

Alice 3

How-to Guide

(Part 5 - Transfer Alice Project to Java IDE)



Wanda Dann

Don Slater

Laura Paoletti

Dennis Cosgrove

Dave Culyba

Pei Tang

1st Edition: Copyright May, 2012

2nd Edition: Copyright September, 2014

This material may not be copied, duplicated, or reproduced in print, photo, electronic, or any other media without express written permission of the authors and publisher.

Cover artwork by Laura Paoletti, 2012.

Transfer Alice to Java

26. View Alice Code with Java Syntax	176
27. Transfer an Alice Project to a Java IDE	178

26. View Alice Code with Java Syntax

[VIDEO: JAVA DISPLAY VIEW](#)

Side by Side display preference

Alice provides a preference setting for a side-by-side panel display of Alice and Java code. To enable the side-by-side display, select **Preferences/Java Code on the Side** from the **Window** menu, as shown in Figure 26.1. An example of side-by-side Alice and Java code display is shown in Figure 26.2.

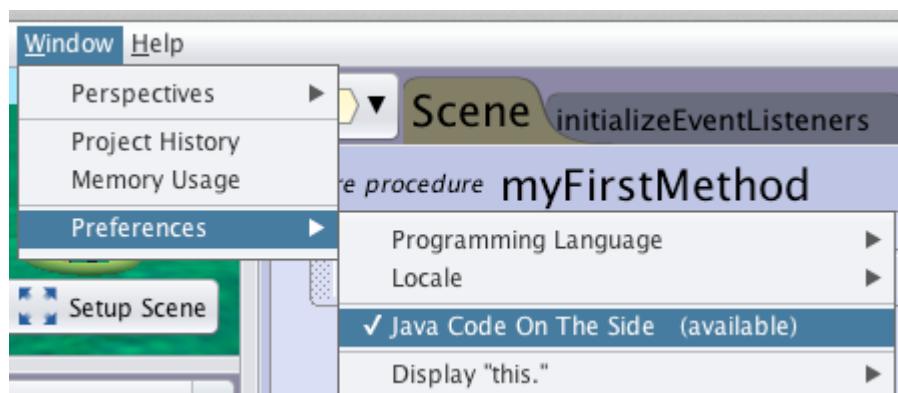


Figure 26.1 Preference setting for Java Code on the Side

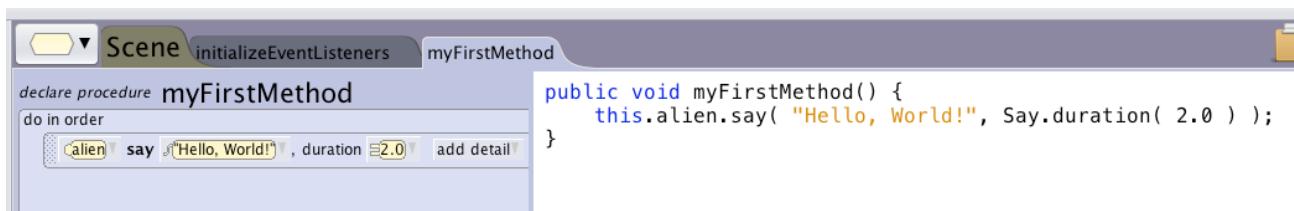


Figure 26.2 Alice and Java side-by-side code panels

In the side-by-side display mode, Alice's drag-and-drop Code editor is fully functional, allowing you to create and modify program code. As changes are made in the Alice code, the Java code display is dynamically updated. The Java code appears very much the same as in any Java IDE,

including all the syntactic features (e.g., curly braces for code blocks). In this mode, the Java code **CANNOT** be directly edited using either drag-and-drop or keyboard entry.

Java language preference

By default, program statements in Alice are displayed using Alice **syntax**. The term syntax refers to the rules of grammar that govern how statements are written. That is, syntax defines the expected ordering of words and punctuation marks for program code. The graphic tiles used in the Alice IDE are used to create program statements with a simple syntax, having a minimal number of quote marks, parentheses, and semicolons.

For those who prefer a “real world” language look and feel, however, Alice provides a preference for changing the IDE display to Java. To change to Java syntax, select **Preferences/Programming Language** in the **Window menu**. Then, in the cascading menu, select **Java**, as shown in Figure 26.3. Program statements in the Code editor will now be displayed with greater fidelity to Java syntax than in the default Alice display.

