

# Alice 3

## *How-to Guide*



Wanda Dann  
Don Slater  
Steve Cooper

Dennis Cosgrove  
Dave Culbya  
Laura Paoletti

---

*1st Edition*

Copyright: May, 2012

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 Pearson/Prentice-Hall publishers.

Cover artwork by Laura Paoletti, 2012.

Preface .....	4
Notes to Instructors.....	6
Exploring the Interface .....	7
How to download and install Alice 3.1 .....	7
A brief tour of the Alice 3 IDE .....	13
A brief tour of the Menu Bar .....	19
How to Set Preferences.....	29
Touring the Gallery.....	34
Setting up a Scene .....	41
Adding an object to a scene .....	41
How to set object properties in the Scene editor .....	54
How to set special effects in a scene.....	59
Marking and changing the camera's position .....	64
Positioning objects with markers.....	72
How to position and resize an object in the Scene editor.....	77
Positioning sub-parts in Scene editor.....	85
Relative positioning with camera viewpoints.....	89
How to align objects using a Snap grid .....	96
Precise positioning with one-shots .....	99
How to Cut, Copy, and Paste with the Clipboard .....	106

# 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.

## *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, Creative Professional

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

Wanda Dann, Carnegie Mellon University

Steve Cooper, Stanford University

Don Slater, Carnegie Mellon University

### *Acknowledgements*

Over the last four years, the Sun Microsystems Foundation, the Oracle 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 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).

We thank Tracy Dunkelberger Johnson (Pearson/Prentice Hall) for her patience and support.

### *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](http://www.alice.org)

As always, we welcome your comments and suggestions.

Wanda Dann, [wpdann@cs.cmu.edu](mailto:wpdann@cs.cmu.edu)

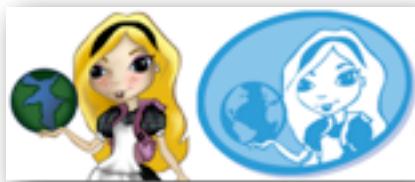
Steve Cooper, [coopers@cs.stanford.edu](mailto:coopers@cs.stanford.edu)

Don Slater, [dslater@cmu.edu](mailto:dslater@cmu.edu)

## NOTES TO INSTRUCTORS

Over the last decade, the Alice Project has emerged as a pioneer and a leading innovator in providing a software tool for teaching and learning problem solving and fundamental concepts of computer programming. Alice 2 has been the flagship for this effort. During the last four years, Alice 3 has been under development and is now in its first (out-of-beta) release.

Originally, we expected Alice 3 to replace Alice 2. This plan has been restructured. We are now providing an Alice Suite of tools composed of both Alice 2 and Alice 3, as shown in Figure 0.1.



*Figure 0.1 The Alice Suite: Alice 2.x and Alice 3.x*

# Exploring the Interface



## HOW TO DOWNLOAD AND INSTALL ALICE 3.1

The goal of this section is to provide information and instructions for downloading, installing, and starting Alice 3.1. Alice (all versions) is free and available for download at [www.alice.org](http://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.

## *Important Concepts:*

The Alice installer makes use of the Java JDK (Java SE Development Kit 7). 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](http://www.alice.org) homepage includes a Downloads menu. Click on Downloads on the menu bar, as shown in Figure 1.a.1. Select “Get Alice 3.1”.



Figure 1.a.1 Downloads menu on [www.alice.org](http://www.alice.org)

The Alice 3.1 webpage should be displayed, as shown in Figure 1.a.2.



Figure 1.a.2 The Alice 3.1 Download page

## *On-line Installer*

The preferred method for installation on a single machine having a broadband internet connection is with the On-Line Installer. Click the On-line Installer download button for the

appropriate platform (Windows, Mac, or Linux). The On-line Installer automatically activates a wizard that downloads files one at a time, as needed, to complete the installation. On broadband connections, this process typically takes 10 – 30 minutes, depending on the speed of the connection.

### *Off-line Installer*

The preferred method for installation on a single machine having a dialup internet connection is with the Off-Line Installer. To download the Offline Installer, click on the Off-Line Installer link for the appropriate platform (Windows, Mac, or Linux). The Off-line Installer automatically activates a download of the entire Alice 3.1 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 clicking on the downloaded file. A wizard will provide step-by-step instructions.

### *Universal zip*

The preferred method for installation on multiple platforms and/or on networked machines is with the Universal zip. 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 3.1 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 Vista or Windows 7, 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*

If Alice 3 has been properly installed using the On-Line or Off-Line installer, a shortcut icon will have been created on the desktop of the machine. The screen capture in Figure 1.a.3 was taken from the desktop of a PC, but Mac and Linux systems have similar icons. The icon may be pulled to a dock, if available on the computer system. Double click the icon to start Alice 3.1.



Figure 1.a.3 Icon on PC

If Alice has been installed on a machine using the Universal zip, 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.a.4. Four start files are highlighted in the red box in Figure 1.a.4.

Name	Date modified	Type
AliceApp	8/16/2012 13:27 AM	File Folder
application	8/16/2012 13:27 AM	File Folder
auxiliary	8/16/2012 13:27 AM	File Folder
end	8/16/2012 13:27 AM	File Folder
gallery	8/16/2012 13:27 AM	File Folder
lib	8/16/2012 13:27 AM	File Folder
<b>Alice3.bat</b>	8/30/2012 1:40 PM	Windows Batch File
<b>alice3.sh</b>	8/30/2012 1:40 PM	SH File
<b>Alice3Icon.ico</b>	8/30/2012 1:40 PM	Icon
<b>alice364bit.sh</b>	8/30/2012 1:40 PM	SH File
EUUA_Alice3DM	8/30/2012 1:40 PM	Text Document
EUUA_LookingGlassWalkthroughs.inf	8/30/2012 1:40 PM	Text Document
EUUA_The Sims 3/MediaAsset.inf	8/30/2012 1:40 PM	Text Document
install.exe	8/30/2012 1:39 PM	EXE File

Figure 1.a.4 Start files in the Alice 3.1 Universal version

These start files are designed to start Alice 3.1 on the specific operating system (OS) as is appropriate. To start Alice 3.1, click on the appropriate start file for the OS installed on the computer system, as described here:

- |                       |  |
|-----------------------|--|
| <b>Alice3.bat</b>     | Starts Alice 3 for a PC system (either 32 or 64 bit) |
| <b>alice3.sh</b>      | Starts Alice 3 for a Linux system, 32 bit            |
| <b>alice364bit.sh</b> | Starts Alice 3 for a Linux system, 64 bit            |

**Alice3Icon.ico**

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

*Welcome dialog box*

When Alice starts, a Welcome dialog box is displayed, as shown in Figure 1.a.5. The Welcome 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 a scene. If the Cancel button in the dialog box is clicked without selecting a template or a previously written project, Alice will close the Welcome dialog box without displaying the Code editor. To reopen the Welcome dialog box, click File in the menu bar at the top left of the Alice window and select New from the menu.

