Alice 3

How-to Guide (Part 1 - Getting Started)

Wanda Dann
Don Slater
Laura Paoletti

Dennis Cosgrove
Dave Culyba
Pei Tang

Cover artwork by Laura Paoletti, 2012.

Preface

Welcome to Alice 3. Alice 3 has been under development since late 2007. A Beta version was made available for adventuresome souls in 2009. This guide has been prepared for release in-sync with the first official (non-beta) release in 2012. This publication also marks the 5th anniversary of the Last Lecture presented by Dr. Randy Pausch, the founder of the Alice Project at Carnegie Mellon University.

The Alice Team

The Alice team consists of a group of software engineers, character artists, professors, and authors. A proud distinction of this team is the devotion each team member has for Alice. The life and breath of Alice software is dependent on the members of our creative and energetic development team:

Dennis Cosgrove, Lead architect and Senior Software Engineer

Dave Culyba, Software Engineer

Matthew May, Junior Software Engineer

Laura Paoletti, Character Artist

Pei Tang, Character Artist

The instructional support materials, including this How-To guide are prepared and tested by members of our authoring and curriculum team:

Wanda Dann, Carnegie Mellon University, wpdann@cs.cmu.edu

Don Slater, Carnegie Mellon University, dslater@cmu.edu

Acknowledgements

The Oracle Foundation, the Sun Microsystems Foundation, the Hearst Foundation, and Electronic Arts have contributed support for the development of the Alice 3 system, for which we are deeply grateful.

The content in this guide is based upon work partially supported by the National Science Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Our deep gratitude goes to early testers and users of Alice 3 for their helpful comments and suggestions: Daniel Green (Oracle), Caron Newman (Oracle Academy), Susan Rodger (Duke University), Pam Lawhead (University of Mississippi), Leslie Spivey (Edison College), William McKenzie (Roger Williams University), Bill Taylor, Anita Wright, and Rose Mary Boiano (Camden County College), Tebring Daly (Collin College), Eileen Wrigley and Don Smith (Community College of Allegheny County).

Community

We are proud to recommend the Alice Educator's listserv as a community for sharing questions and answers. The listserv is monitored and restricted to instructors. A link for subscribing to the Educator's listserv is available at: www.alice.org

As always, we welcome your comments and suggestions.

The Alice Team

Exploring Alice 3

1. HOW TO INSTALL AND START ALICE 3

The goal of this section is to provide information and instructions for downloading, installing, and starting Alice 3.x. Alice (all versions) is free and available for download at www.alice.org.

Minimum system requirements

- Desktop or laptop computer. Alice runs okay on some netbooks. However, many netbook models are not powerful enough to support 3D graphics animation. We suggest a trial run of a sample Alice 3 program on any netbook being considered for purchase.
- Windows XP, Vista, Windows 7, Mac OSX (Leopard, Snow Leopard, Lion, or Mountain Lion), or Linux
- 1 GB RAM (2 GB or more is recommended)
- VGA graphics card capable of high (32 bit) color and at least 1024x768 resolution (3D video card gives faster performance)

- Two- or three-button mouse is recommended. The touchpad on a laptop may be used. Please note, however, that arranging 3D objects in a virtual world is easier to control with a mouse than with a touchpad.

Java JDK:

The Alice installer makes use of the Java JDK (Java SE Development Kit). If the Alice installer indicates the Java JDK has not been installed, then see the instructions at http://help.alice.org for downloading and installing the JDK prior to downloading and installing Alice. If working on a networked system, ask the system administrator to install the JDK.

Downloading and Installing Alice 3

The www.alice.org homepage includes a Downloads menu. Click on Downloads on the menu bar, as shown in Figure 1.1. Select "Get Alice 3.x". The Alice 3.x webpage should be displayed, as shown in Figure 1.2.

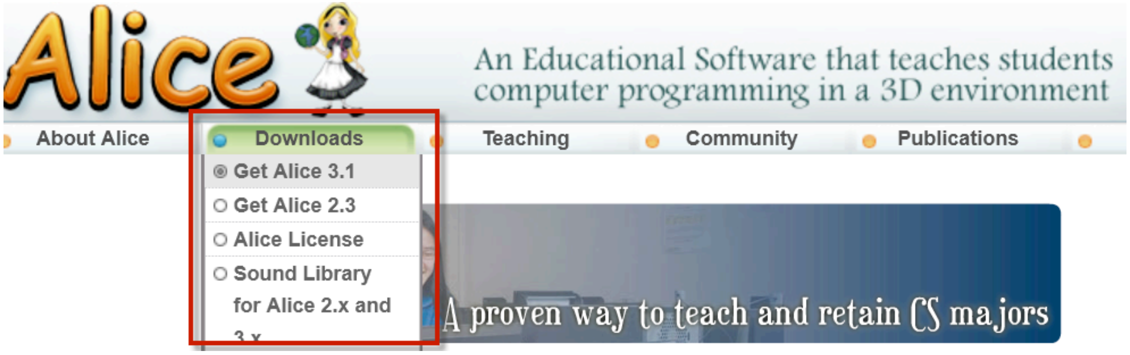
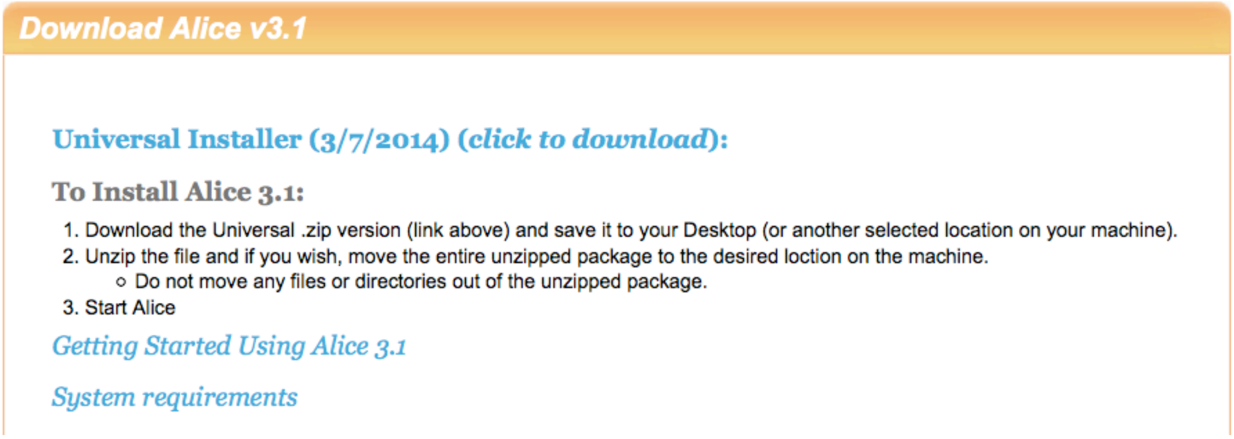

Alice
An Educational Software that teaches students computer programming in a 3D environment
About Alice
Downloads
Teaching
Community
Publications
Get Alice 3.1
Get Alice 2.3
Alice License
Sound Library for Alice 2.x and 3.x
A proven way to teach and retain CS majors

Figure 1.1 Downloads menu on www.alice.org



Download Alice v3.1

Universal Installer (3/7/2014) (*click to download*):

To Install Alice 3.1:

1. Download the Universal .zip version (link above) and save it to your Desktop (or another selected location on your machine).
2. Unzip the file and if you wish, move the entire unzipped package to the desired loction on the machine.
   - Do not move any files or directories out of the unzipped package.
3. Start Alice

Getting Started Using Alice 3.1

System requirements

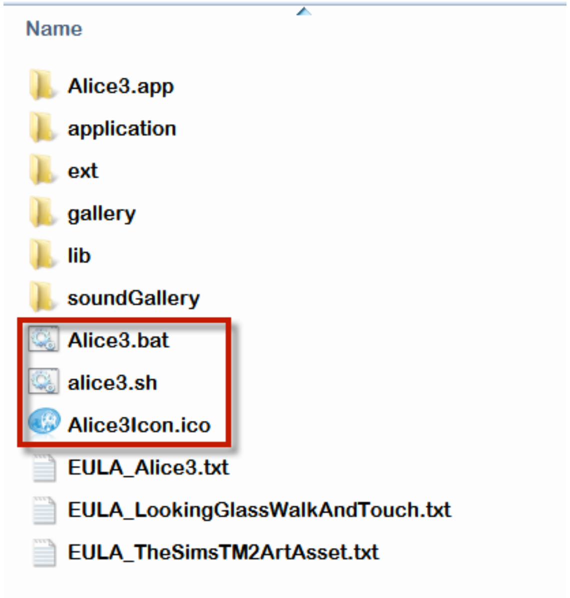
Figure 1.2 The Alice 3.x Download page

Universal installer

The Universal installer works on **multiple platforms and/or on networked machines**. To download the Universal zip, click on the Universal zip Installer link. The Universal zip works on Windows, Mac, and Linux platforms. The Universal zip automatically activates a download of the entire Alice 3.x system. On dial-up connections, this process typically takes 1 ½ - 2 hours, depending on the speed of the connection. After the download has completed, install by unzipping the downloaded file using a compression software application such as WinZip or 7-zip. The Universal zip file should extract to a folder named Alice 3. NOTE: For Windows 7 or 8, unzip to the desktop and then drag to the Program Files folder on the C:\ drive. This will avoid pop-up messages regarding administrator permissions.

Starting Alice 3

Because the downloaded file has just been unzipped, no shortcut icon has been created. Open the unzipped Alice 3 folder to view a list of folders and files, as shown in Figure 1.4. Four start files are highlighted in the red box in Figure 1.4.



Name

Alice3.app
application
ext
gallery
lib
soundGallery
Alice3.bat
alice3.sh
Alice3Icon.ico
EULA_Alice3.txt
EULA_LookingGlassWalkAndTouch.txt
EULA_TheSimsTM2ArtAsset.txt

Figure 1.4 Start files in the Alice 3.x Universal version

These start files are designed to start Alice 3.x on a specific operating system (OS), as is appropriate. To start Alice 3.x, click on the appropriate start file for the OS installed on the computer system, as designated here:

Alice3.bat Starts Alice 3 for a PC system (either 32 or 64 bit)

alice3.sh Starts Alice 3 for a Linux system, 32 bit

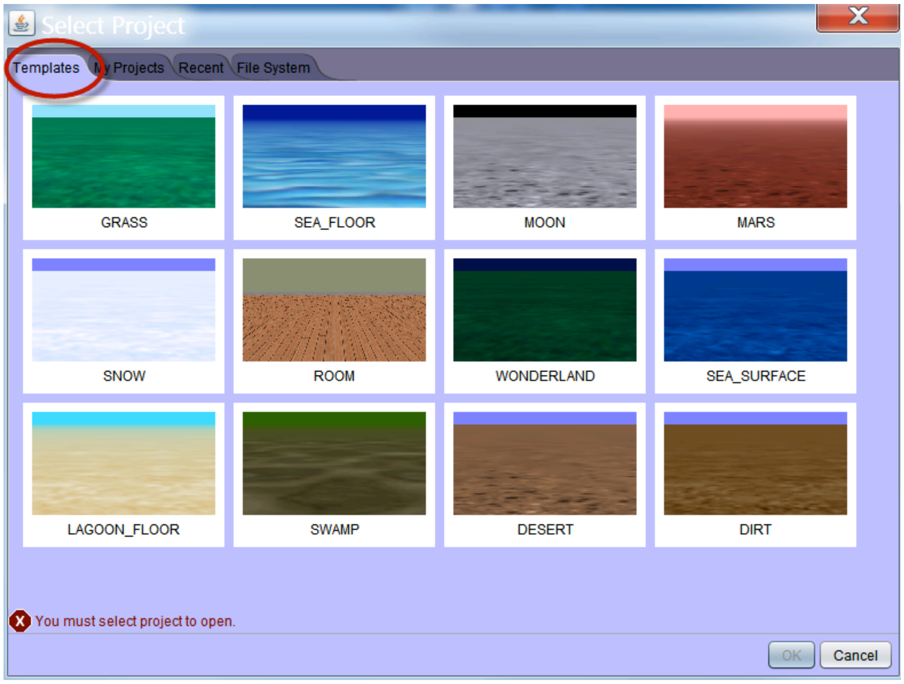
alice364bit.sh Starts Alice 3 for a Linux system, 64 bit

Alice3Icon.ico Drag this icon to a Mac OSX dock. Then click the icon to start Alice 3 on a Mac system (Leopard, Snow Leopard, Lion, or Mountain Lion).

Select Project dialog box

When Alice starts, a Select Project dialog box is automatically displayed, as shown in Figure 1.5. The Select Project dialog box has four tabs and the Templates tab is automatically selected. Choose any one of the templates or click on one of the other tabs to select a previously written Alice project.

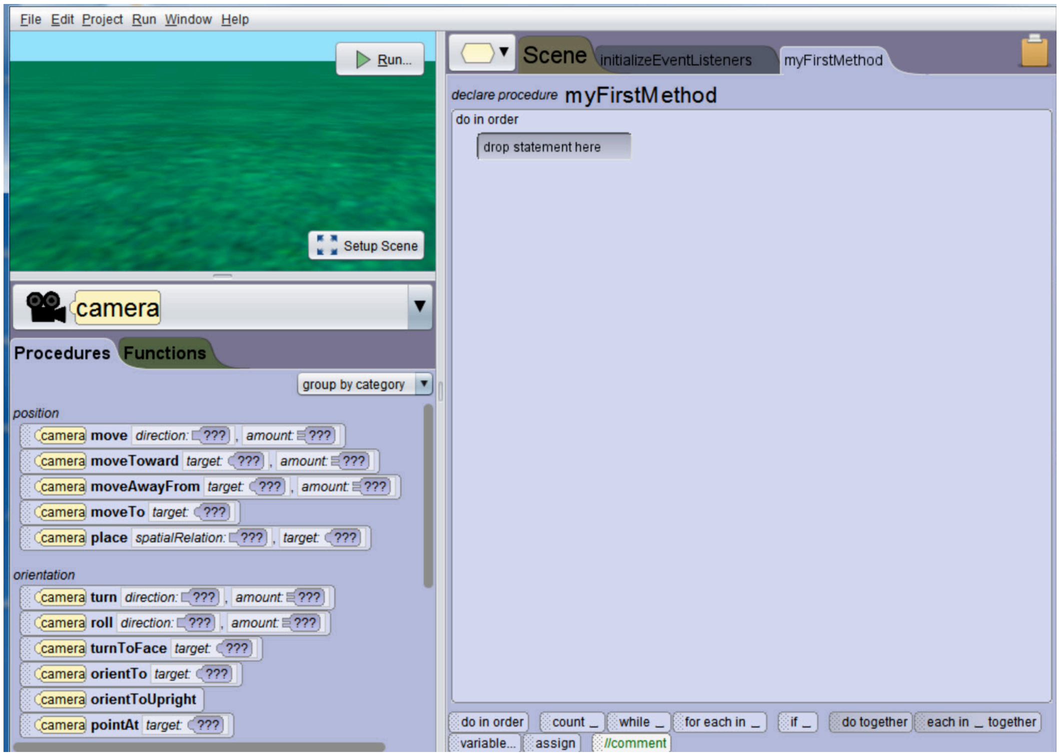
The red X message in the bottom left of the window is a warning message that indicates a template or a previous project must be selected in this window in order for Alice to display the Code Editor and a current scene. If the Cancel button is clicked without selecting a template or a previously written project, Alice will close the Select Project dialog box but the Code editor will not be opened. To reopen the Select Project dialog box, click File in the menu bar at the top left of the Alice window and select New from the menu.



