

Getting Started with Java Using Alice

Develop a Complete Animation

Objectives

This lesson covers the following objectives:

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

Animations

An animation is a sequence of actions that simulate movement. Use Alice 3 to construct the scene and write the sequence of actions for your animation, and Alice 3 will render the animation for you.

Rendering is a process where the software program converts your code into the animation that you see. Alice 3 renders the animation based on the instructions provided by the programmer.

Animations Require Planning

Animations can be complex to plan and develop. To simplify and organize this complex task, you can:

- Use a methodical process to identify and solve the problems that arise along the way.
- Break down the development into manageable steps.

Functional decomposition is the methodical process of identifying a complex problem and breaking it down into smaller steps that are easier to manage.

Functional Decomposition Example

Examine this high-level process:

- Consider a math concept that requires many steps.
- Identify the high-level steps for the math concept.
- Further refine and define the low-level tasks needed for each high-level step.

Animation Development Process

The process to develop an animation is similar to the problem-solving process. Compare the two processes below.

Problem-Solving Process



Animation Development Process



Step 1: Define the Scenario

Professional animators begin by developing a scenario—or story—that gives the animation a purpose.

Examples:

- A story that presents a conflict and resolution.
- A lesson that teaches a math concept.
- A simulation that demonstrates a process.
- A game that entertains or trains.

Scenario and Animation Examples

Scenario Type	Scenario	Animation
Problem and a solution.	A cat needs help to get down from a tree.	A firefighter climbs up the tree to save the cat.
Teach a concept.	Memorizing chemistry symbols is difficult.	A timed game matches chemistry symbols with their definitions.
Simulate or demonstrate a process.	A car has a flat tire.	A demonstration shows how to change a tire on a virtual car.
Play a game.	An airplane must avoid objects in its path as it flies through the sky.	An interactive game maneuvers an airplane around objects in the sky.

Step 2: Design a Storyboard

Two types of storyboards are often used to plan an animation:

- **Visual:** A series of illustrated images that represent the main scenes of the animation.
- **Textual:** A detailed, ordered list of actions that each object performs within each scene of the animation.

Storyboard Formats

Develop your visual and textual storyboards using a variety of formats.

Examples:

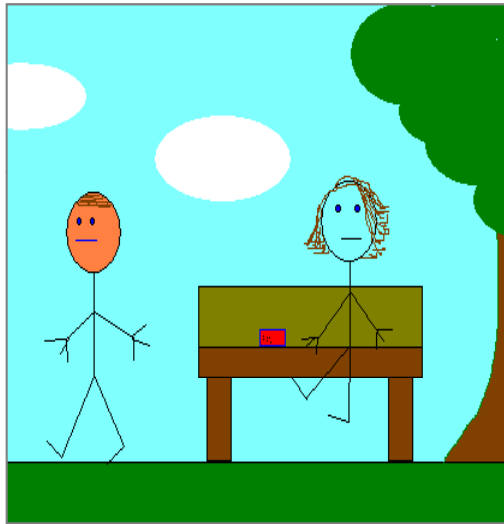
- Draw them on paper with a pencil.
- Create them using digital tools such as a word processor, drawing software, or presentation program.
- Write a textual storyboard within Alice 3 in the Code editor using comments to organize the steps in your program.

Visual Storyboards

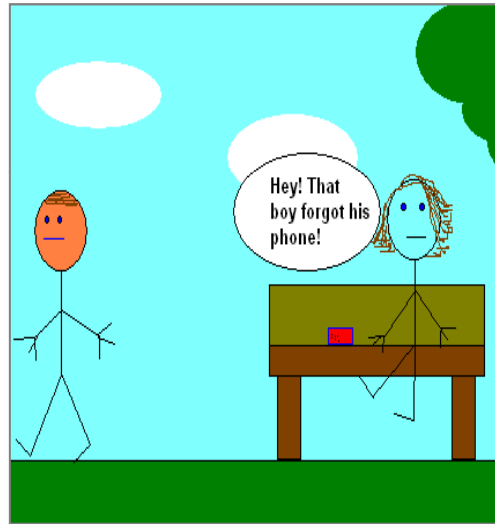
The visual storyboard helps a reader understand:

- The scene components.
- How the initial scene will be set up.
- The moving and non-moving objects in a scene.
- The actions that will take place.
- The user interactions that occur during the animation execution.