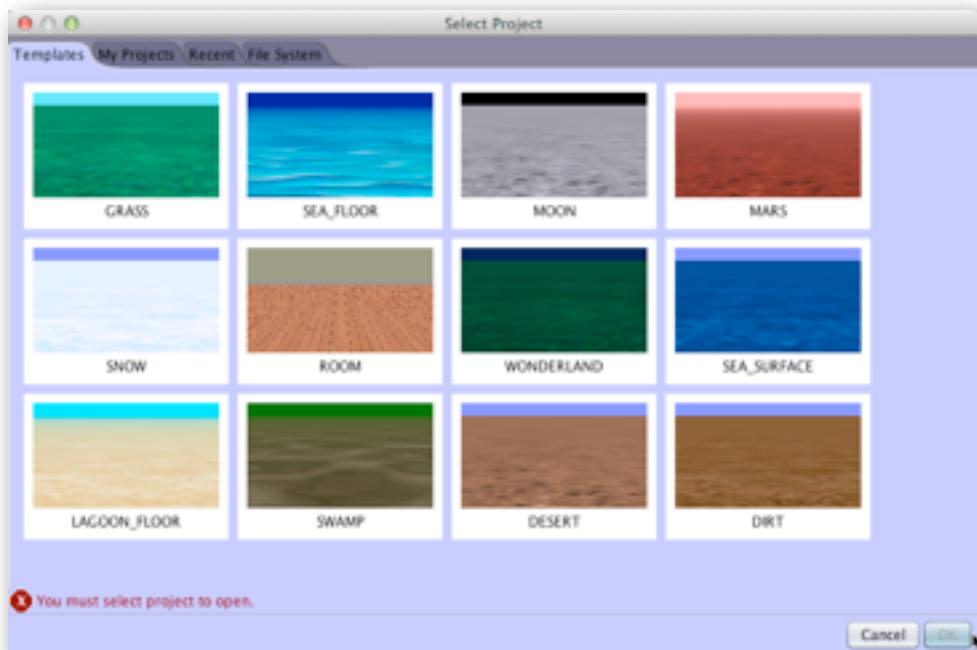


Figure 1.a.5 Welcome dialog box

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.a.6. (The display may vary somewhat, but the basic organization should be the same.)

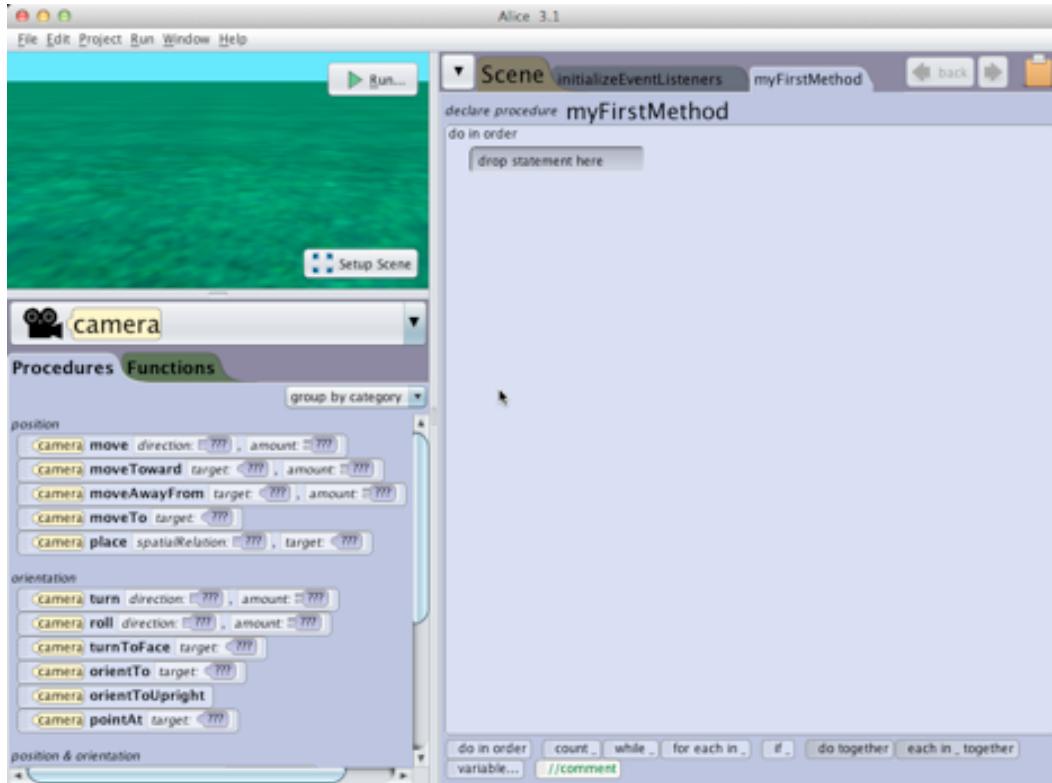


Figure 1.a.6 Start Screen, Code Editor

### 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>, Updating video drivers for Windows machines.

# 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 code editor and the scene editor in the Alice Interactive Development Environment (IDE). The components are briefly described and screen shots identify the individual components. Later chapters will provide greater details and demonstration examples. The Alice 3 components include:

- **Welcome dialog box:** Select a template for the scene
- **Code editor:** Scene view, Editor tabs, Control tiles, Methods panel
- **Scene editor:** Scene view, Handles palette, Setup Panel, Gallery

### *Selecting a template in the Welcome dialog box*

When Alice starts, select a template for a project scene by clicking on one of the template thumbnail images in the Welcome dialog box. In Figure 1.b.1, select a template by a single-click on the template and then click OK. It is also possible to just double-click on the template.