You must select project to open.

Figure 1.5 Select Project dialog box

Upon successful selection of a template, Alice will display the selected template scene in the upper left corner of the Code Editor, as shown in Figure 1.6. (The display may vary somewhat, but the basic organization should be the same as shown here.)



```
declare procedure myFirstMethod
do in order
    drop statement here
```

File Edit Project Run Window Help
Run...
Setup Scene
camera
Procedures
Functions
group by category
position
camera move direction: ??? , amount: ???
camera moveToward target: ??? , amount: ???
camera moveAwayFrom target: ??? , amount: ???
camera moveTo target: ???
camera place spatialRelation: ??? , target: ???
orientation
camera turn direction: ??? , amount: ???
camera roll direction: ??? , amount: ???
camera turnToFace target: ???
camera orientTo target: ???
camera orientToUpright
camera pointAt target: ???
Scene
initializeEventListeners
myFirstMethod
do in order
count _
while _
for each in _
if _
do together
each in _ together
variable...
assign
//comment

Figure 1.6 Start Screen: Selected template in the Code editor

Troubleshooting: PC display driver updates

If Alice does not start or if the templates are not properly displayed, the display driver may need to be updated. For Windows PC users, we advise updating the display driver for the computer system directly from the video display card's manufacturer website (rather than Windows Update). Instructions for updating the display driver for a computer system may be found at http://help.alice.org, in the section labeled "Updating video drivers for Windows machines."

2. A BRIEF TOUR OF THE ALICE 3 IDE

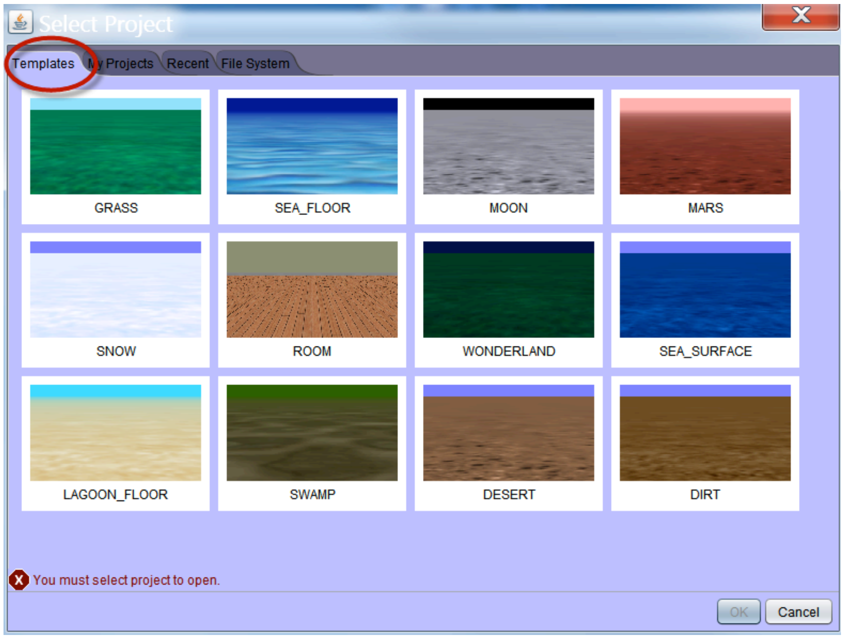
Video: A Brief Tour of the Alice IDE

The goal of this chapter is to provide an overview of the components in the Alice IDE (Interactive Development Environment). The components are briefly described and screen shots identify the individual components. Later chapters will provide greater details and demonstration examples. The IDE components include:

- **Select Project dialog box**: Select a scene template or existing project
- **Code editor**: Camera view, Editor tabs, Control tiles, Methods panel
- **Scene editor**: Camera view, Handles palette, Setup Panel, Gallery

Select Project dialog box

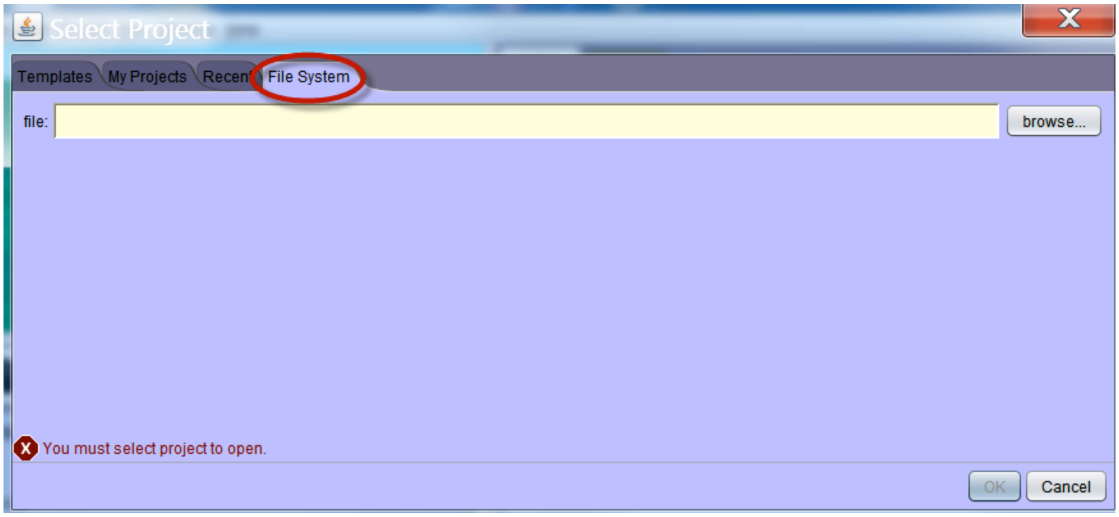
The Select Project dialog box has four tabs, which allow selection of a scene template or an existing project. Figure 2.1 shows the templates each of which contains a surface (for example, grass, moon dust, snow, dirt, or water) and an atmosphere (for example, blue sky, greenish fog, or black outer-space). You can either single-click on the template image and then click the OK button or you can just double-click on the template image.



You must select project to open.

Figure 2.1 Select a template

Other tabs (**My Projects, Recent**, and **File System, as seen in Figure 2.2**) in the Select Project dialog box are for the purpose of opening an existing project. **My Projects** provides a list of existing projects stored in Alice's Projects folder, **Recent** provides a list of recently opened projects, and **File System** provides a directory browser for finding a file in other locations on your computer or a storage device (e.g., thumb drive, CD, or DVD). The **File System** browser is shown in Figure 2.2.



You must select project to open.

Figure 2.2 Select an existing project from the file system

Initial display window and menu bar

Upon selection of a template or starter for the scene, Alice will display the scene in the upper left corner of the window, as shown in Figure 2.3. In Alice, the interface is a programming environment where a virtual world (a scene with actors and props) and a program (a script that gives instructions to the actors) can be created to enable interaction and communication between Alice and a programmer (user).

Code editor

In addition to displaying the Camera View of the scene (upper left), the opening interface displays Edit panel (right) with tabbed panes where different parts of a program are created. The Code editor also has a Methods panel (lower left) and a Controls panel (lower right), as labeled in Figure 2.3. When Alice is first started with a new template, the camera is the selected object, the Camera view displays the scene you would see if you were looking through the camera, and *myFirstMethod* (the main method defined for a scene) is the default open tab in the Edit panel.

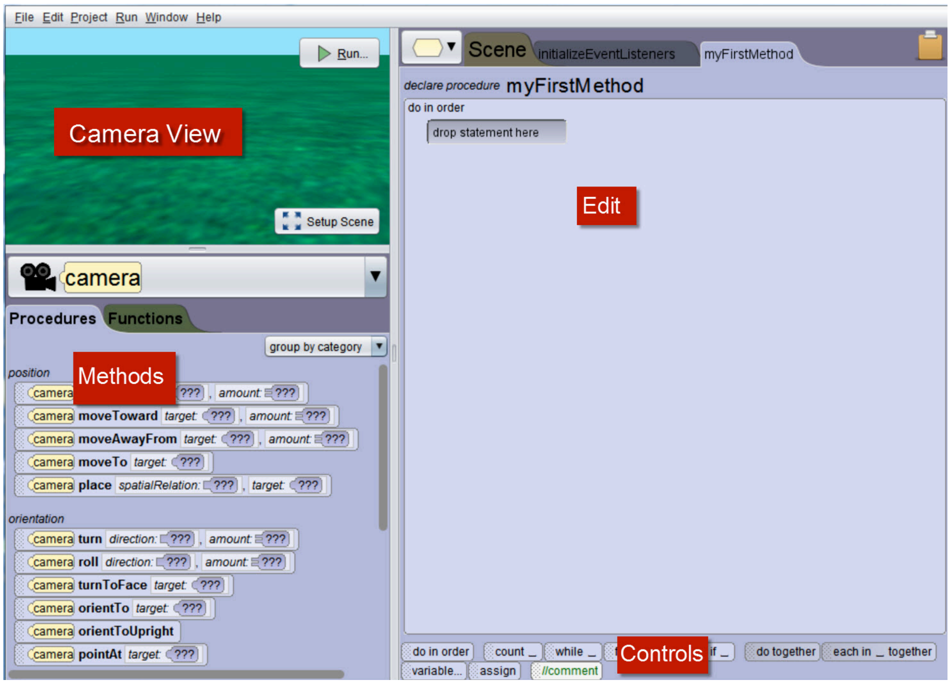
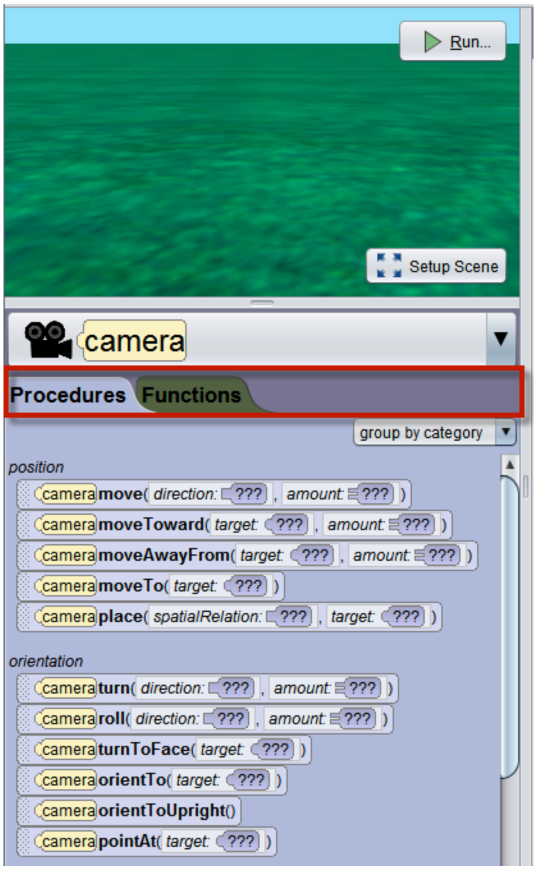

File Edit Project Run Window Help
Run...
Camera View
Setup Scene
camera
Procedures
Functions
group by category
position
Methods
camera ??? , amount ???
camera moveToward target ??? , amount ???
camera moveAwayFrom target ??? , amount ???
camera moveTo target ???
camera place spatialRelation: ??? , target ???
orientation
camera turn direction: ??? , amount ???
camera roll direction: ??? , amount ???
camera turnToFace target ???
camera orientTo target ???
camera orientToUpright
camera pointAt target ???
Scene
initializeEventListeners
myFirstMethod
declare procedure myFirstMethod
do in order
drop statement here
Edit
do in order
count _
while _
Controls
if _
do together
each in _ together
variable...
assign
//comment

Figure 2.3 Code editor panels

Methods: Procedures and Functions

In the Methods panel, each tile represents a method. A method is an action performed on or by an object (animal, person, prop, fish, or some other entity). As shown in Figure 2.4, the Methods panel categorizes methods for display on two tabs: Procedures (methods that perform an action), and Functions (methods that ask a question or compute a value). In the screen capture of this example, the camera object's Procedures tab displays method tiles such as *move, moveToward*, …, *turn, roll*, and others.



```
camera move( direction: ??? , amount: ??? )
camera moveToward( target: ??? , amount: ??? )
camera moveAwayFrom( target: ??? , amount: ??? )
camera moveTo( target: ??? )
camera place( spatialRelation: ??? , target: ??? )

camera turn( direction: ??? , amount: ??? )
camera roll( direction: ??? , amount: ??? )
camera turnToFace( target: ??? )
camera orientTo( target: ??? )
camera orientToUpright()
camera pointAt( target: ??? )
```

Figure 2.4 Methods: Procedures and Functions

Control panel

In the Control panel, each tile represents a statement for managing instructions and data in program code. Figure 2.5 highlights the control tiles.

Most control tiles manage the order in which instructions (method statements) are performed. As an example, the **do in order** tile is used to specify which instructions should be performed in the order in which they are listed. However, the **do together** tile is used to specify which instructions should be performed simultaneously. The *//comment* tile is used to create a statement that is NOT performed.

Some tiles in the Control panel are for managing information (data). As an example, *variable* is used to set aside some memory space for holding data. The memory space is labeled with a name (that is, a variable has a name). The *assign* tile is used to create an instruction that stores data in a variable's memory space.