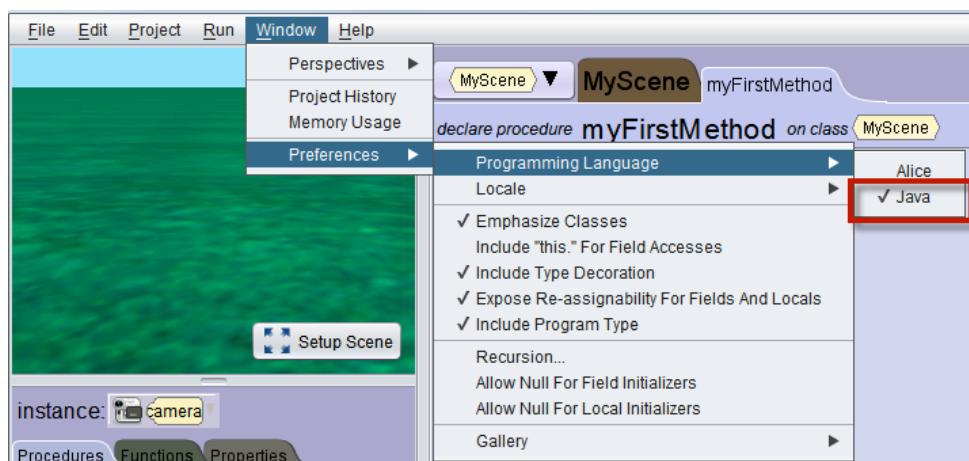


Figure 26.3 Setting the Programming Language Preference to Java

As a comparison, Figures 26.4 and 26.5 show the exact same code in Alice (Figure 26.4) and Java (Figure 26.5). Differences in corresponding statements are highlighted by the green ovals and red boxes in the two figures. The green ovals highlight differences in punctuation marks and the red boxes highlight other differences.

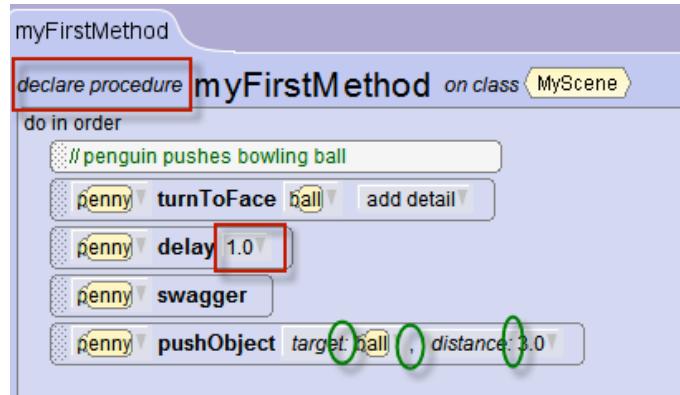


Figure 26.4 Alice code displayed with Alice Programming Language setting

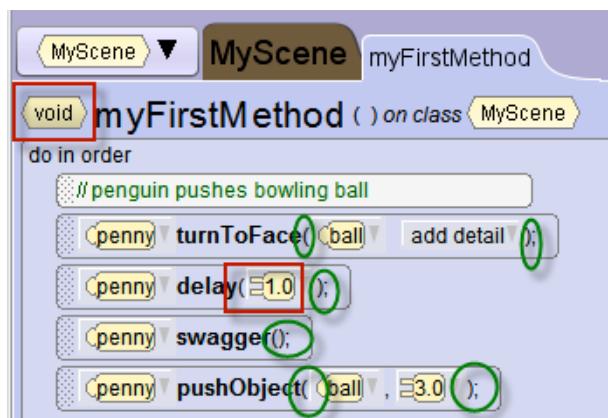


Figure 26.5 Alice code displayed with Java Programming Language setting

In the Java mode, Alice's drag-and-drop Code editor is fully functional, allowing you to create and modify program code. The resulting code is Java code but it **CANNOT** be directly edited using keyboard entry.

27. Transfer an Alice 3 project to NetBeans (Java IDE)

[*Video: From Alice 3 to NetBeans IDE*](#)

For those who want to create an Alice animation project using a text-based editor, Alice provides a plugin for NetBeans, a Java IDE. Using the Alice plugin for NetBeans, you can create a virtual world in Alice, transfer it to NetBeans, and then create code using keyboard entry.