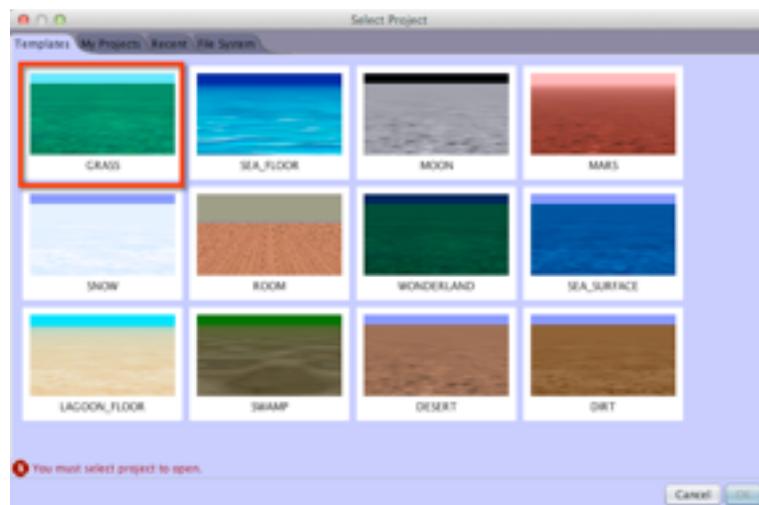


Figure 1.b.1 Select a template

### *Initial display window and menu bar*

Upon selection of a template for the scene, Alice will display the scene in the upper left corner of the window, as shown in Figure 1.b.2. 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 a view of the scene (upper left), the opening interface displays the Code editor 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 Control tiles panel (lower right), as labeled in Figure 1.b.3. When Alice is first started with a new template, the camera is the selected object, the Scene view displays the selected template scene, and myFirstMethod (the main method defined for a scene) is the open tab in the Code editor panel.

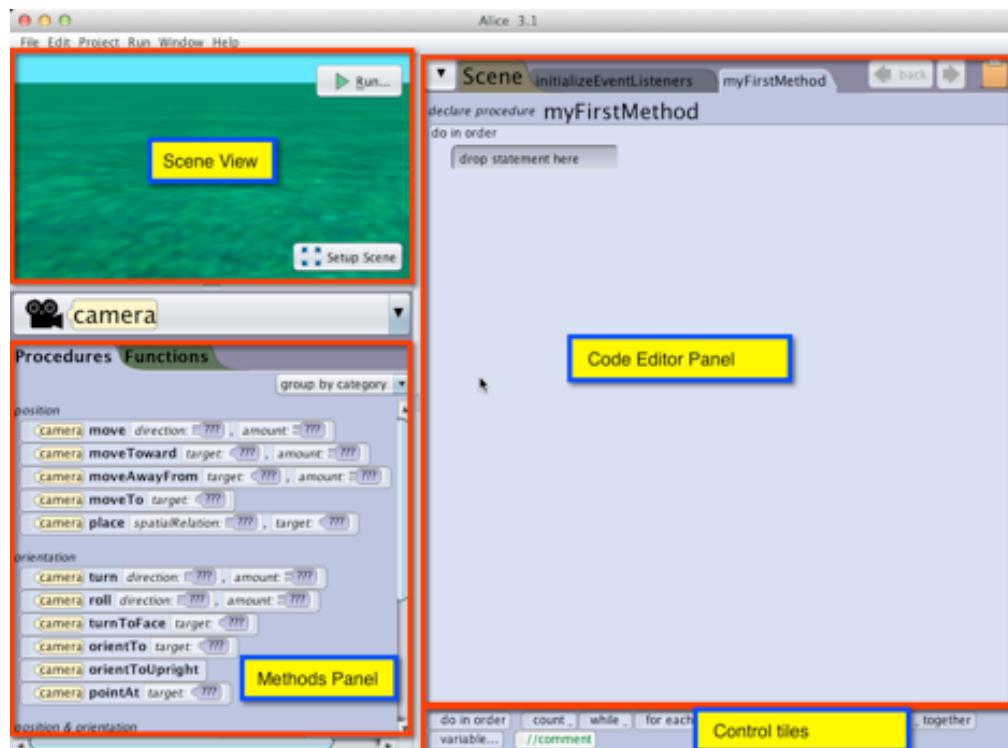


Figure 1.b.2 Alice 3 interface: Code editor panels

## *Methods: Filtered into procedures, functions, properties*

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) in a scene. As shown in Figure 1.b.3, the Methods panel categorizes methods for display on two tabs: Procedures (methods that perform an action), and Functions (methods that ask a question). In this example, the camera object's Procedures tab displays methods such as move, moveToward, ..., turn, roll, and others.



Figure 1.b.3 Methods: Procedures and Functions

### Control panel

In the Control panel, each tile represents a statement that organizes and manages instructions and information (data). Figure 1.b.4 highlights the control tiles. Most control tiles specify the order in which to perform methods. As a convenience, a comment tile (//) and a variable tile (for storing data) are also on the Control panel.

