panel and procedures tab
- 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 Rotation and Randomization

- Correlate storyboard statements with program execution tasks
- Add a control statement to the Code Editor
- Use random numbers to randomize motion

Declare Procedures

- Compare and define an animation and a scenario
- Write a storyboard
- Flowchart a storyboard
- Describe inheritance and how traits are passed from superclasses to subclasses
- Describe when to implement procedural abstraction
- Demonstrate how to declare a procedure
- Identify and use procedural abstraction techniques to simplify animation development

Use Control Statements

- 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

Use Functions

- Use functions to control movement based on a return value

Use the IF and WHILE Control Structures

- Use the IF control structure to effect execution of instructions
- Use the WHILE control structure to create a conditional loop for repetitive behavior

Use Expressions

- Create an expression to perform a math operation
- Interpret a math expression

Use Variables

- Understand variables and how they are used in programming

Use Keyboard Controls

- Use keyboard controls to manipulate an animation

Develop a Complete Animation

- Use functional decomposition to write a scenario and storyboard
- Complete an animation
- Test an animation
- Reposition objects at runtime
- Plan a presentation of a completed animation project

Animation Design Worksheet and Academic Examples

- Use an animation design worksheet to complete an animation
- Observe how animations can support a variety of academic topics

To search and register for events scheduled in your area, visit the Academy events calendar.

*Suitable for ages 8-22, Alice is the subject of 11 textbooks. It is taught in thousands of schools globally, including 19% of US colleges and universities. One study shows that exposure to Alice prior to or concurrent with CS1 improved performance in the course by a full letter grade and that middle school girls are perfectly willing to learn to write computer software if you present it as a storytelling activity.