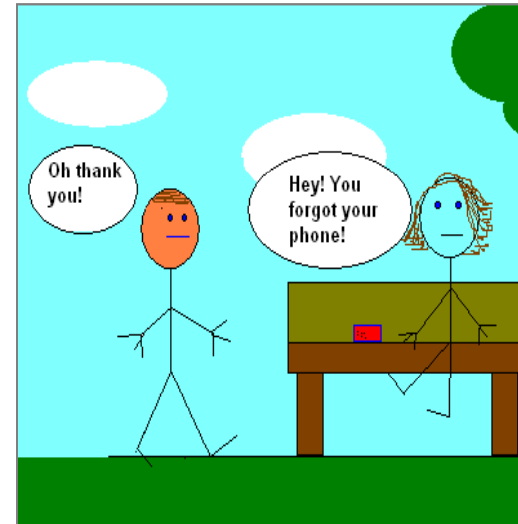
Visual Storyboard Example



Boy and girl sit on a park bench. The boy walks away, leaving his phone behind.



Girl notices the mobile phone. She thinks, "Hey! That boy forgot his phone!"



Girl says out loud, "Hey! You forgot your phone!"
Boy turns around and walks back to bench. He says, "Oh! thank you!"

Textual Storyboards

A textual storyboard helps the reader understand the actions that will take place during the animation.

The moving and non-moving objects can be easily identified within the action statements, but a more detailed description may be necessary if multiple programmers are involved in implementing any scene.

In computing, a textual storyboard is called an algorithm: a list of actions to perform a task or solve a problem.

Textual Storyboard Example 1

Program the following actions in order:

Boy and girl sit on a park bench.

Boy stands up and walks away, leaving his phone on the park bench.

Girl turns to look at the phone.

Girl thinks, “Hey! That boy forgot his phone!”

Girl says out loud, “Hey! You forgot your phone!”

Boy stops and turns around.

Boy walks back to the park bench and says, “Oh! Thank you!”

Textual Storyboard Example 2

This example shows how you can develop your storyboard by first writing comments in the Code editor of your program. Afterwards, you can start to develop the animation directly from the storyboard.



```
Scene initializeEventListeners myFirstMethod
declare procedure myFirstMethod
do in order
  // Boy and girl sit on a park bench.
  // Boy stands up and walks away, leaving his mobile phone on the park bench.
  // Girl turns to look at the phone.
  // Girl thinks "Hey! You forgot your phone!"
  // Boy stops and turns around.
  // Boy walks back to the park bench and says, "Oh! Thank you!"
  // End of program.
```