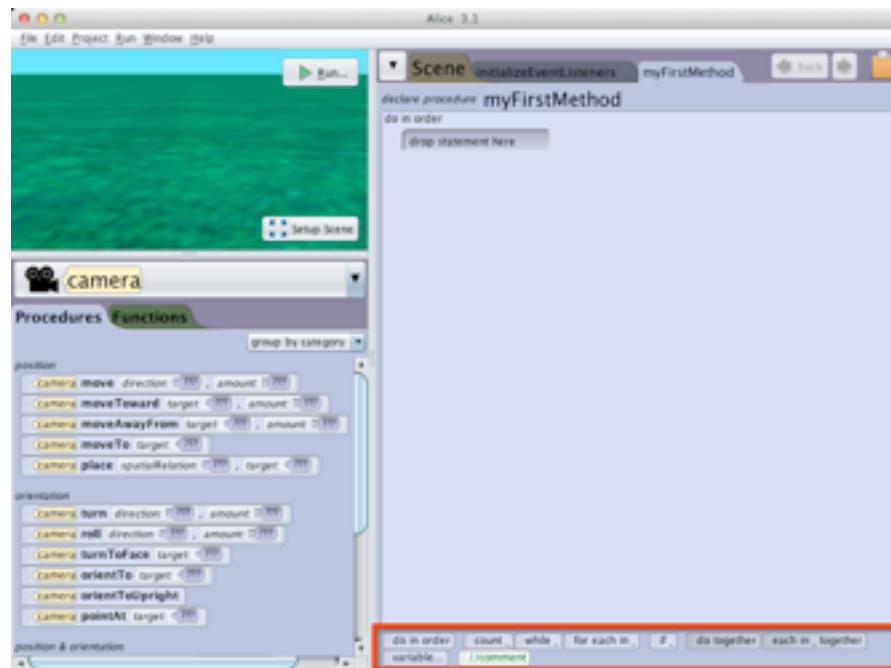


Figure 1.b.4 Control tiles

Overall, the Code editor provides a drag-and-drop environment. Method and control tiles are dragged into the editor to create 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 1.b.5.

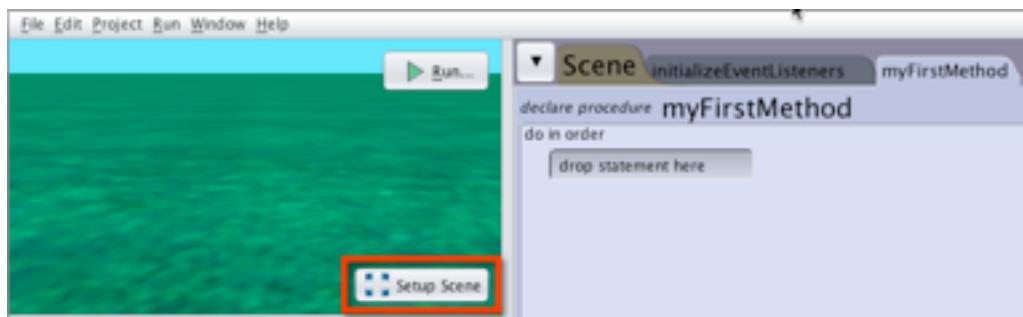


Figure 1.b.5 Setup Scene button switches to Scene editor

As labeled in Figure 1.b.6, 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 provides buttons for positioning objects in the scene and changing the size, color, and other properties of these objects.



Figure 1.b.6 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 1.b.7.

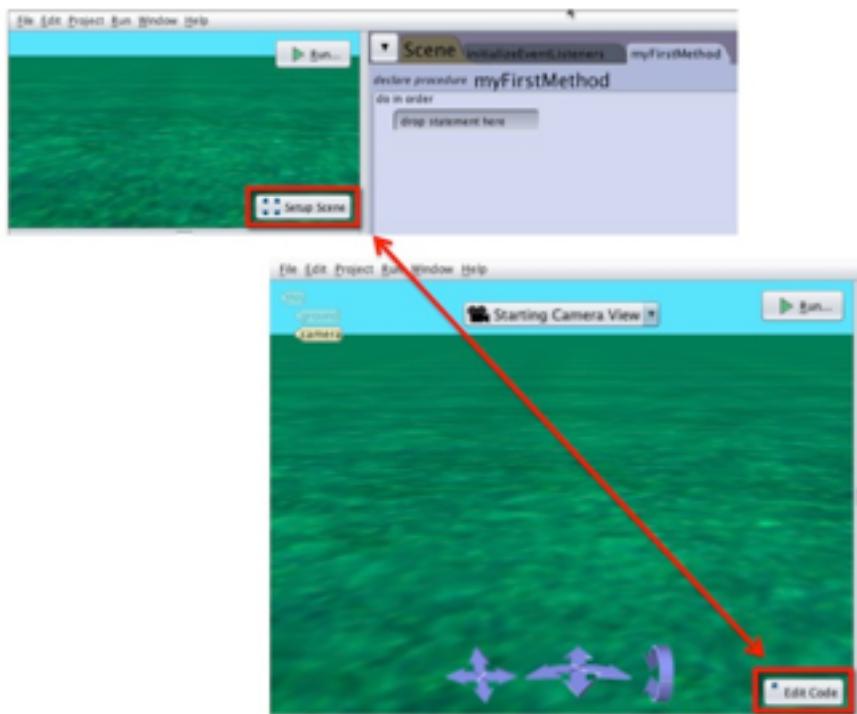


Figure 1.b.7 Toggle between Code and Scene editors

## 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 is provided. Some items are specific to the operation of Alice, in which case we provide a brief description and illustration of the items.

In Alice 3, a menu bar is displayed in the upper left corner of the window, as shown in Figure 1.c.1. The menus include: File, Edit, Project, Run, Window, and Help.



Figure 1.c.1 Scene view and Menu bar

### *File menu*

The File menu contains options for managing and editing files in a project, as shown in Figure 1.c.2. The items in the File menu are: New, Open, Recent Projects, Save, Save As, Revert, Print, and Exit.

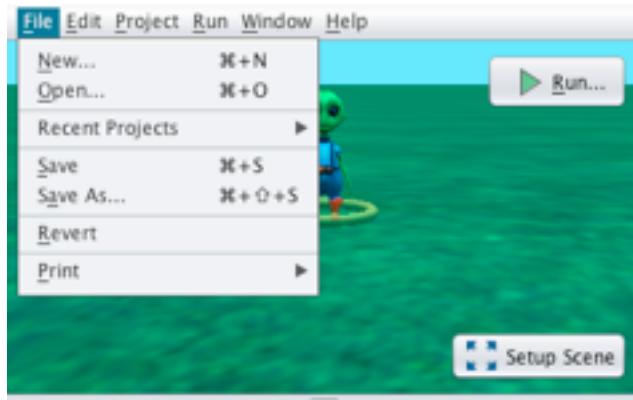


Figure 1.c.2 Menu for managing files

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 edited Alice 3 projects that have been saved.
- **Revert** restores the 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.
- The **Print** menu item is available for printing program code. As shown in Figure 1.c.3, the Print menu item cascades to three choices for 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 that is part of the Alice system.) Print Current Code will print only the code in the method currently being edited in the Code editor. Print Scene Editor will print a screen capture of the Scene Editor, including a screen capture of the Scene View and the Setup panel but not including the Gallery panel.