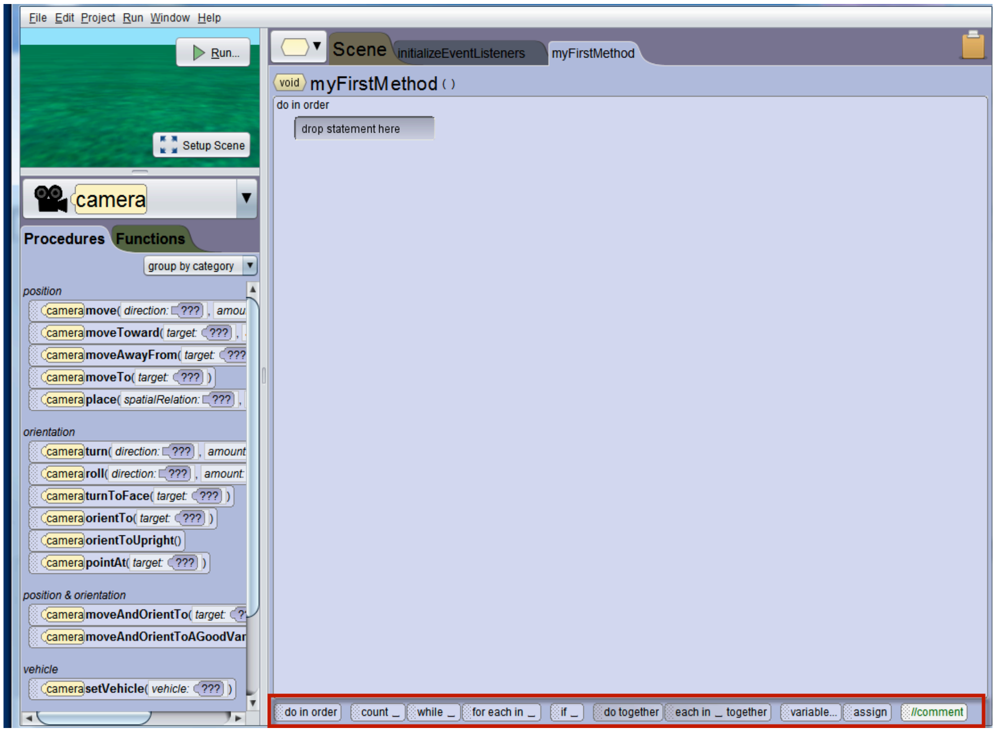

File Edit Project Run Window Help
Run...
Setup Scene
camera
Procedures Functions
group by category
position
camera move(direction: ??? , amou
camera moveToward(target: ??? ,
camera moveAwayFrom(target: ???
camera moveTo(target: ???)
camera place(spatialRelation: ??? ,
orientation
camera turn(direction: ??? , amount
camera roll(direction: ??? , amount:
camera turnToFace(target: ???)
camera orientTo(target: ???)
camera orientToUpright()
camera pointAt(target: ???)
position & orientation
camera moveAndOrientTo(target: ?
camera moveAndOrientToAGoodVar
vehicle
camera setVehicle(vehicle: ???)
Scene
initializeEventListeners
myFirstMethod
void myFirstMethod ( )
do in order
drop statement here
do in order
count _
while _
for each in _
if _
do together
each in _ together
variable...
assign
//comment

Figure 2.5 Control tiles

In summary, the Code editor provides a drag-and-drop environment where method and control tiles are dragged into the edit space to create **instructions** (method and control statements) that compose a program. In Alice, a program animates objects in a scene.

Scene editor

To view the Scene editor, click the Setup Scene button in the lower right corner of the scene, as shown in Figure 2.6.

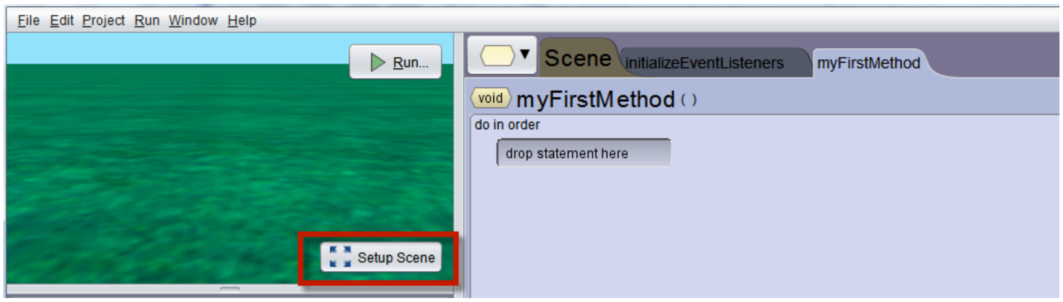
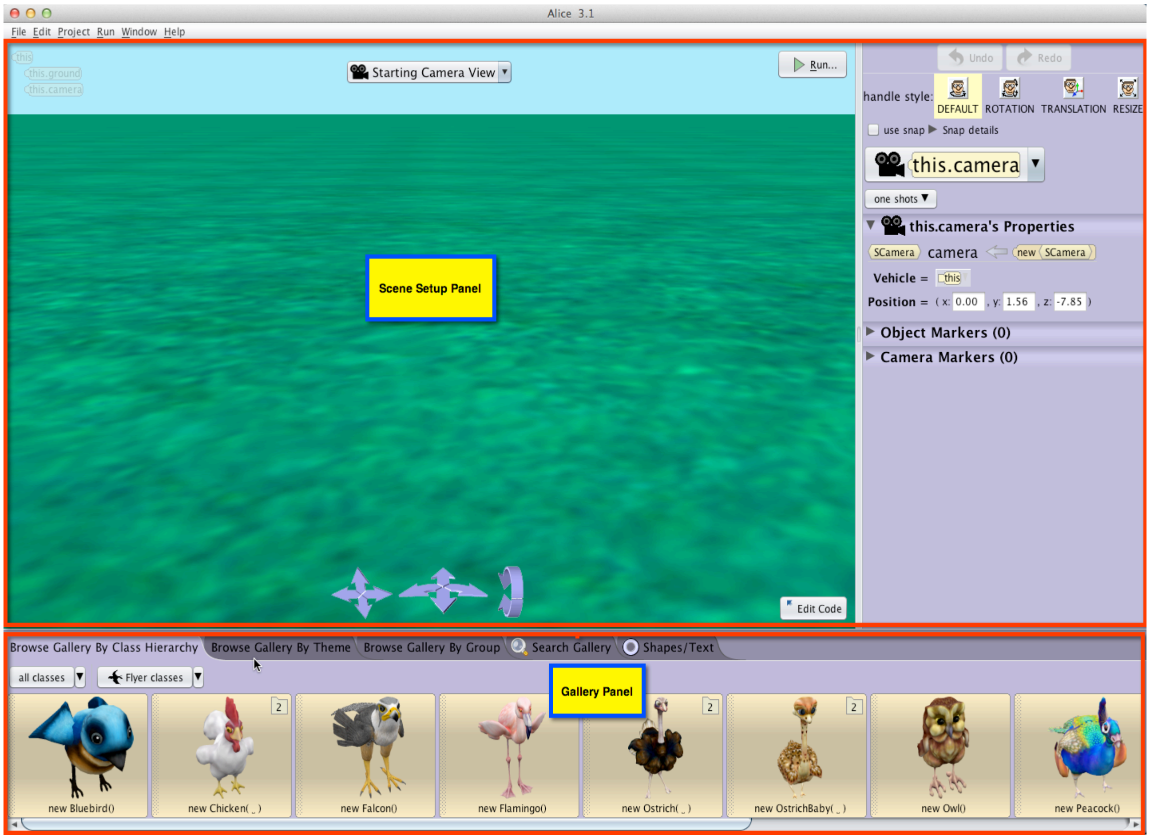


Figure 2.6 Setup Scene button toggles to the Scene editor

As labeled in Figure 2.7, the Scene editor has two panels: Scene Setup and Gallery. The purpose of this editor is to create a virtual world by adding and arranging the objects in a scene. The Gallery contains 3D models that are used to create objects in the scene. The SetUp panel provides mouse control handles for positioning objects in the scene and menus for changing size, color, vehicle, position, and other properties of objects in the scene.



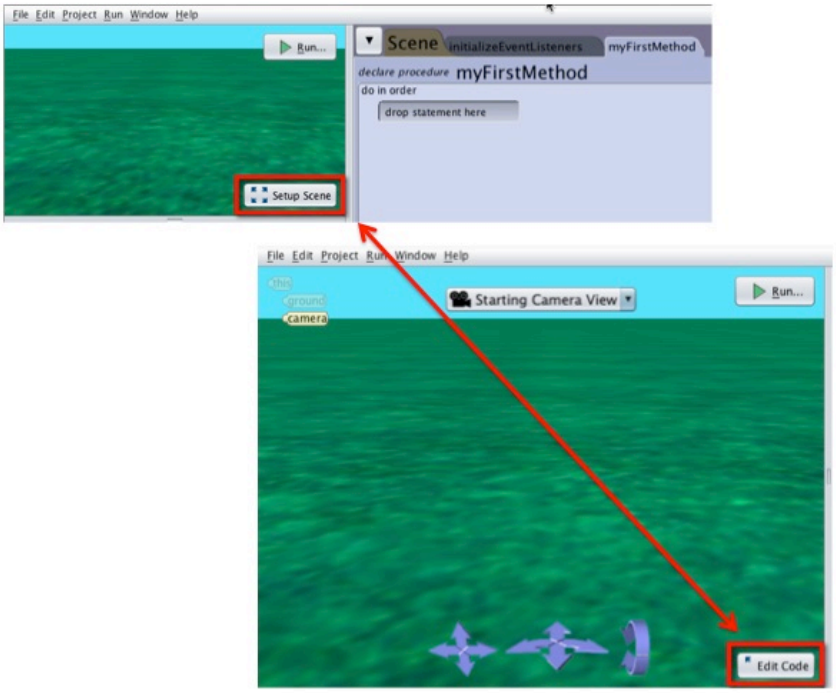
Vehicle = this

Position = (x: 0.00 , y: 1.56 , z: -7.85)

Figure 2.7 Alice 3: Scene editor panels

Toggle between two editors

The creation of an animation often involves frequent switching back and forth between the Code and Scene editors. To toggle between the two editors, click the Setup Scene button in the Code Editor or click the Edit Code button in the Scene editor, as shown in Figure 2.8.



```
declare procedure myFirstMethod
do in order
  drop statement here
```

Figure 2.8 Toggling between Code and Scene editors

3. A BRIEF TOUR OF THE MENU BAR

The purpose of this section is to introduce the menus in the menu bar. Many menu items are typical of commonly used software applications and their operations are well-known. We will assume the reader is familiar with these items and no illustration will be provided. Some items are specific to the operation of Alice, in which case we provide a brief description and an illustration of the items.

In Alice 3, a menu bar is displayed in the upper left corner of the window, as shown in Figure 3.1. The menus include: File, Edit, Project, Run, Window, and Help.

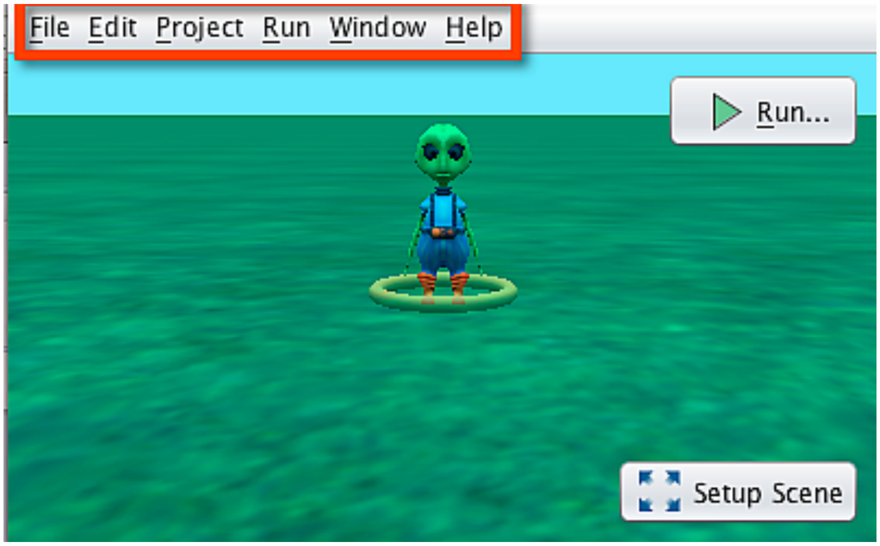


Figure 3.1 Menu bar

File menu

The File menu contains options for managing and editing files in a project, as shown in Figure 3.2. The items in the File menu are: **New, Open, Recent Projects, Save, Save As, Revert, Upload to YouTube, Print,** and **Screen Capture.** On Windows machines, you may also see **Exit** (at the bottom of the menu).

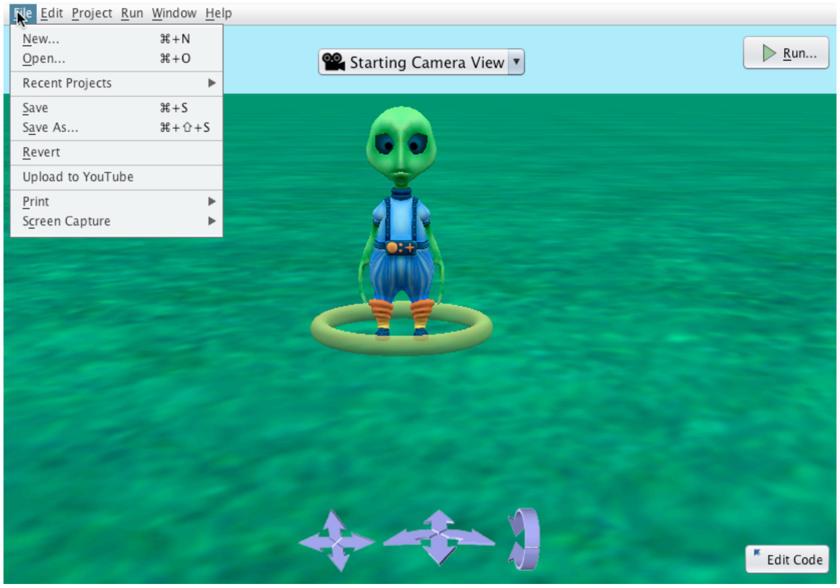


Figure 3.2 Menu for managing files

New, Open, Save, Save As, and **Exit** are typical of many software applications and their usage is well-known. Our only suggestion is that the first time an Alice project is saved, use Save As instead of Save. Save As guarantees that the file will be saved in a user-selected directory rather than a default directory. Files can be saved on the C drive (hard drive), a networked drive, a USB drive, or a read-write CD/DVD.

- **Recent Projects** provides links to recently saved Alice 3 projects.
- **Revert** restores a scene to its initial state when the world was first opened in the current session. In other words, all actions in the current editing session are backtracked and removed.
- **Upload to YouTube** (at the time of this writing, this feature is not yet fully implemented) will allow you to capture a video of the currently running animation and export (upload) to a YouTube account. Alternately, the video can be saved to your computer for later playback.
- The **Screen Capture** menu item is available for capturing images in either Code or Scene editor mode. As shown in Figure 3.3, the **Screen Capture** menu item cascades to three choices for selecting an area of the screen to be captured. The ***Capture Entire Window*** option will copy a screen shot of the entire Alice 3 interface to your system clipboard, which can then be pasted into another document. ***Capture Rectangle*** cross-hatches the entire Alice IDE. You then click and drag over the portion of the Alice interface that you wish to copy to the clipboard. ***Screen Capture…*** brings up a dialog box that allows you to choose to capture the Entire Window, the Content Pane, or a Rectangle region. It will also allow you to set the resolution dpi for the image, as illustrated in Figure 3.4.