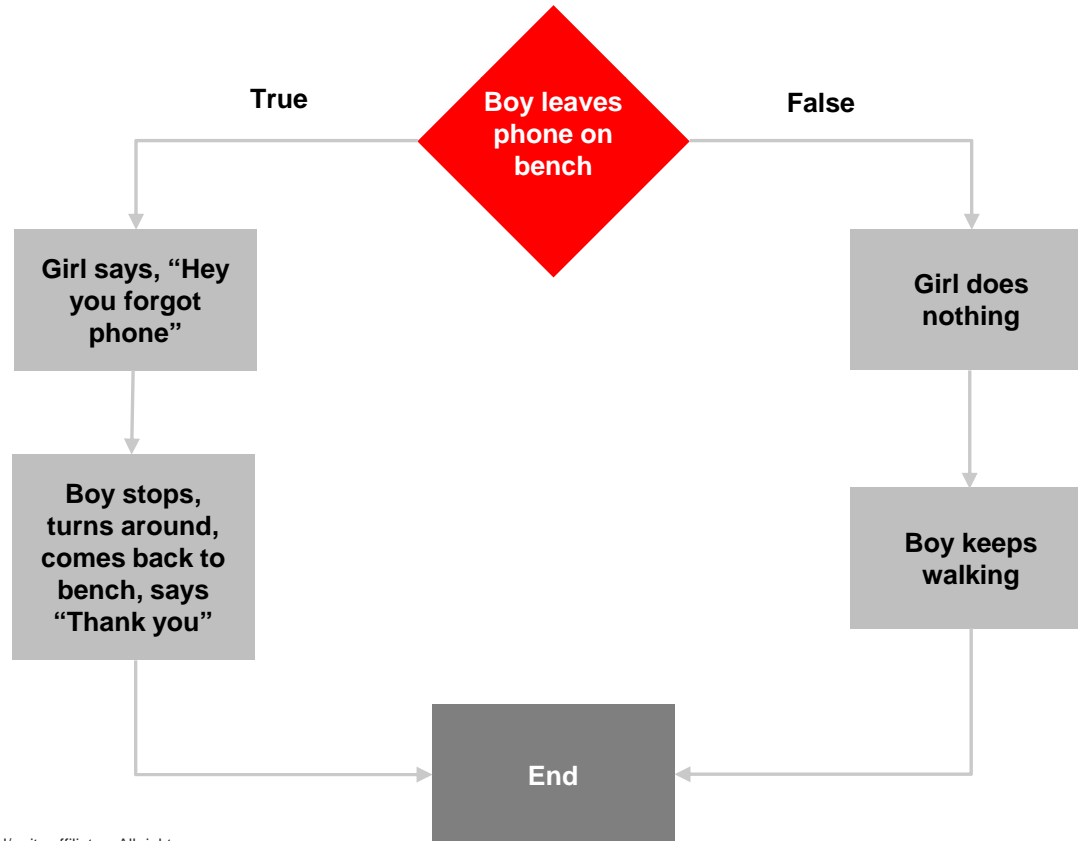
Storyboard Components

Textual storyboards should contain the components listed in the table below to support the program development of the animation.

Component	Definition	Examples
Scene	The place (or “world” in Alice 3) where your story occurs.	Park, library, school, home
Objects	Moving or non-moving characters that you program to move and act.	Animals, cars, people, trees
Actions	Instructions for how each object should act in the scene.	Walk 2 meters, turn left, say “Hello!”
User Interactions	Ways in which the user viewing the animation can manipulate the objects in the animation.	Keyboard commands or mouse clicks to make objects move
Design Specifications	How the objects and scenery should look in the animation.	Size, position, location, color

Storyboard Process Flow

Flowcharting a storyboard helps you organize the flow of actions and conditions within an animation.



How Storyboards are Helpful

There are several ways that storyboards aid in program development:

- Textual storyboards can be used to generate program comment statements and organize program development.
- Storyboards can also help programmers identify repetitive actions, or actions that may be performed by more than one object.

Step 3: Program the Animation

After completing the storyboard, the next step is to program the animation in Alice 3.

- As you program the animation, refer to your storyboard for the animation's design specifications.
- The code written in Alice 3 provides the instructions for the animation's look at run-time.

Animation Checklist

During the animation development process, use this checklist to ensure that your animation meets all animation principles.

Checklist for Animation Completion

Area	Check	<input checked="" type="checkbox"/>
Scenario	Did you clearly define the scenario?	<input type="checkbox"/>
Storyboard	Did you think-through the animation by creating a storyboard?	<input type="checkbox"/>
Textual storyboard	Did you think-through the programming code by creating a textual storyboard?	<input type="checkbox"/>
Program	Did you complete the programming of the animation?	<input type="checkbox"/>

Create a Complete Animation

Creating a complete animation requires a thorough understanding and application of all of the components that you have learned so far.

Completed scene of multiple objects from multiple classes, including props and shapes	Vehicle riding with the setVehicle procedure
Declared procedures	Functions
Movement procedures	IF and WHILE control structures
Object rotation and object sub-part rotation procedures	Random numbers
Simultaneous movement with the Do Together control structure	Math expressions
Variables	Keyboard controls

Step 4: Run the Animation

Run the animation to test that it functions properly and executes the actions planned in the storyboard. This process is often referred to as testing and debugging the software.

Programs can be tested by entering an unintended value in the argument field of a method in an effort to try and “break” the code. When something is broken or doesn't work as you intended in a software program, it is often referred to as a “bug”. Debugging is the process of finding and eliminating bugs in a software program.

Debugging Tasks

Test and debug the animation frequently as it is being developed. Use some of the following debugging techniques:

- Adjust the direction, distance, and duration that objects move.
- Adjust math calculations that help refine the distance or duration that objects move.
- Refine instructions in the code that do not work as intended.
- Resolve errors created by the programmer.

Test Elements of Your Animation

Test every element to prove that it works without error.

- Math expressions calculate as expected.
- Objects move with smooth timing.
- Control structures operate as expected.
- Event listeners trigger the correct responses.

Insure that your code includes comments that clearly identify the purpose or the functionality of blocks of statements in your program. Referring to the comments will help you test and debug your program.