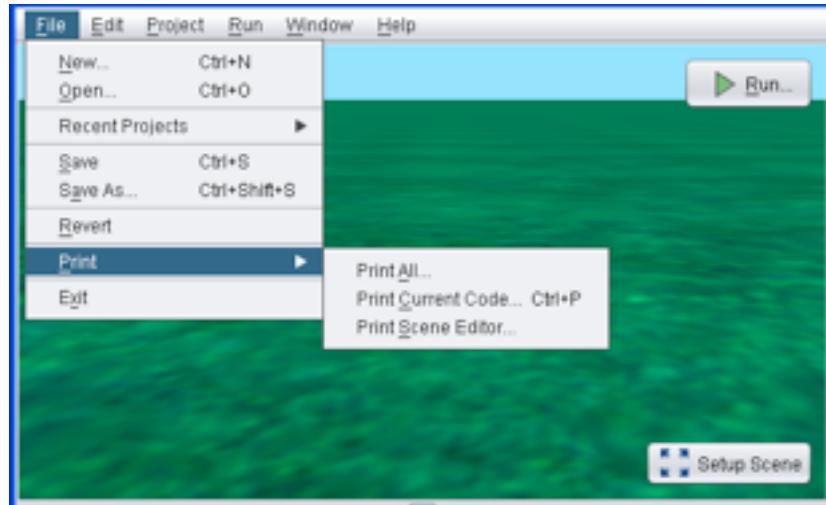


Figure 1.c.3 Print options in the File menu

### Edit menu

The edit menu contains Undo, Redo, Cut, Copy, and Paste, as shown in Figure 1.c.4. These are all standard editing actions. As of this writing, Cut, Copy, and Paste are not implemented but are listed in the menu to provide for future modifications. Truthfully, the traditional cut, copy, and paste actions work well in a text editing environment. However, these actions are of limited usefulness in an editor containing graphic statement tiles. Section 16 of this guide provides detailed instructions on using the clipboard for Cut, Copy, and Paste in a drag-and-drop programming environment.

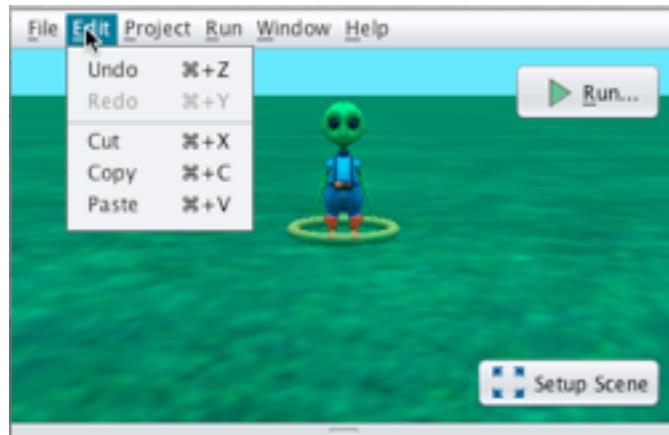


Figure 1.c.4 Menu for Edit options

### Project menu

The project menu contains Manage Resources, Find, and Statistics. The Manage Resources item opens a dialog box for importing (or removing) resource files. A resource file may be either

an audio (sound) or image (picture) file. Alice does not provide sound or image editing capabilities.

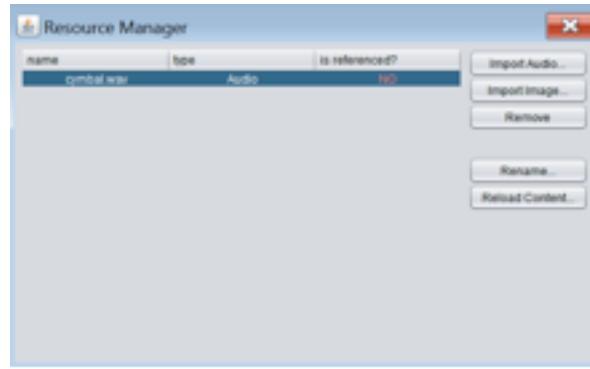


Figure 1.c.5 Resources dialog box

The Find item pops up a dialog box for searching the program code to find where a method tile is used. In Figure 1.c.6, the scene's method named setActiveScene has been selected in the Find dialog box. When setActiveScene is selected in the Find dialog box, a message appears in the column on the right to show that a tile has been found in the setActiveScene program's main method. Also, the main method is opened in a new tab for viewing the setActiveScene tile within it.

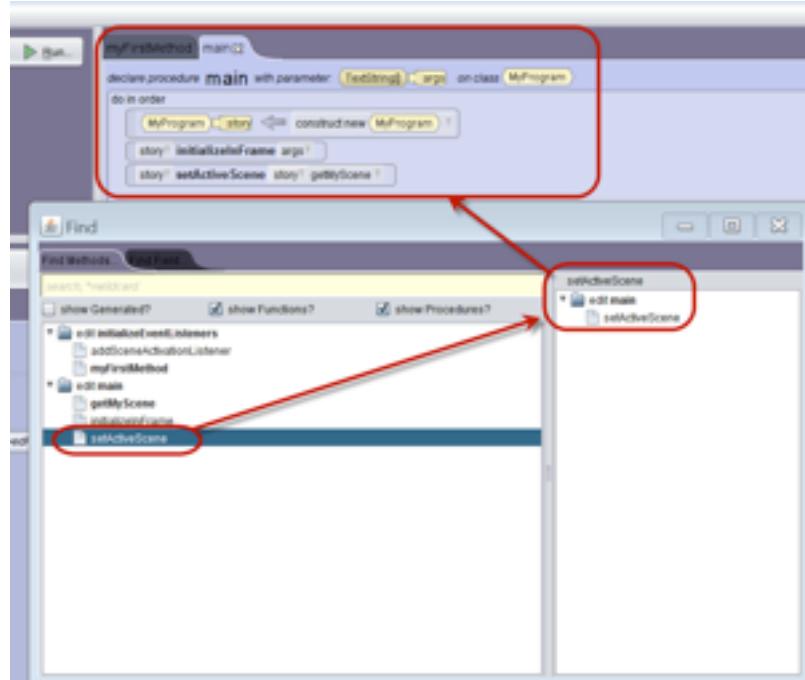


Figure 1.c.6 Find where a method tile is used in the program

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 1.c.7.