NOTE: In this section, we assume that Java 8, NetBeans, and the Alice Plugin for NetBeans have been installed on your computer. If you have not already done so, please find instructions for download and installation using the following URLs:

- The Alice Plugin Download
 - http://www.alice.org/index.php?page=downloads/download_alice3.1
- Java 8 and NetBeans Download and Installation
 - <http://alice3.pbworks.com/w/page/76386062/java download and installation>
- Alice Plugin Installation
 - <http://alice3.pbworks.com/w/page/57586346/Download and Install Plugin>

Step 1: Save an Alice 3 project

The first step in transferring an Alice project to Java is to save the Alice 3 project before closing Alice. When saving a project, Alice 3 tries to save the file to a default **Alice3/MyProjects** directory, which was automatically created when Alice 3 was installed. It is possible, however, to save the project elsewhere (e.g., on a USB drive or CD). In any case, remember where the project is saved on the computer, as NetBeans will need to locate and open the Alice project during the transfer process.

Step 2: Create a new project in NetBeans

Start NetBeans. (If working on a computer that has limited RAM, we recommend closing Alice before opening NetBeans. Alice does not have to remain open during the transfer process.) In NetBeans, select the File menu and New Project, as shown in Figure 27.1.

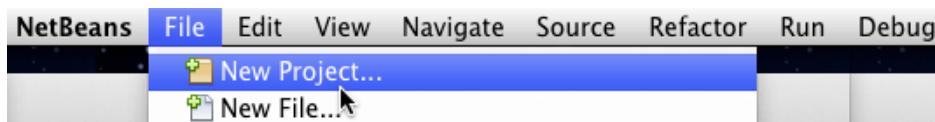


Figure 27.1 Select *New Project...* from the *File* menu in NetBeans

Alternatively, on the NetBeans toolbar, click the New Project icon, as shown in Figure 21.5.

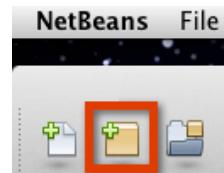


Figure 27.2 *New Project* icon on the *NetBeans* toolbar

A New Project dialog box will open, as shown in Figure 27.3. In the **Categories** box, select **Java**. Then, in the **Projects** box, select **Java Project from Existing Alice Project**. Then, click on **Next**.

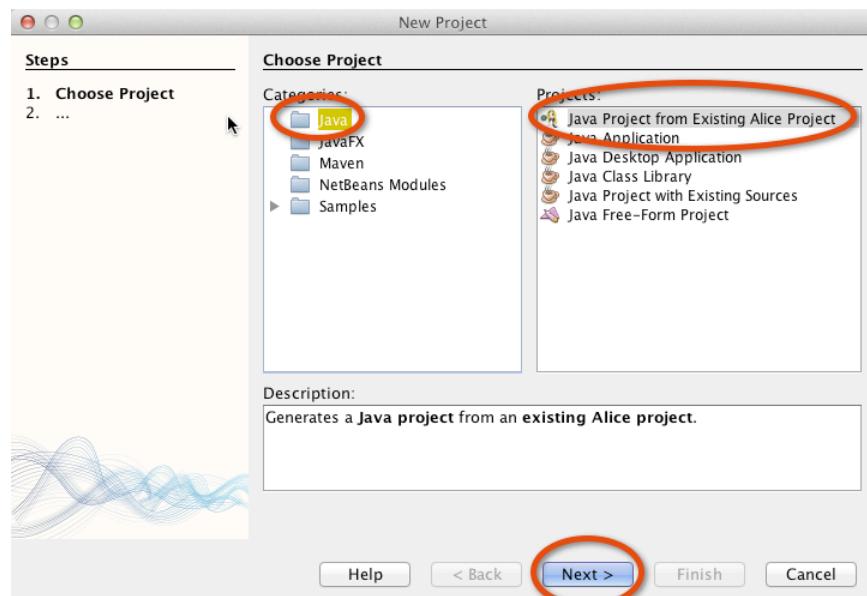


Figure 27.3 New Project dialog box

Step 3: Select an Alice 3 project to import

A selection box will pop up with a prompt to select an Alice project, as shown in Figure 27.4. Click on the **Browse** button.