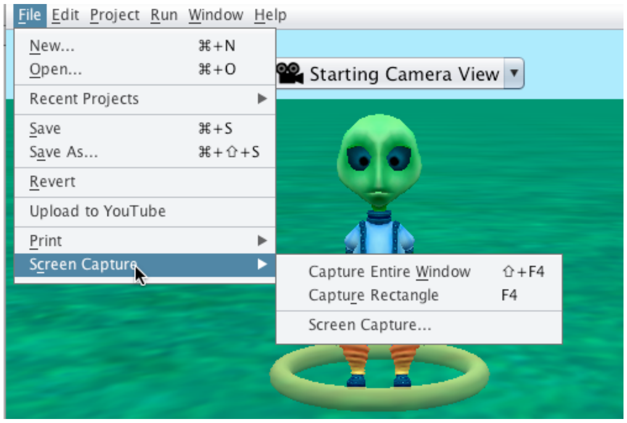
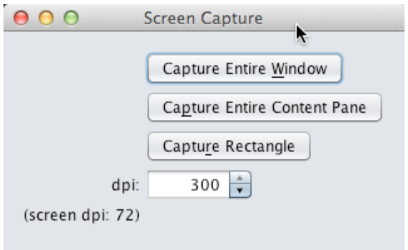


Figure 3.3 Screen Capture options in the File menu



dpi: 300

(screen dpi: 72)

Figure 3.4 Screen Capture… submenu dialog box

NOTE: The built-in Screen Capture in Alice 3 does NOT capture images from the runtime window while an animation is playing. However, this can be accomplished on a Windows PC system by using the Alt/PrintScreen keyboard shortcut. The captured image is automatically saved to the system clipboard. Just paste it into Paint or a Word doc, using CTL/V. On a Mac OS X system, the keyboard shortcut for capturing the runtime window is Command-Shift-4. A cross-hair cursor will appear and you can click and drag to select the area you wish to capture. When you release the mouse button, the screen shot will be automatically saved as a PNG file on your desktop.

- The **Print** menu item is available for printing program code. As shown in Figure 3.5, the **Print** menu item cascades to three choices for selecting how much of the program code to print. The ***Print All*** option will print all the code created in the program. (Of course, this does not include pre-written code which is part of the Alice system.) ***Print Current Code*** will print only the code in the currently active method tab in the code editor. P***rint Scene Editor*** will print a screen capture of the Scene Editor, including a screen capture of the Camera View and the Setup panel but not including the Gallery panel.

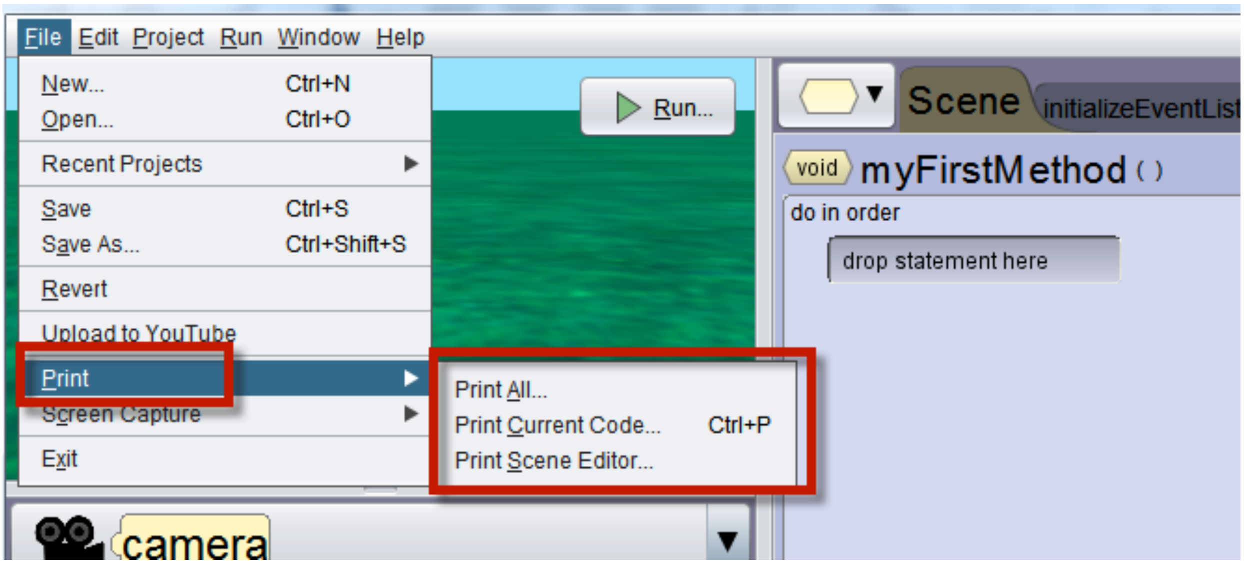


Figure 3.5 Print options in the File menu

Edit menu

The **Edit** menu contains ***Undo***, ***Redo***, ***Cut***, ***Copy***, and ***Paste***, as shown in Figure 3.6. These are all standard editing actions. As of this writing, ***Cut***, ***Copy***, and ***Paste*** are not implemented but are listed in the menu to allow for future modifications. Truthfully, although the traditional cut, copy, and paste actions work well in a text editor, these actions are of limited usefulness in a drag-and-drop editor. Section 16 of this guide provides detailed instructions on using the clipboard for ***Cut***, ***Copy***, and ***Paste*** in a drag-and-drop programming environment.

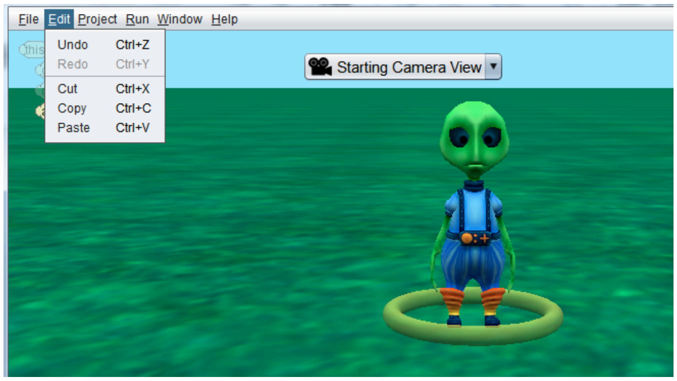


Figure 3.6 Menu for Edit options

Project menu

The project menu contains ***Resource Manager***, ***Find***, and ***Statistics***. The ***Resource Manager*** item opens a dialog box for importing (or removing) resource files as shown in Figure

3.7. A resource file may be either an audio or image file. Alice does not provide sound or image editing capabilities.

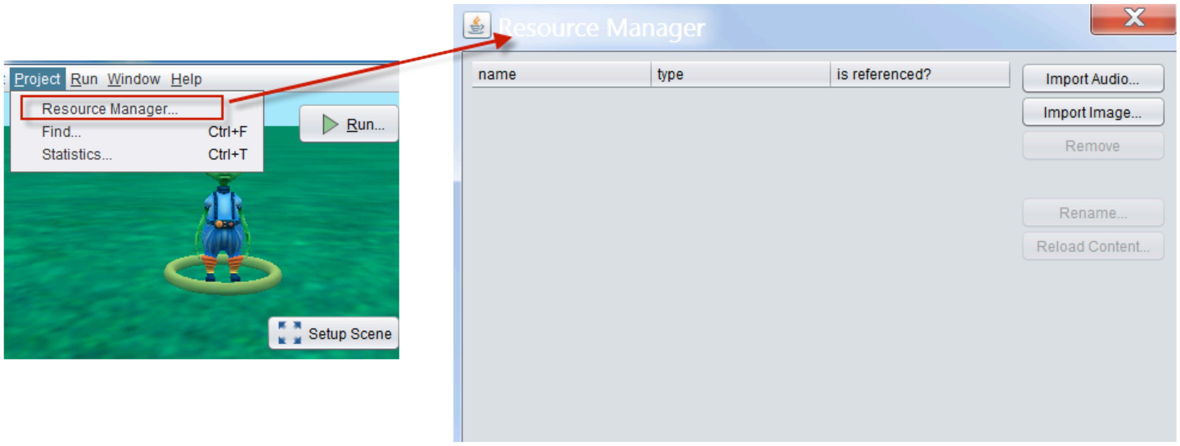
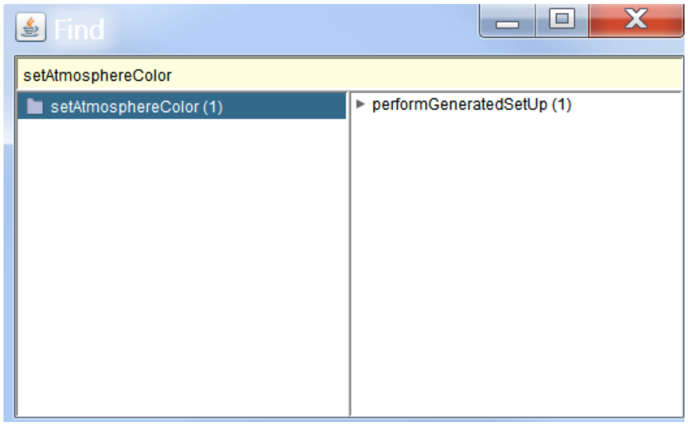


Figure 3.7 Project menu and Resources Manager

The ***Find*** item pops up a dialog box for searching the program code to find where a method is called. In Figure 3.8, the scene's method named ***setAtmosphereColor*** has been selected in the Find box. A message appears in the column on the right to show that the method has been called (one time) in the program's *performGeneratedSetUp* method.



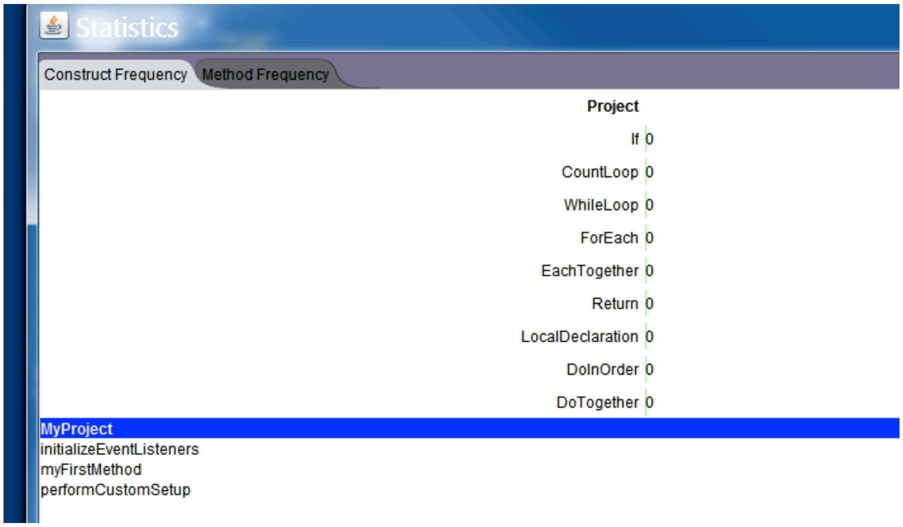
setAtmosphereColor

setAtmosphereColor (1)

performGeneratedSetUp (1)

Figure 3.8 Find where a method is called in the program

The ***Statistics*** item pops up a window that displays a frequency analysis of constructs and method calls within the current project, as shown in Figure 3.9.



Statistics

Construct Frequency | Method Frequency

Project	
If	0
CountLoop	0
WhileLoop	0
ForEach	0
EachTogether	0
Return	0
LocalDeclaration	0
DoInOrder	0
DoTogether	0

MyProject
initializeEventListeners
myFirstMethod
performCustomSetup

Figure 3.9 Statistics popup window

Window menu

The Window menu contains ***Perspectives***, ***Project History, Memory Usage***, and ***Preferences***, as shown in Figure 3.10. These items control the display of the Alice 3 environment in terms of the number of open windows and their content.

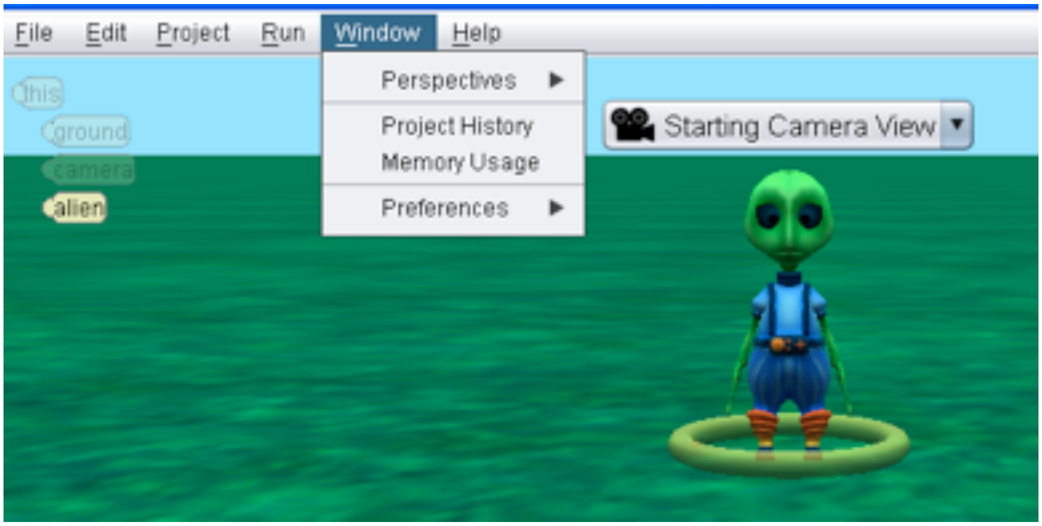


Figure 3.10 Window menu

The ***Perspectives*** menu provides an alternate means of toggling between the Code Editor or Scene Editor display, as shown in Figure 3.11.

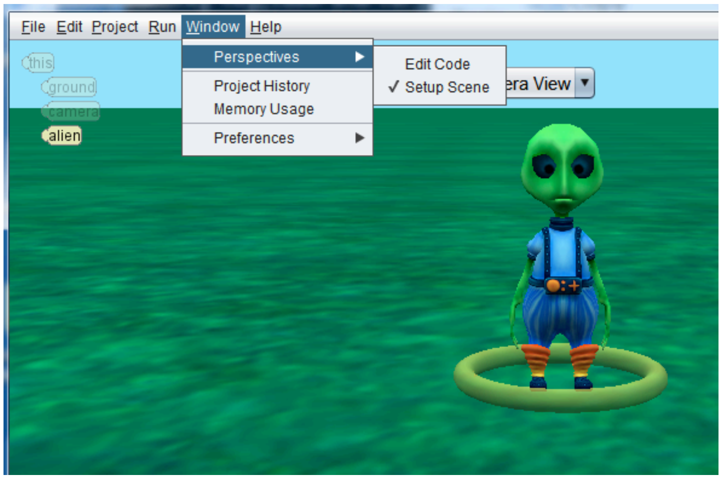
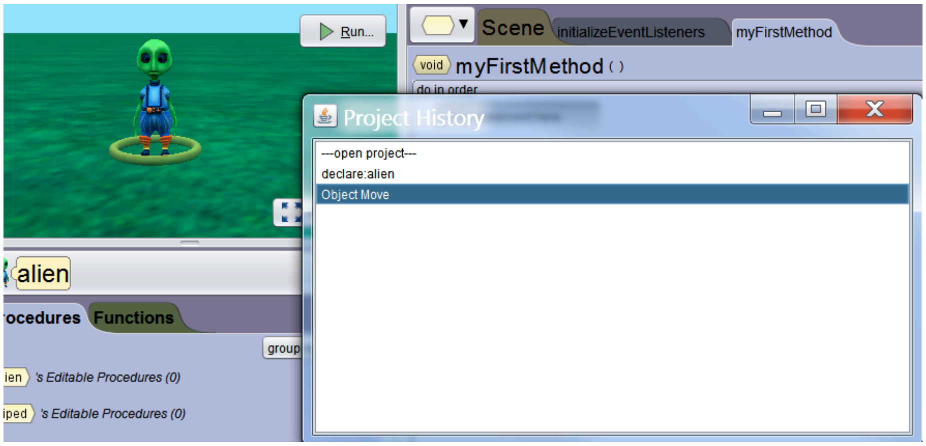


Figure 3.11 Perspectives

The ***Project History*** item opens a new window containing a list of all actions performed (thus far) in the current editing session. Figure 3.12 shows a ***Project History*** window in which the actions listed include the declaration of an alien object (was added to the scene) and then the object was moved. The actions in the history are listed in the order they were performed. The history does not extend over the life of the project, only having a record of actions in the current editing session.