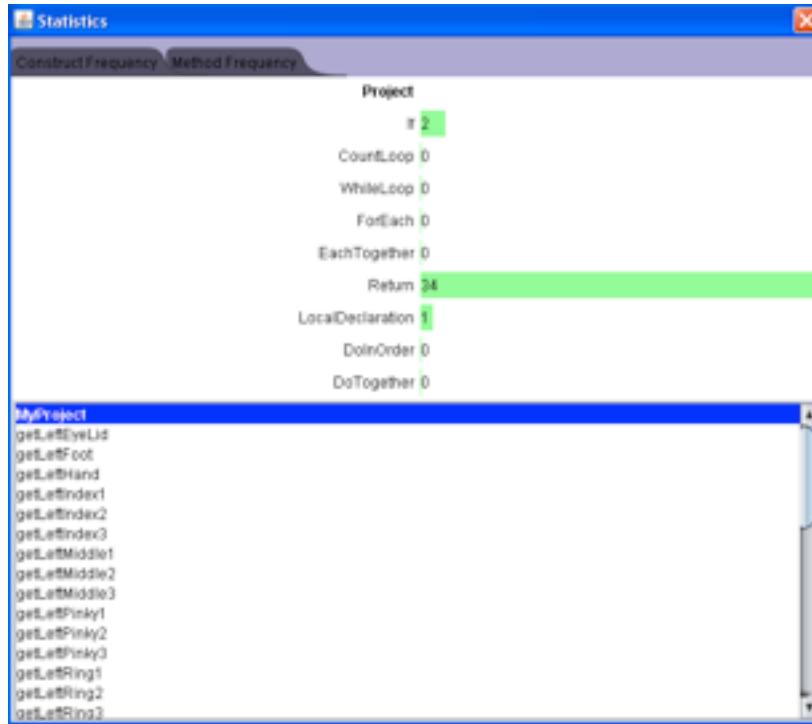


Figure 1.c.7 Statistics popup window

## Window menu

The Window menu contains Perspectives, Project History, Memory Usage, and Preferences, as shown in Figure 1.c.8. These items control the display of the Alice 3 environment in terms of the number of open windows and their content.



Figure 1.c.8 Window menu

The Perspectives menu provides a choice of Code editor or Scene editor display, as shown in Figure 1.c.9. This menu option is an alternative to using the toggle buttons.

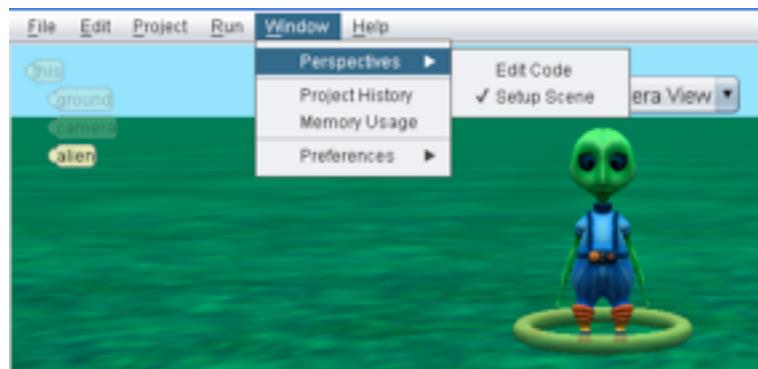
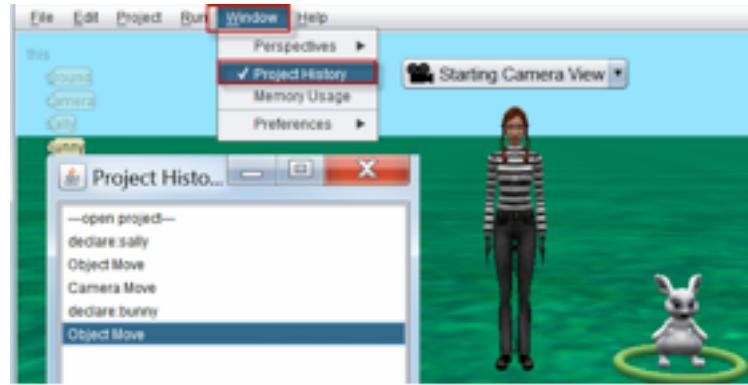


Figure 1.c.9 Perspectives

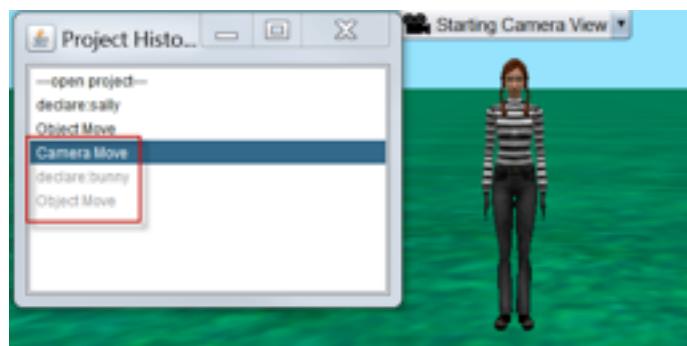
The Project History item opens a new window containing a list of all actions performed (thus far) in the current editing session. Figure 1.c.10 shows a Project History window in which sally (a Sims2 character) is declared (added to the scene), sally is moved in the scene, the camera is moved, a bunny is added to the scene, and the bunny is moved in the scene. 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.