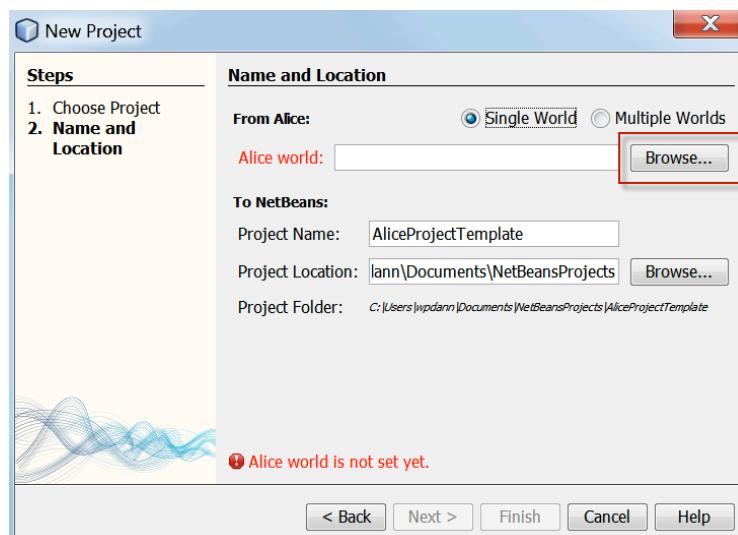


Figure 27.4 Selection box for selecting an Alice world

A navigation box, named **Select Alice World to Import**, will open for browsing to the location where the Alice project has been saved. As shown in Figure 27.5, NetBeans assumes that the Alice world will be in the default directory (**Projects** in the **Alice3** directory). If the desired

project is in the Projects folder, select the world by a single-click on it and then click the Choose button. In Figure 27.5, we selected an example Alice 3 world named *PenguinBowling.a3p* in the Projects folder. If the desired project is not in the Projects directory but is saved elsewhere on the computer, then use the navigation box to locate the directory in which it was saved.

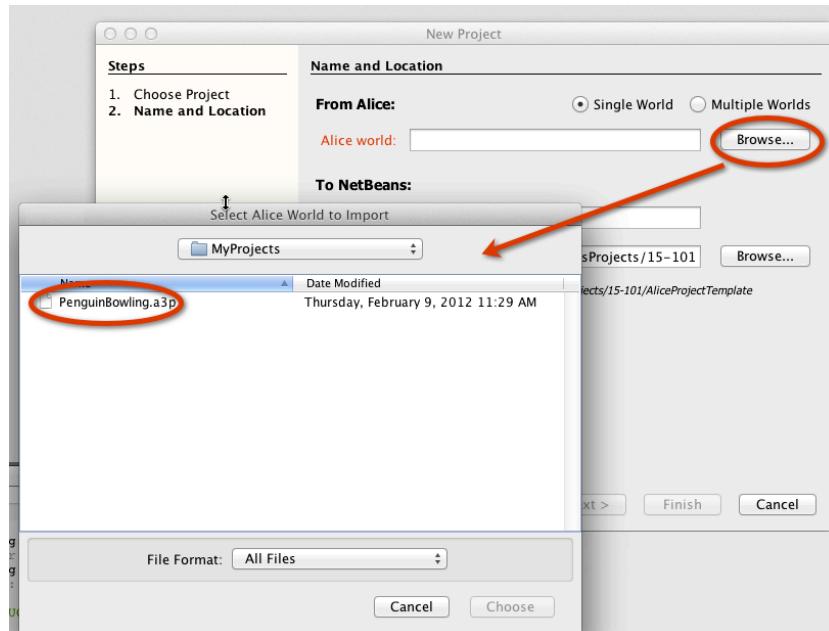


Figure 27.5 Navigation box to locate and choose the Alice project

Once the desired Alice world has been located and the Choose button clicked, the navigation box closes and focus returns to the selection box.

Step 4: Transfer

When an Alice 3 project has been successfully selected in the New Project dialog box, click the **Finish** button, as shown in Figure 27.6. Patiently wait for the transfer to occur. (Transfer may take as few as 10 seconds or more than 30 seconds, depending on the size of the Alice 3 world.)