---open project---
declare:alien
Object Move

Figure 3.12 View the history of actions in this session

Backtrack In History

It is possible to backtrack to a previous state (objects in the scene, their locations, and their properties) of the world by clicking an earlier item in the Project History. Selecting an item (other than the last one) causes all later actions in the list to be "played backwards." To illustrate, we clicked **declare alien** in the Project History and the Object Move action was played backwards, moving the alien back to its initial location when it was added to the world. The state of the world is now displayed, as shown in Figure 3.13.

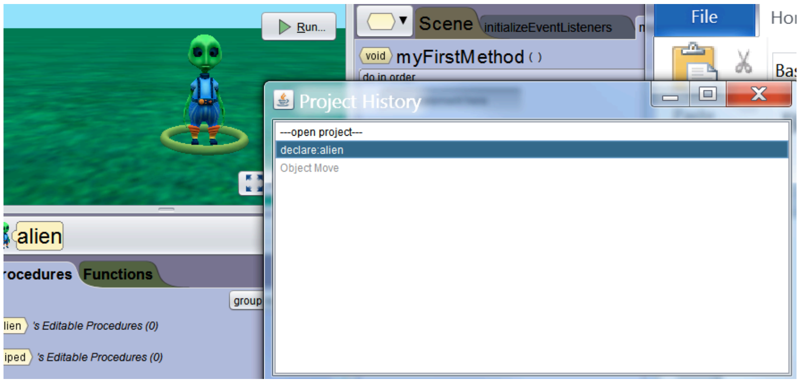


Figure 3.13 Backtracking to a previous state

Selecting the Memory Usage item in the Window menu opens a popup window in which memory usage is tracked, as shown in Figure 3.14. An alert is displayed in the window's task bar when Java's garbage collection is in progress.

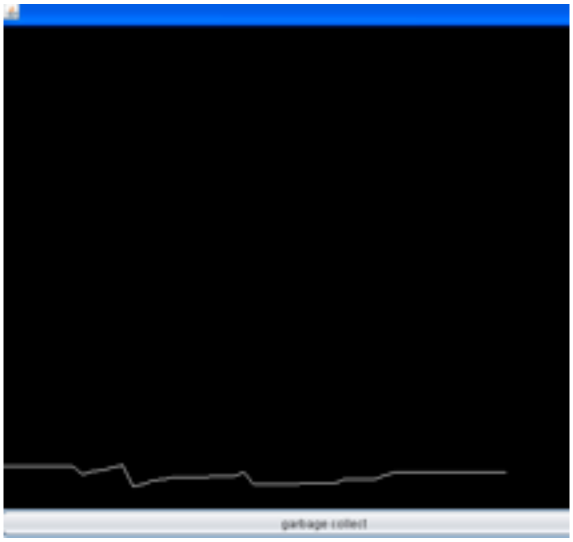


Figure 3.14 Memory usage window

Selecting the Preferences menu item opens a cascading menu for setting preferences in the Alice 3 environment, as shown in Figure 3.15.

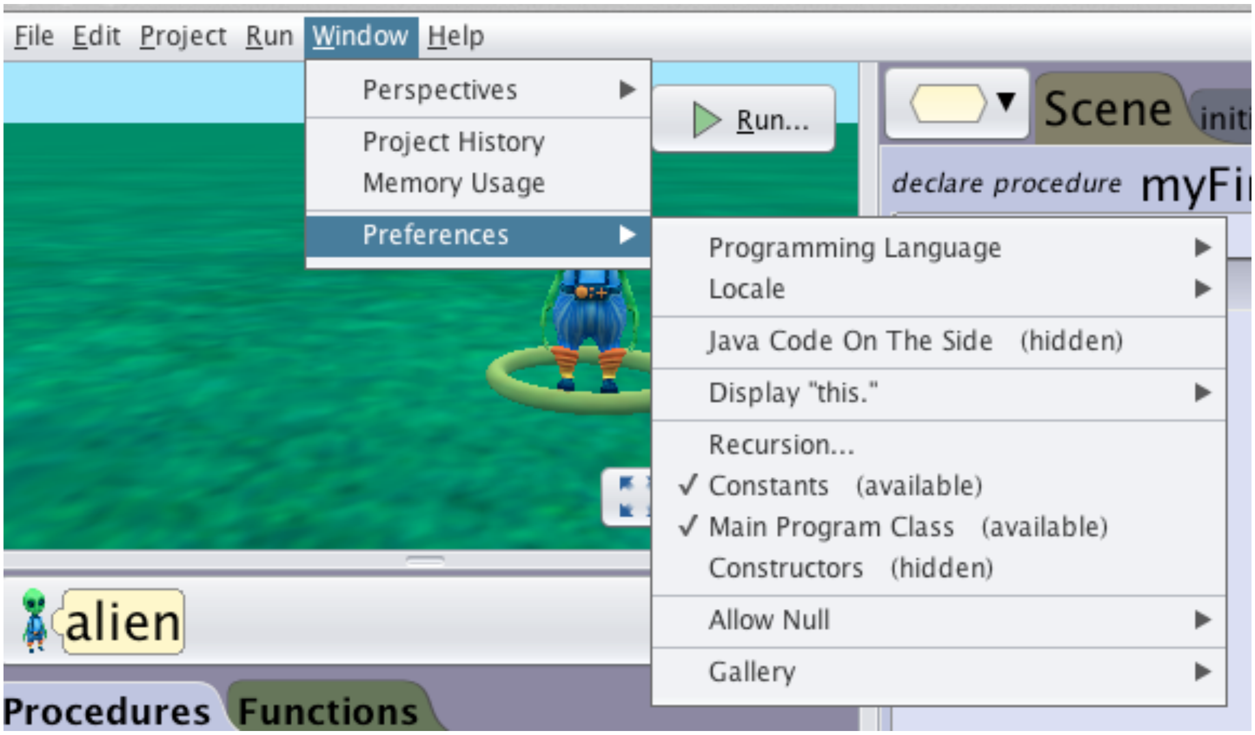
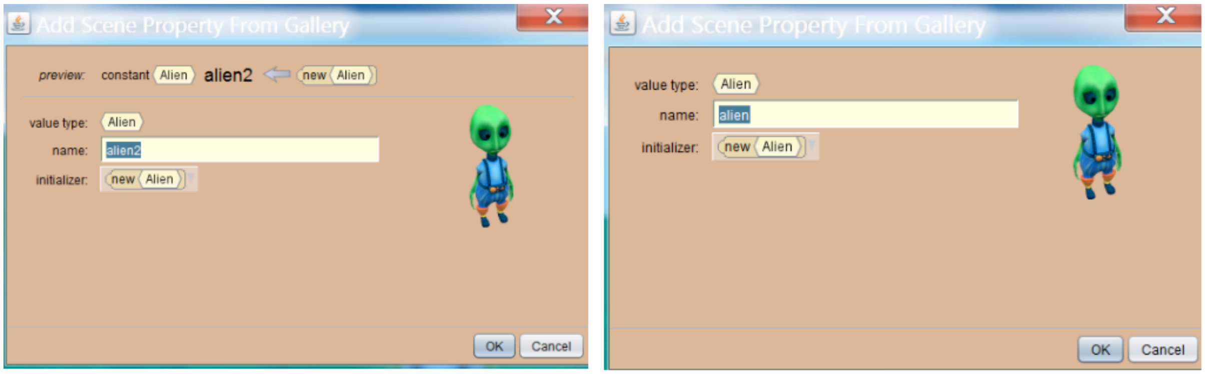


Figure 3.15 Preferences cascading menu

A quick overview of the Preferences menu items is provided here. Details for setting preferences are provided in the next section of this How-To guide.

- **Programming Language:** Display code using Alice or Java syntax
- **Locale**: Display code in the natural language selected (English, Spanish, Chinese, Portuguese, Russian, and others).
- **Java Code On The Side**: Display Alice code side-by-side with its Java syntax form. The Alice code can be edited and the Java code will automatically update. However, the Java code cannot be edited directly in this view.
- **Display "this.":** Display "this." in program code to represent any object of the currently selected class. An option is available to disable "this." when writing code in the Scene class.
- **Recursion**: Enable the use of recursion in Alice programs.
- **Constants**: Allows the user to declare constant fields in the program.
- **Main Program Class**: Includes the Program class in the Class menu list. Program contains the main method, which is the first method executed when the Run button is clicked.

- **Constructors**: Add a constructor option to the Class tab for each class used in a scene.
- **Allow Null**
  - Allow Null for field initializers: Allow a class variable to be declared without an initial value
  - Allow Null for local initializers: Allow a local variable to be declared without an initial value.
- **Gallery**: Preference settings for the dialog box are displayed when adding an object to a scene. Gallery preference settings include enabling or disabling a preview of the declaration for creating an object and an option for auto-naming. Figure 3.16 shows the dialog box with (left) and without (right) a preview of the declaration statement.



Add Scene Property From Gallery

preview: constant Alien alien2 ⇐ new Alien

value type: Alien

name: alien2

initializer: new Alien

Add Scene Property From Gallery

value type: Alien

name: alien

initializer: new Alien

Figure 3.16 Preview on (left) and off (right) in the preference settings

Help menu

The Help menu contains: **Help…, Help with Graphics Problems…, Report a Bug…, Suggest improvement…, Request a New Feature…, Show Warning…, Show System Properties…,** and **Browse Release Notes[web]**, as shown in Figure 3.17.

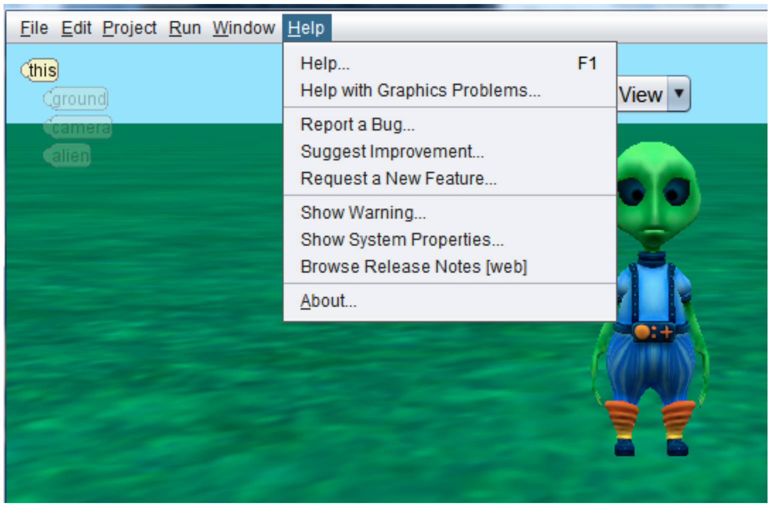
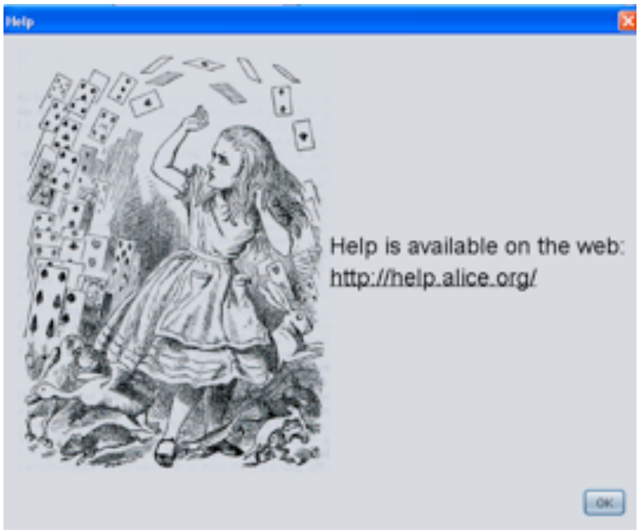


Figure 3.17 Help menu

The **Help…** item opens a window containing a link to the help page at *http://help.alice.org*, as shown in Figure 3.18.

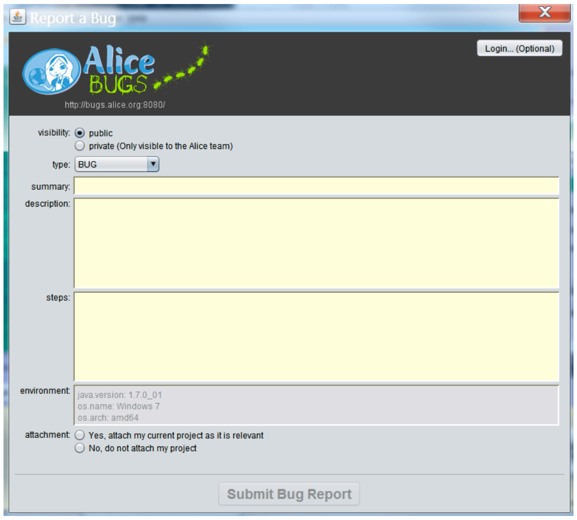


Help is available on the web:
http://help.alice.org/

Figure 3.18 Help link

The **Report a Bug…, Suggest Improvement…**, and **Request a new Feature**… items each open a window containing a form for the specified action. This feature allows Alice users to submit a bug report, suggest improvements, and provide ideas for new features. A copy of the bug report form is shown in Figure 3.19.

For any of these forms, submitting the form requires the computer be actively connected to the internet. If not connected to the internet, the report will simply be deleted when Alice is closed on your computer.



visibility: ● public
○ private (Only visible to the Alice team)

type: BUG

summary:

description:

steps:

environment: java.version: 1.7.0_01
os.name: Windows 7
os.arch: amd64

attachment: ○ Yes, attach my current project as it is relevant
○ No, do not attach my project

Figure 3.19 Bug report form

4. How to Set Preferences

Setting preferences changes the "look and feel" of the Alice 3 IDE. The purpose of this section is to demonstrate how to set a preference. Any combination of preference settings is possible, as selected by the user.