*Figure 1.c.10 View project history*

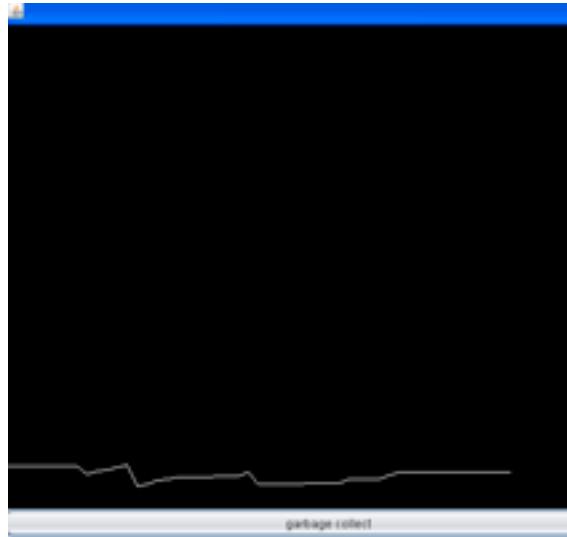
### *Backtrack In History*

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 list, as shown in Figure 1.c.10 above. Selecting an item (other than the last one) causes all later actions in the list to be “played backwards.” To illustrate, we clicked Camera Move in the Project History and the two later action items (declare bunny and Object Move) were played backwards. So, the bunny moved back to its initial location and then was deleted from the world. The state of the world when Camera Move was executed is now displayed in the scene view shown in Figure 1.c.11. Notice, in Figure 1.c.11, that the declare bunny and Object Move actions are still in the history and clicking on one of these will track forward to that state. However, if a middle state is selected, then the history box is closed, and some other action is performed, the project history is updated and later states are removed from the history.



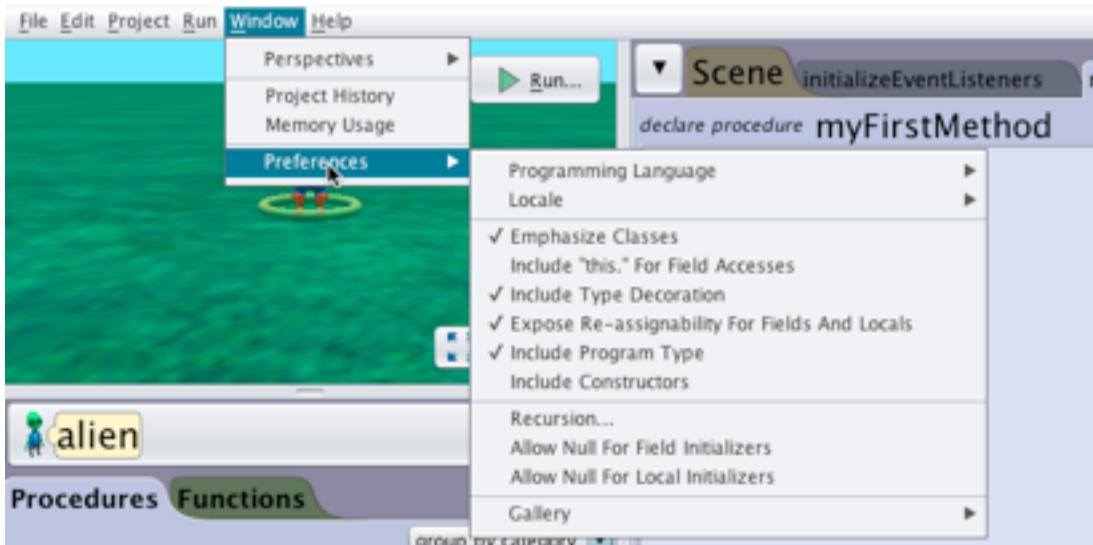
*Figure 1.c.11 Backtracking to a previous state*

The Memory Usage item opens a popup window in which memory usage is tracked, as shown in Figure 1.c.12. An alert is displayed in the window's task bar when Java's garbage collection is in progress.



*Figure 1.c.12 Memory usage window*

The Preferences item opens a cascading menu for setting preferences in the Alice 3 environment, as shown in Figure 1.c.13.



*Figure 1.c.13 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 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).
- Emphasize Classes: Display a Class Panel in the Code editor. Class panels are particularly useful to those preparing to learn a production level language such as Java.
- Include "this." For field access: Display "this." in program code in the Code editor. The word "this" represents the currently selected object (which can be the scene or an object within the scene).
- Include Type Decoration: Display decorative symbols on method tiles to indicate the type of entity represented (for example, an object, number, string, Boolean, and so forth).
- Include Program Type: Include Program in the Class menu list. Program contains the main method, which is the first method executed when the Run button is clicked.
- Include Constructors: Add an edit constructor option to the Class tab for each type of object in a scene.
- Recursion: Enable the use of recursion in Alice programs.
- 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 1.c.14 shows the dialog box with (left) and without (right) a preview of the declaration statement.



Figure 1.c.14 Preview on (left) and off (right) 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 1.c.15.

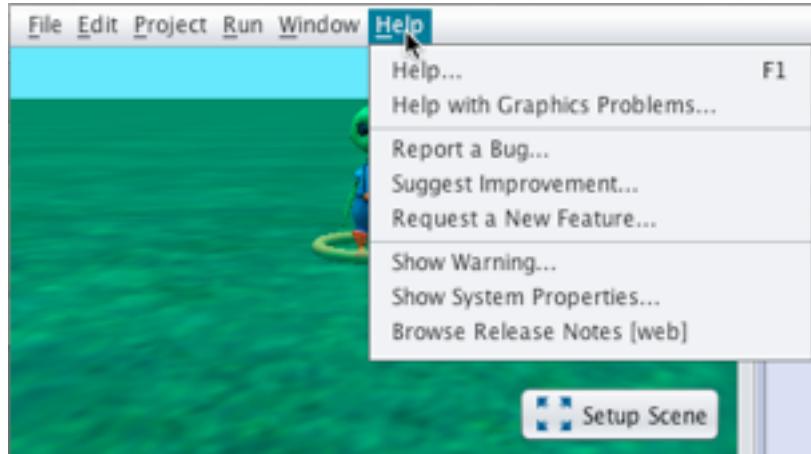


Figure 1.c.15 Help menu

The Help... item opens a window containing a link to the help page at alice.org, as shown in Figure 1.c.16.