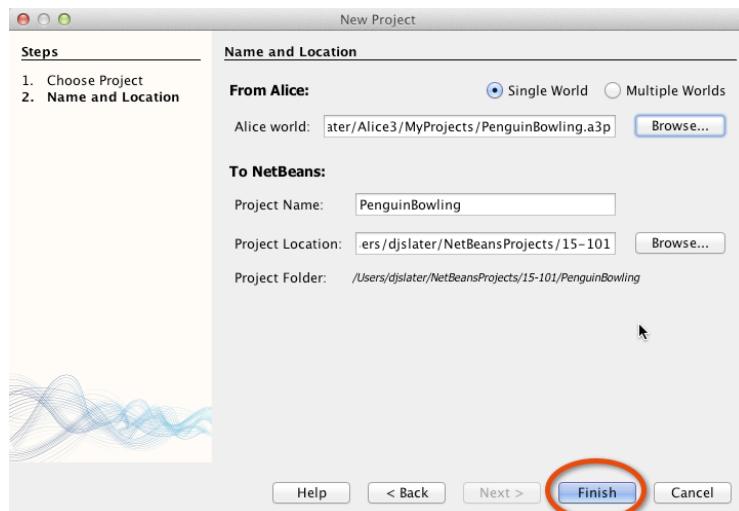


Figure 27.6 Click Finish to begin the transfer

After a successful transfer of an Alice 3 project to a NetBeans project, the new project will be listed in the Projects window in the upper left corner of the NetBeans window, as shown in Figure 27.8. In this example, the **PenguinBowling** project was transferred. Notice, however, that other projects (**SharkAndClownFinshA3Soln** and **PhilosopherSoln**) are also listed in Figure 27.8. As new projects are created in NetBeans, each new project is listed in the NetBeans Projects window along with any previously created projects.

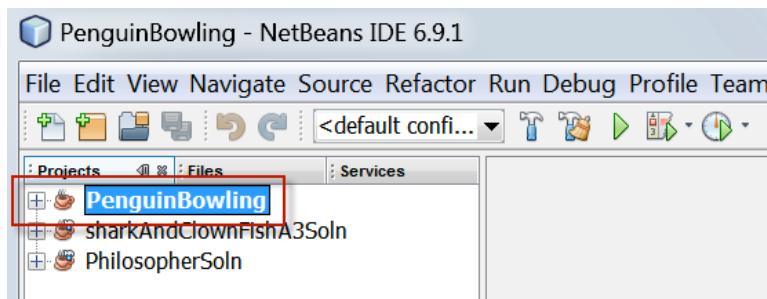


Figure 27.8 A list of projects in NetBeans' Projects window

An Alice project is transferred to NetBeans as a [copy of the original Alice 3 project](#) and is modified as needed to create a new NetBeans project. We refer to the new project as a **Java-Alice3 project**. The original Alice project file still exists but will *not be affected by any changes made in NetBeans* because the changes are only made to the Java-Alice3 project. Also, a Java-Alice3 project cannot be exported back to Alice.

Once an Alice project is transferred to NetBeans, you can view the list of classes in the project by expanding the Project's Source Package, as shown in Figure 27.9.

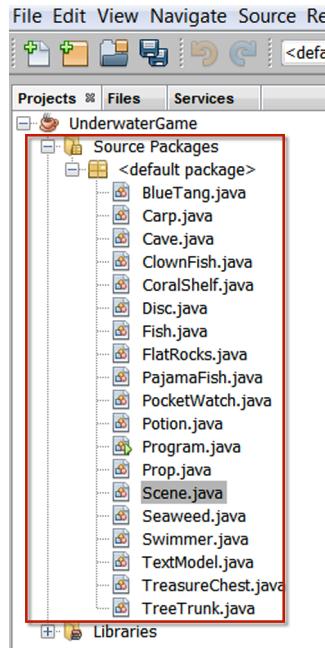


Figure 27.9 Classes in the default Project Source Package

To view the Scene class, double click on Scene.java in the list of classes, as shown in Figure 27.10.