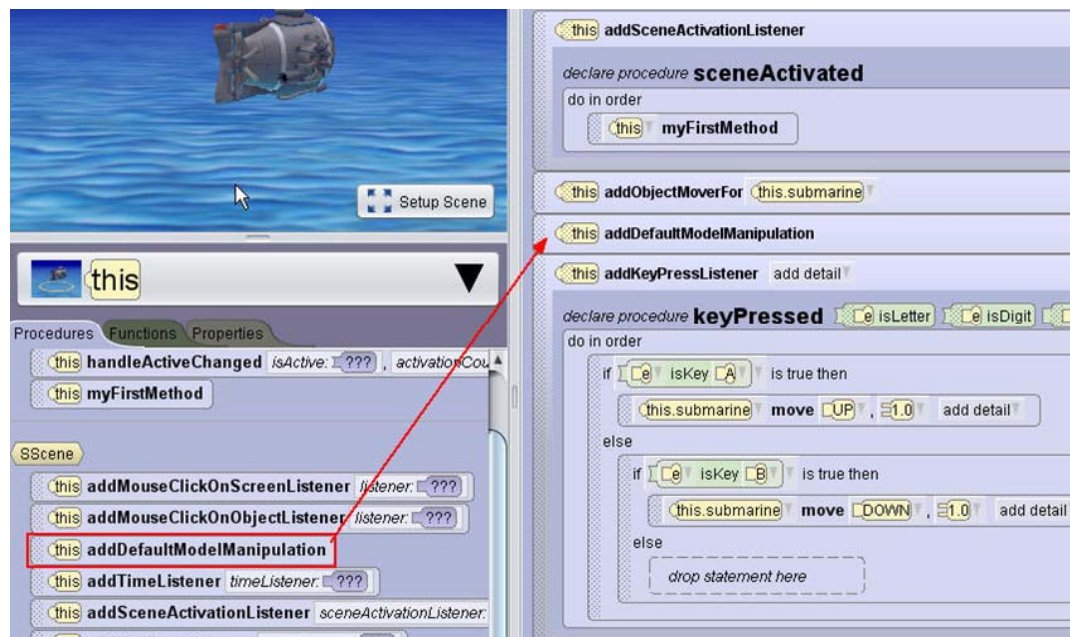
User Input to Reposition Objects at Run-Time

To create an interactive program, a Listener object must be added to the scene.

The `addDefaultModelManipulation` procedure creates a Listener object that targets a mouse-click on any object in the scene and responds by allowing the user to drag that object around the scene while the animation is running.

Steps to Add the addDefaultModelManipulation Procedure

1. Go to the Scene tab.
2. Click the edit button next to initializeEventListeners.
3. Drag the addDefaultModelManipulation procedure (Scene class) into the initializeEventListeners Code editor.



Using the addDefaultModelManipulation Procedure

This procedure allows you to reposition objects at run-time:

- Click and drag the object with your cursor to move it around the scene.
- Press the Control (Ctrl) key, and then click and drag the object with your cursor to turn it right and left.
- Press the Shift key, and then click and drag the object to move it up and down.

Present Your Animation Project

It's time to present your complete animation project. Here are some steps to follow when organizing your animation project presentation:

- Make sure your presentation is thoroughly tested and complete.
- Plan to demonstrate how you used each of the concepts learned in this course.
- Find out how much time you will have for your presentation.
- If it is a group presentation, plan who will do which parts of the presentation.

Prepare a Presentation Outline

Create a presentation outline to plan your complete animation project presentation.

The following slides show an example presentation outline.

Presentation Outline: Section 1

Section 1: Introduction

- Gain your listeners' attention.
- Introduce the theme of the animation project.
- Preview your animation.
- Establish your credibility as a speaker.
- Give the audience a reason to listen to the presentation.

Presentation Outline: Section 2

Section 2: Presentation of animation project

- Organize the presentation in a logical flow.
- Show all capabilities of your animation project.
- Demonstrate how each course concept was used in the animation project.
- Time visual aids to support the spoken presentation.
- Include helpful transitions between ideas.

Presentation Outline: Section 3

Section 3: Conclusion

- Summarize the animation in a memorable way.
- Motivate the audience to respond.
- Provide closure.

Practicing Your Presentation

When rehearsing your presentation:

- Rehearse aloud.
- Time your speech; if it is too long or short, revise it.
- Rehearse standing up.
- Rehearse in front of someone.
- Tape-record or videotape your speech. If you do not have access to recording equipment, practice in front of a mirror.
- Rehearse using visual aids and technology.

Group Presentations

If you will be making your presentation with a group, here are some steps your group can follow to make sure all group members have a part in preparing and delivering the presentation.

- Include all group members in all steps of the presentation planning.
- Divide presentation tasks equally among group members.
- Practice the presentation as a group, and give one another constructive feedback immediately following.

Summary

In this lesson, you should have learned how to:

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