Default preferences

The Alice installer has a pre-defined set of preferences for the "look and feel" of the Alice environment. The default settings are shown in Figure 4.1, where only one item (Constants) is selected in the menu. Enabling the Constants preference turns on the ability to create a named value that cannot be modified at runtime.

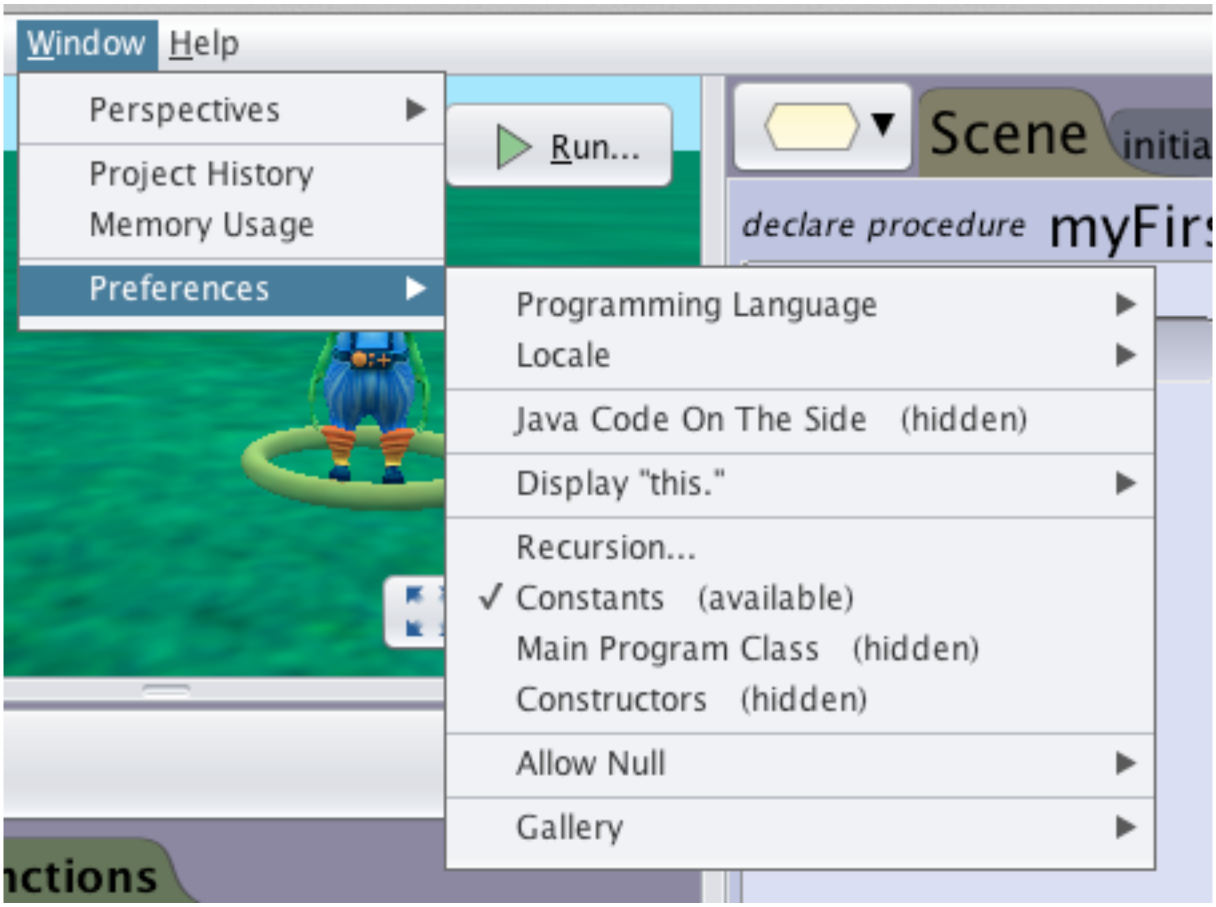
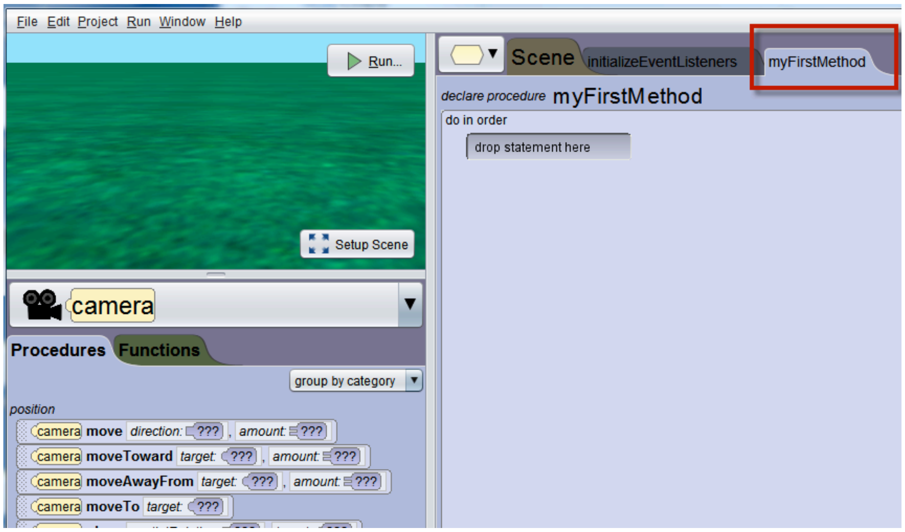


Figure 4.1 Default preference settings

With the default preference settings, Alice starts with Scene as the currently active class, as shown in Figure 4.2. The active editor tab is *myFirstMethod*, a method belonging to the Scene class. After code has been created and the user clicks on the Run button, the scene will be displayed in a popup window (runtime window) and then the code in *myFirstMethod* will be executed (run).



```
declare procedure myFirstMethod
do in order
    drop statement here
```

```
position
camera move direction: ??? , amount: ???
camera moveToward target: ??? , amount: ???
camera moveAwayFrom target: ??? , amount: ???
camera moveTo target: ???
```

Figure 4.2 myFirstMethod tab

Another default preference setting is for the keyword, *'this'*. In both Alice and Java, the keyword *'this'* refers to the current object of this class. As an example, in the Scene class *'this'* is the current scene, in the Alien class *'this'* is the current alien, and in the Penguin class *'this'* is the current penguin.

In an Alice world, a scene often contains objects of other classes, as shown in Figure 4.3. This scene contains ground, camera, alien, and penguin objects. In the pull-down menu, 'this' is the scene and the objects belonging to the scene are labeled *this.ground, this.camera, this.alien,* and *this.penguin.*

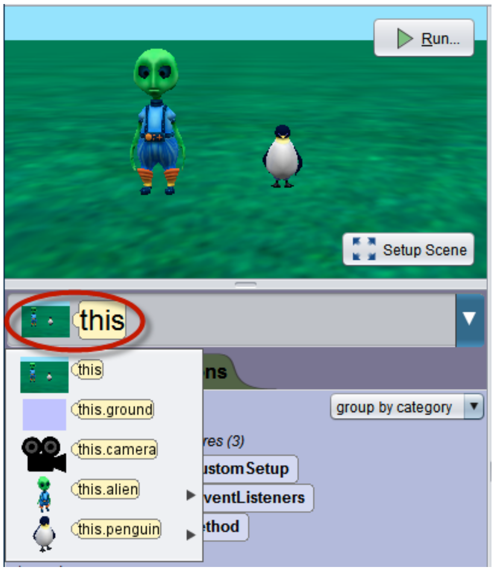


Figure 4.3 'this' is the scene

Displaying the keyword *'this'* is enabled by default, as shown in Figure 4.4. If you find this practice to be confusing or distracting, you may elect to turn it off in the preferences menu, as shown in Figure 4.5. (**Note:** You may need to save the world, close, and reopen to refresh the menu display.)

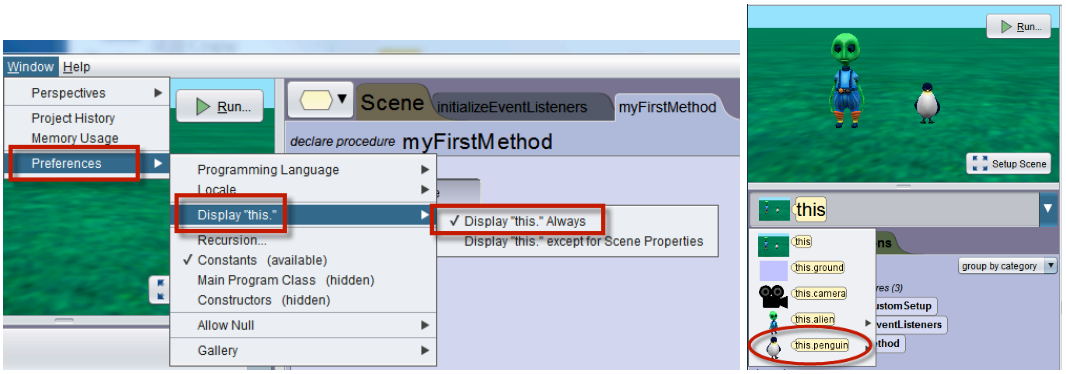


Figure 4.4 'this' is displayed for a scene and each object within it

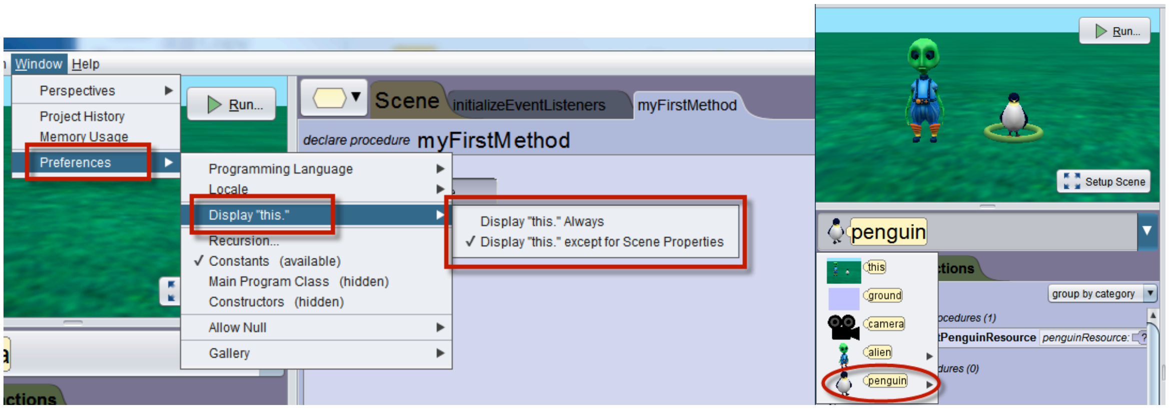


Figure 4.5 'this' is displayed for the scene object but not for objects within

Setting multiple preferences

Any combination of preferences may be set. Figure 4.6 shows two recommended preferences (Constants and Constructors) for those who wish to focus on object-oriented programming concepts with an intention to prepare for learning a production level language, such as Java.

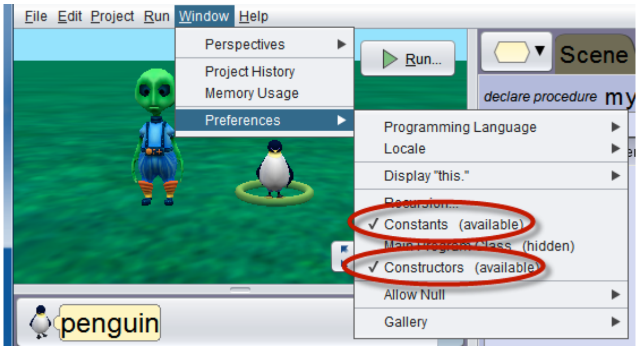
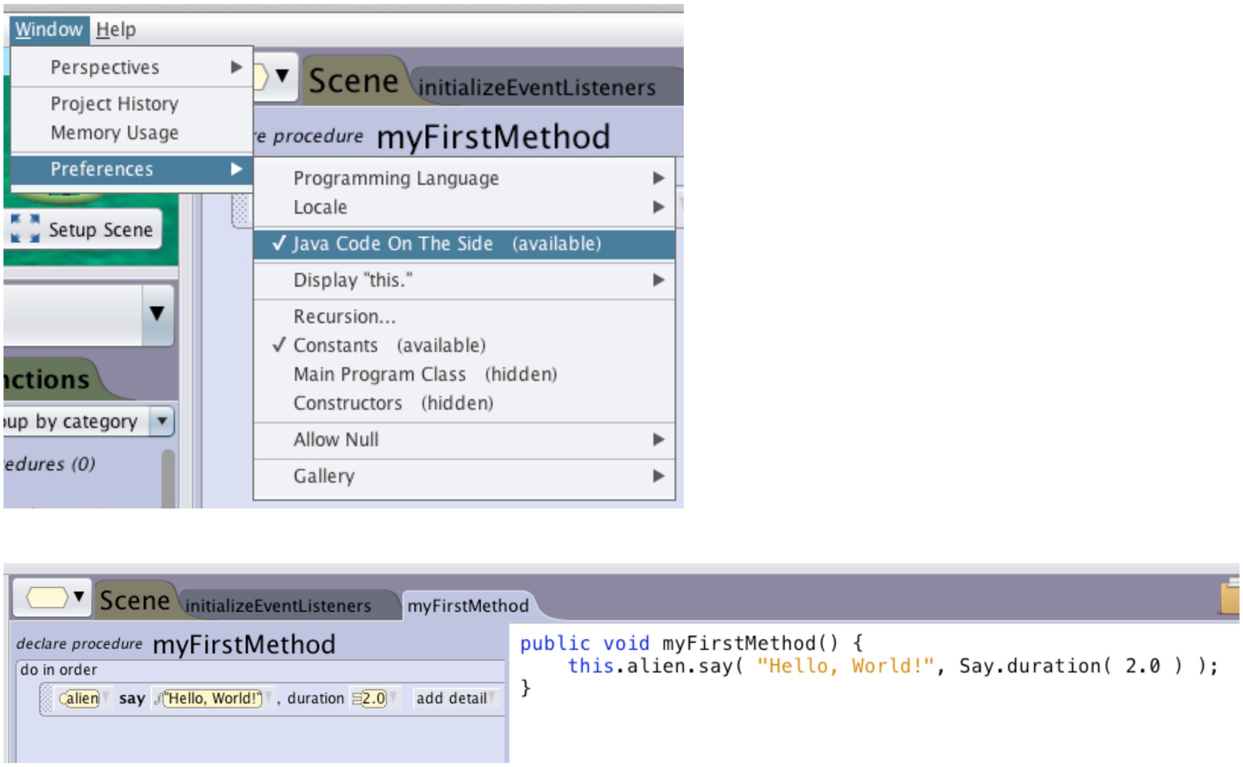


Figure 4.6 A selection of two preferences

Setting a preference for Alice and Java Side-by-Side

For those using Alice as preparation for learning Java, setting the Java Code on the side preference enables a dual display of Alice and Java code in side-by-side panels, as shown in Figure 4.7.