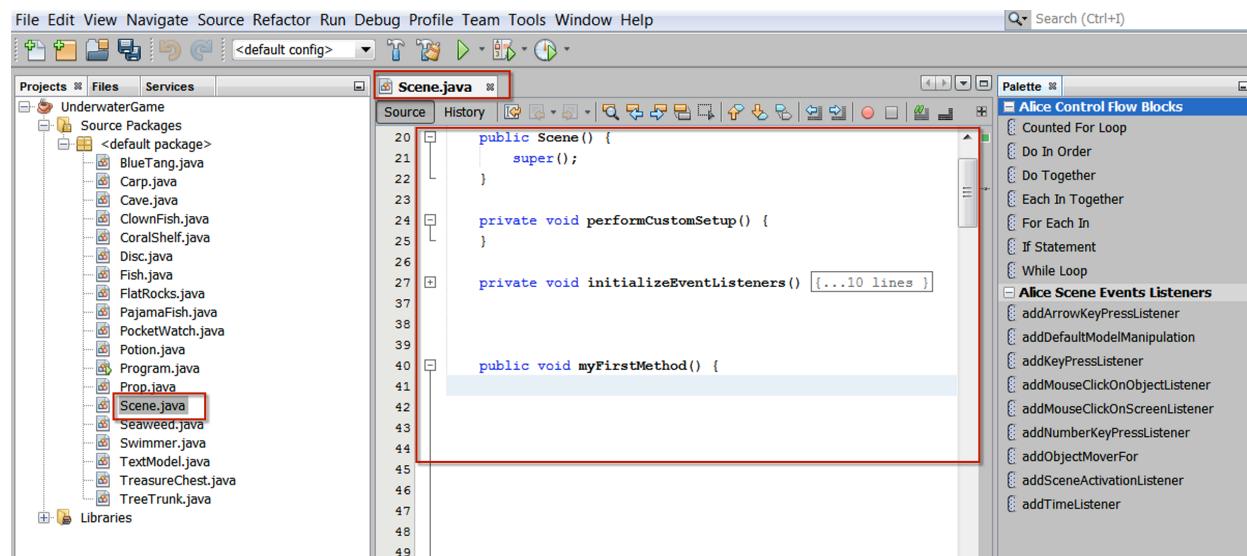


Figure 27.10 View the Scene.java class

On the far right of the IDE is a Palette menu that provides a set of buttons that can be pulled into the editor to create code blocks, as shown in Figure 27.11. The code block contains the appropriate curly braces and a comment,

```
//TODO: Code goes here.
```

The intention is to prompt the programmer to replace the TODO comment with appropriate code.

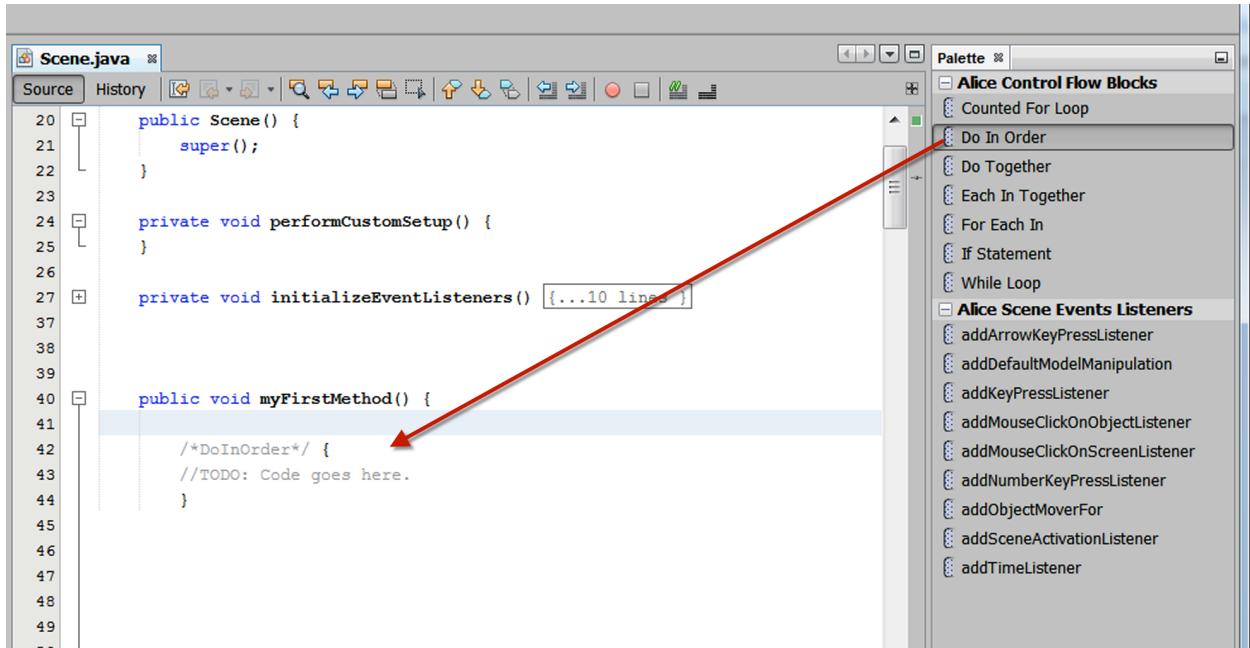


Figure 27.11 Using the Palette to create a code block in the text editor