✓ Java Code On The Side (available)
✓ Constants (available)
Main Program Class (hidden)
Constructors (hidden)

```
public void myFirstMethod() {
    this.alien.say( "Hello, World!", Say.duration( 2.0 ) );
}
```

Figure 4.7 Alice and Java code, side-by-side display preference

5. Classes and the Gallery of 3D Models

Video: Tour of the Alice 3 Gallery

The purpose of this section is to explore the concept of classes and objects as well as illustrate the relationship of the 3D models (as provided in the Gallery) to classes and objects in an Alice 3 project.

Models

In our daily lives, we think of a model in many different ways. We think of a model as a product when we say, "This car is the latest model." We might think of a model as someone to be imitated when we say, "She is a model student." To an architect, a model is a blueprint (a design for construction). Figure 5.1 illustrates a blueprint for house. This blueprint is a model that provides a design. The blueprint model tells a home-builder how to build the house but is not an actual physical instance of a house.

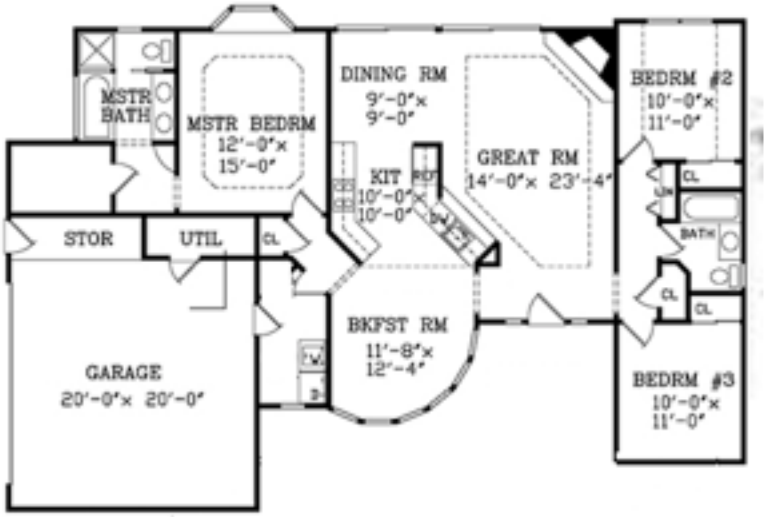


Figure 5.1 A blueprint for constructing a house

3D Models and Classes

In animation film studios such as Disney, Pixar, and DreamWorks, a 3D model is a digital representation of an entity (someone or something) in three dimensions (height, width, and depth). Animation adds motion to a model. A 3D model contains instructions for building the digital object.

In Alice, a class puts together a digital representation of an entity, a plan for constructing it, and instructions for animating, all in the same jar fire. A more general definition is: A class defines a type of object (a modeled entity) and actions that can be performed by that object.

Gallery

The Gallery (in the Scene Editor) contains classes for creating and animating objects in an Alice virtual world. Figure 5.2 shows a collection of classes in the Gallery's Biped collection. Each class is a 3D model for building an object of a specific type (for example, an alien, a cat, or a curupira).

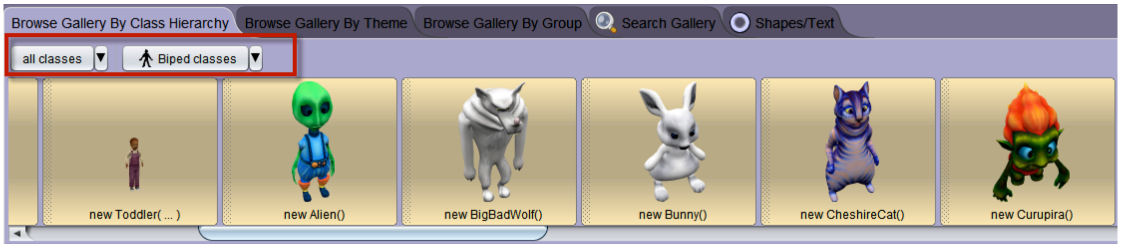
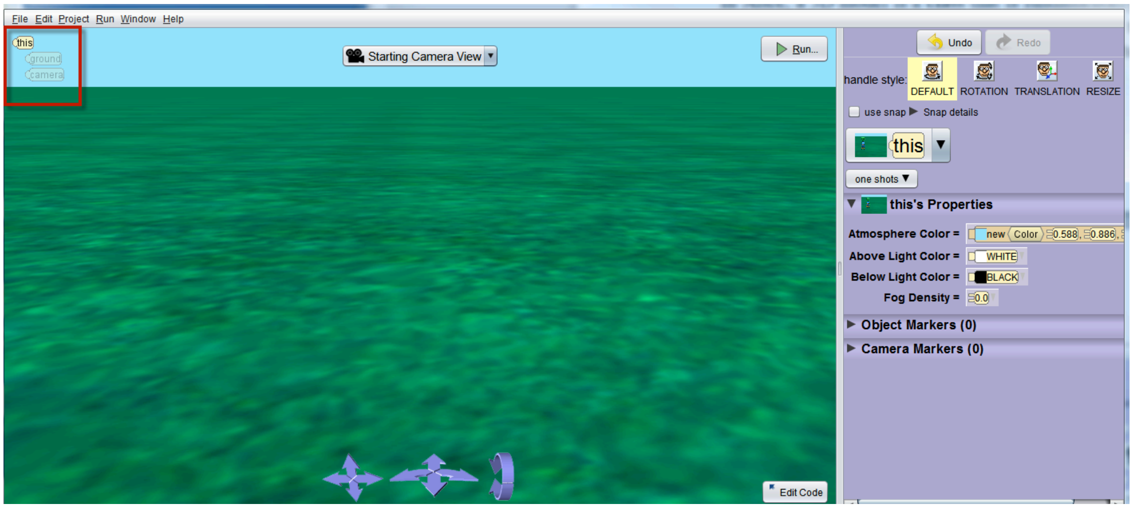


Figure 5.2 Classes (3D Models) in the Alice 3 Biped

Using the Gallery

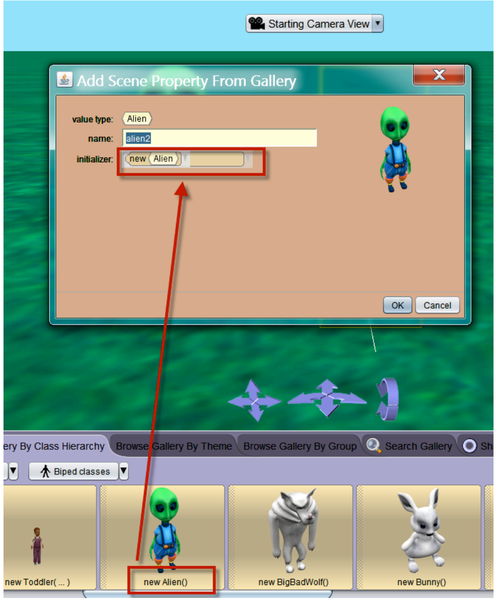
Figure 5.3 shows a newly created Alice world. An Object tree is displayed in the upper left corner of the scene. The Object tree contains a list of all the objects in this scene. A new scene automatically has a ground (or water) surface and a camera. The scene is an object of the Scene class, the ground an object of the Ground class, and the camera an object of the Camera class.



Atmosphere Color = new Color 0.588, 0.886,
Above Light Color = WHITE
Below Light Color = BLACK
Fog Density = 0.0

Figure 5.3 A new Alice world with an object tree (upper left of Scene editor)

A new object can be added to a scene by creating a new instance of a 3D model class. For example, in Figure 5.4 a new alien object is created from the Alice class, which is a 3D model in the Alice Gallery. When an object is added to a scene, the name of the object is automatically added to the **object tree** (upper left of the Scene editor), as shown in Figure. 5.5.



value type: Alien
name: alien2
initializer: new Alien

Figure 5.4 Creating a new Alien object from the Alien 3D model class

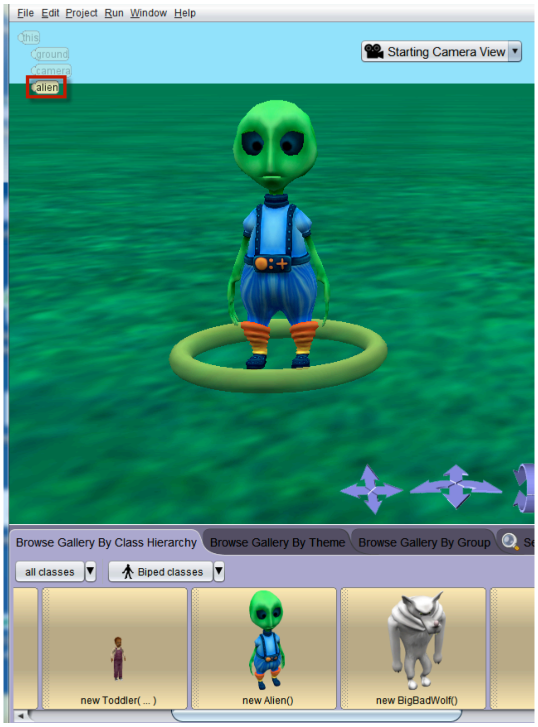


Figure 5.5 New Alien object has been added to the scene and the object tree

Class tree

In addition to the object tree (shown above in Figure 5.5) in the Scene editor, Alice 3 also maintains a class tree in the Code editor. The class tree can be viewed in a pull-down menu, as shown in Figure 5.6. In this example, the list of classes includes: Scene, Biped, and Alien. Alice projects always have the Scene class. Other classes in the list will vary depending on which objects are added to the scene and which preferences have been selected. In Figure 5.6, you may notice that the Alien class tile is indented beneath the Biped tile. This is because the Alien class is a specific type of Biped.

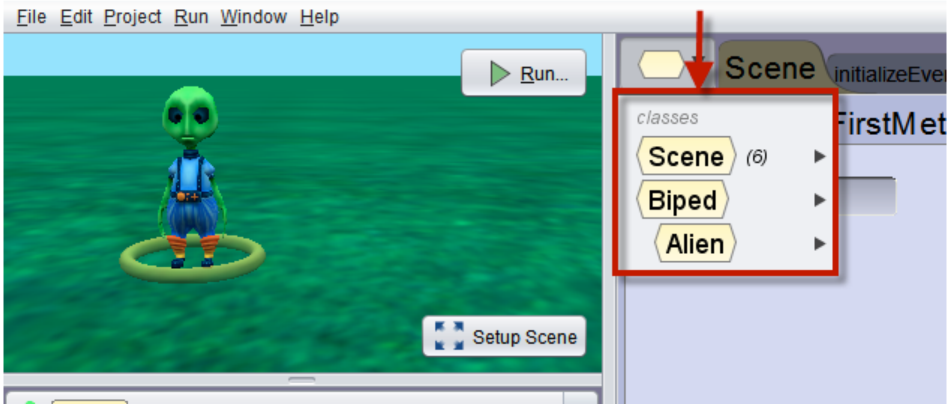


Figure 5.6 The class tree in the Class menu

Viewing a class file in the Code Editor

Selecting one of the classes in the class tree opens a class tab in the Code editor. A class tab displays an overview of the methods defined in that class. For example, Figure 5.7 illustrates a class tab for the Alien class. The class tab contains three components (procedures, functions, and properties).

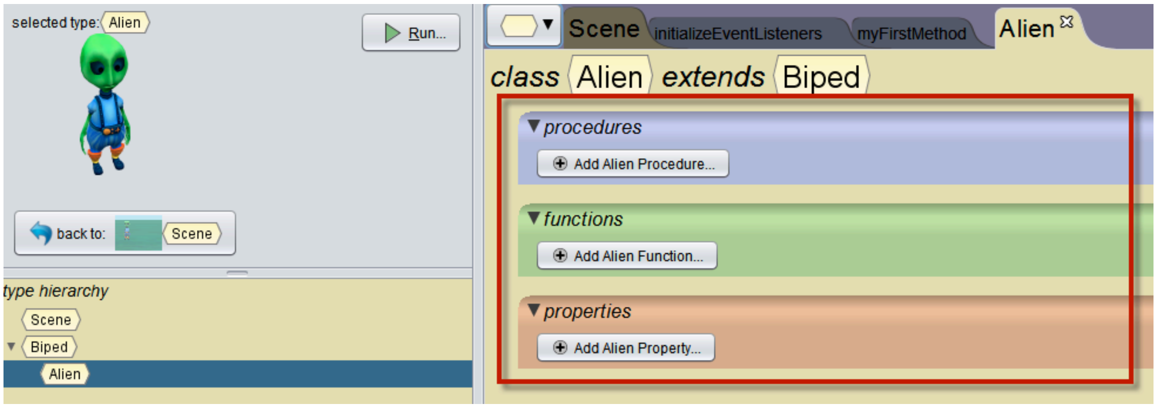


Figure 5.7 Alien class tab in the Code editor

If Constructors have been enabled in the preferences menu, a fourth component named constructors, is also displayed, as shown in Figure 5.8. A constructor is a special kind of method that contains instructions for creating a new object as defined by this class.

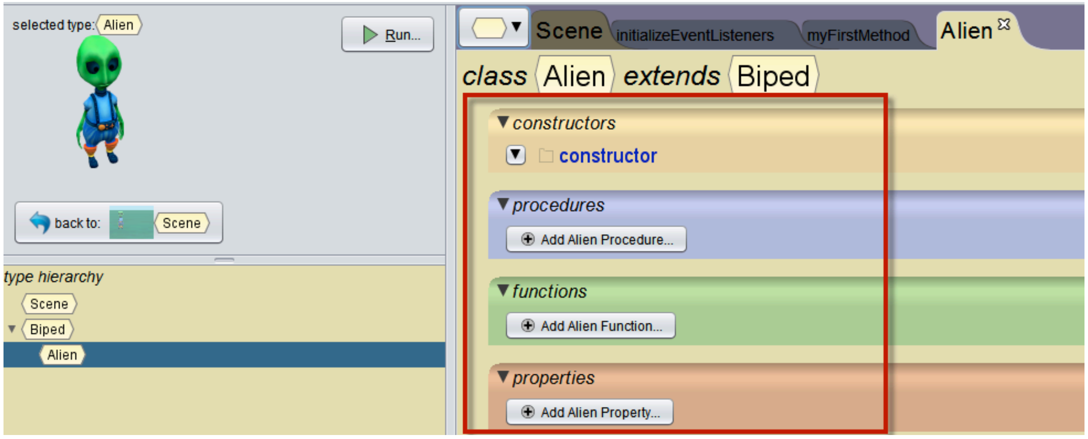


Figure 5.8 Method categories on a class panel

Although the Methods panel is normally displayed in the lower left corner of the Code editor window, when a class tab is opened in the Code editor, the Methods panel is replaced with a class hierarchy diagram, as illustrated in Figure 5.9.

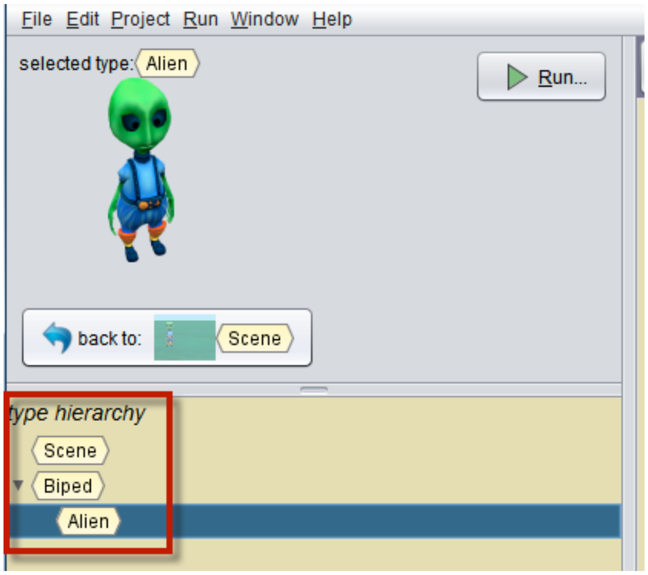


Figure 5.9 Hierarchy of classes in this project

Gallery Organization

The 3D model classes in the Gallery are organized into collections for the purpose of making it easy to find a specific model or type of model. (**Note:** New models are still being developed by members of the Alice team. Each update of Alice 3 will likely include new models. For this reason, screen captures in this How-To guide may occasionally vary from what is displayed on your computer.)

The Gallery has five tabs: three for browsing, one for searching, and one for shapes/text. Each of the three browsing tabs organizes the 3D models into collections:

Class Hierarchy – organized by mode of mobility, how an object "gets around" in a scene (for example, Biped, Flyer, Quadruped), as illustrated in Figure 5.10.

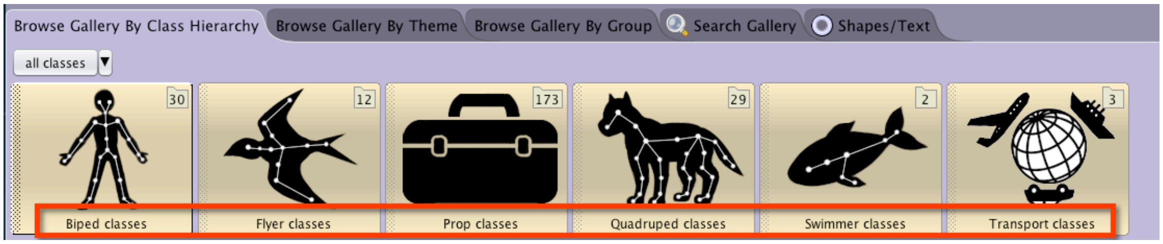


Figure 5.10 Browsing by Class Hierarchy

Theme – organized by region (for example, Amazon, Far East, Southwest) and by folklore context (for example, Fantasy, Wonderland), as illustrated in Figure 5.11.

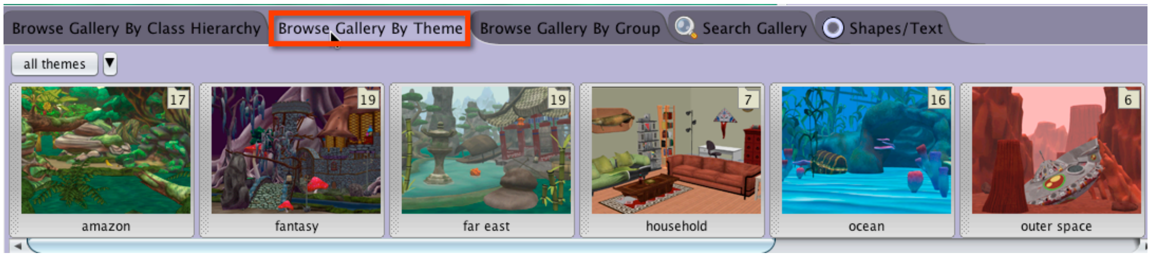


Figure 5.11 Browsing by Theme

Group – organized in common storytelling categories (for example, Animals, Characters, Scenery), as illustrated in Figure 5.12.

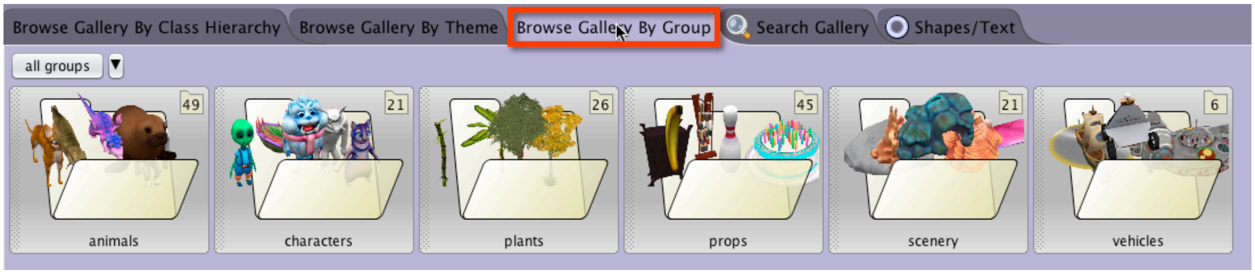


Figure 5.12 Browsing by Group

One way to think about browsing the gallery is that each collection is like a drawer in a file cabinet, as shown in Figure 5.13. A collection contains classes that share some common feature. For example, in the Class Hierarchy tab, the common feature is the mode of mobility -- how an object "gets around" in a scene. Bipeds walk on two legs, Quadrupeds walk on four legs, Flyers use wings, Swimmers use fins, and Vehicles move on wheels. (Props, not depicted in Figure 5.13 are stationary – do not move around on their own.)

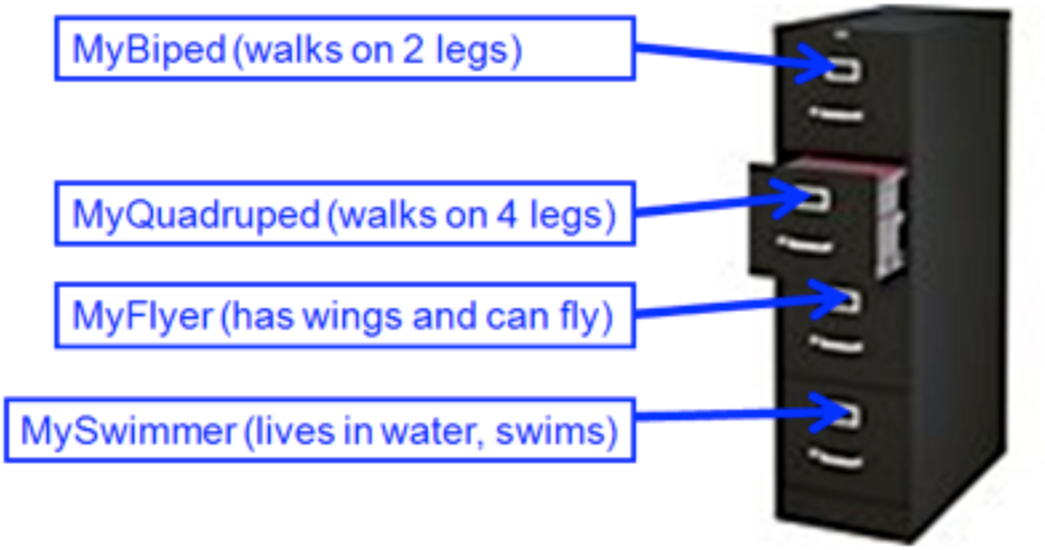

MyBiped (walks on 2 legs)
MyQuadruped (walks on 4 legs)
MyFlyer (has wings and can fly)
MySwimmer (lives in water, swims)

Figure 5.13 Common features used in Class Hierarchy tab

To view the classes in a collection, click on the icon for that collection. In the example shown in Figure 5.14, we clicked on the Flyer collection. A scroll bar at the bottom edge of the Gallery panel can be used to view the complete list of classes in this collection. These classes are in the Flyer folder because they each represent an entity that has two wings for flying and moving around the scene. Notice, however, that each has its own unique properties. For example, the ostrich has black and white feathers, the chicken has a comb, and the flamingo has long legs.

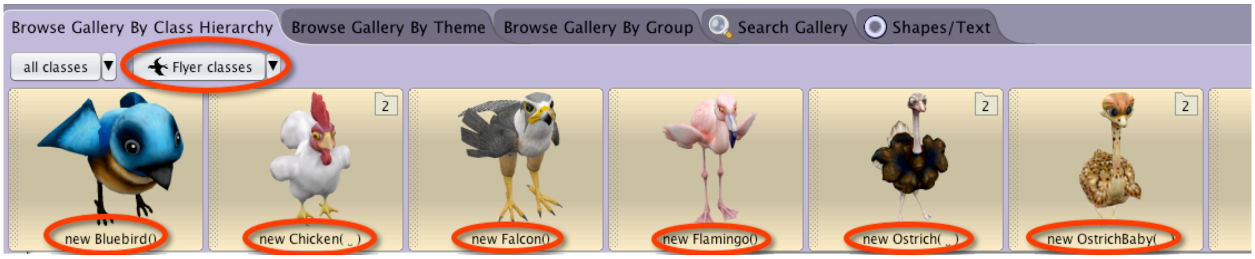


Figure 5.14 3D models in the Flyer folder

How to find a model in the Gallery

One way to find a specific 3D model in the Gallery is to take advantage of the organization system. In the Class Hierarchy tab, one would first think about how the desired object moves around…does it walk on two legs or four legs, or fly, or swim, or roll on wheels? Then, click that class folder and use the scroll bar to find the specific model. For example, to look for a Falcon, select the Flyer folder because a Falcon is likely to fly. Then, click the Falcon thumbnail sketch, as shown in Figure 5.15. Falcon objects belong to the Falcon class in the Flyer folder.

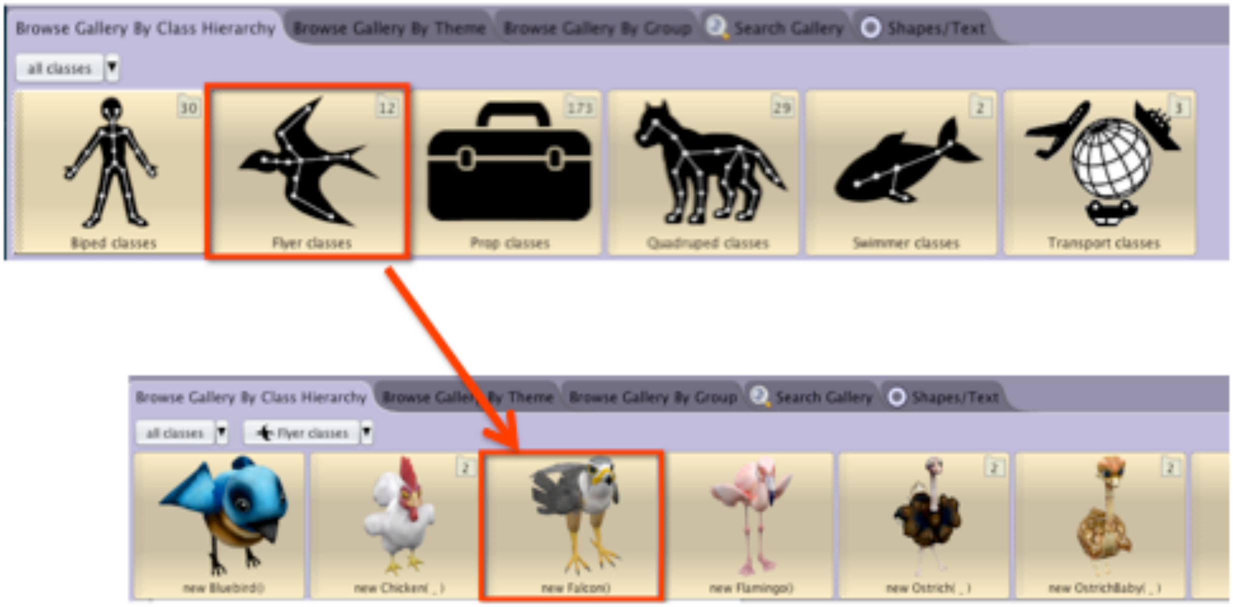
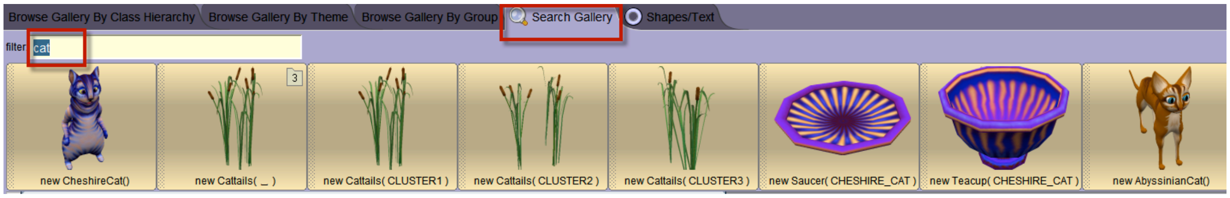


Figure 5.15 Finding a Falcon using Class Hierarchy organization

An alternate way to find a specific type of model is to use the Gallery's Search tab, as shown in Figure 5.16. To activate the search box, click the textbox on the tab. The mouse cursor should begin to blink in the box. Enter a descriptive word for an object. For example, in Figure 5.16, we started typing "cat" and Alice displayed models where "cat" is a significant part of the name. The more characters typed, the more Alice narrows down the possible matches.



filter cat

Figure 5.16 Using the search box

Shapes / Text

The last tab in the Alice Gallery provides 3D models for adding geometric shapes, 3D text, and billboards (importing 2D images) to the scene, as shown in Figure 5.17.

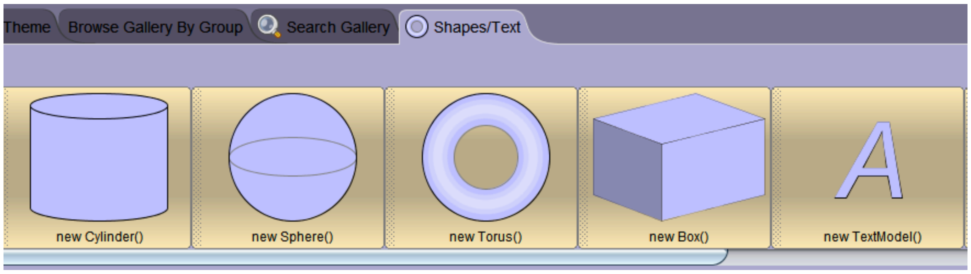


Figure 5.17 Shapes / Text in the Alice Gallery