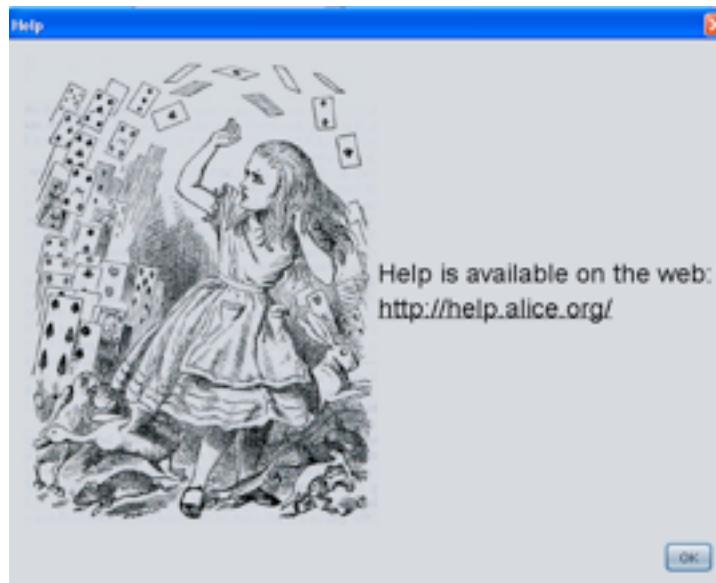


Figure 1.c.16 Help link

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. When the submit button is clicked, the report is immediately submitted to the reports database if the computer is currently connected to the internet. If not connected to the internet, the report is simply deleted when Alice is shut down.

A screenshot of a Windows application window titled "Report Issue". The window has a blue header bar with the title and standard window controls. The main area contains a logo for "Alice BUGS" with the URL "http://bugs.alice.org:9050/" below it. A "Log in... (Optional)" button is in the top right corner. The form fields are as follows:

- visibility:** Radio buttons for "Public" (selected) and "Private (Only Visible to the Alice Team)".
- Type:** A dropdown menu set to "BUG".
- summary:** A text input field with placeholder text "please fill in a one line synopsis (Required)".
- description:** A large text input field with placeholder text "please fill in a detailed description".
- steps:** A text input field with placeholder text "please fill in the steps required to reproduce the bug".
- environment:** A text input field showing system details:

```
java.version: 1.6.0_21  
os.name: Windows XP  
os.arch: x86  
os.version: 5.1  
sun.arch.data.model: 32
```
- attachment:** Two radio buttons: "Yes, attach my current project as it is relevant." and "No, do not attach my project."
- submit bug report:** A large blue button at the bottom.

Figure 1.c.17 Bug report form

## 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. We say these are default preferences. The default preferences are shown in Figure 1.d.1. By default, most of the preference options are disabled (not checked). However, one preference option in this menu is selected: Constants, which enables the ability to create a value that is constant (does not change value at runtime).

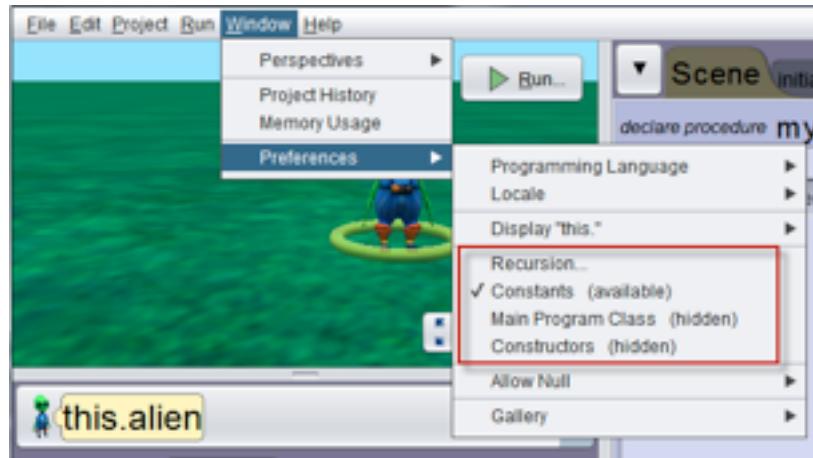


Figure 1.d.1 Default preference settings

With the default preference settings, Alice starts with a pull-down menu button and three tabs in the Code editor, as shown in Figure 1.d.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 (the runtime window) and the code in `myFirstMethod` will be executed (run).

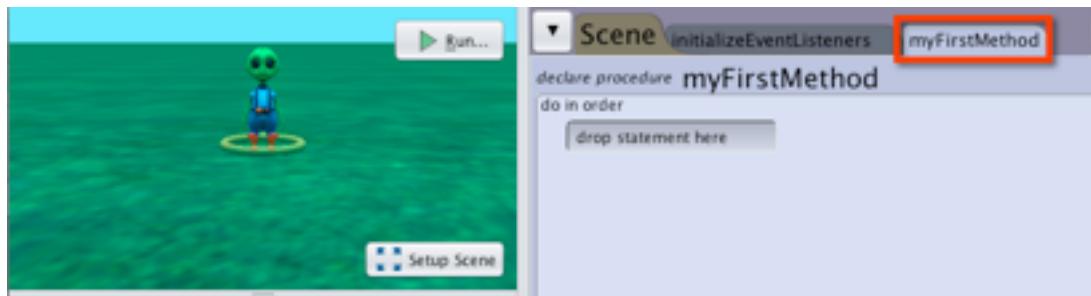


Figure 1.d.2 `myFirstMethod` tab in the Code editor

*Cascading preference menus*

Some preference menu items cascade to a subset of selections. For example, Display “this.” cascades to two options, as shown in Figure 1.d.3. In the screen capture shown here, “this.” is always on. Use of the term “this.” is a style choice and you may elect to turn it off in the methods for the Scene class. To select the alternate option click the second item in the cascaded menu, as shown in Figure 1.d.4.

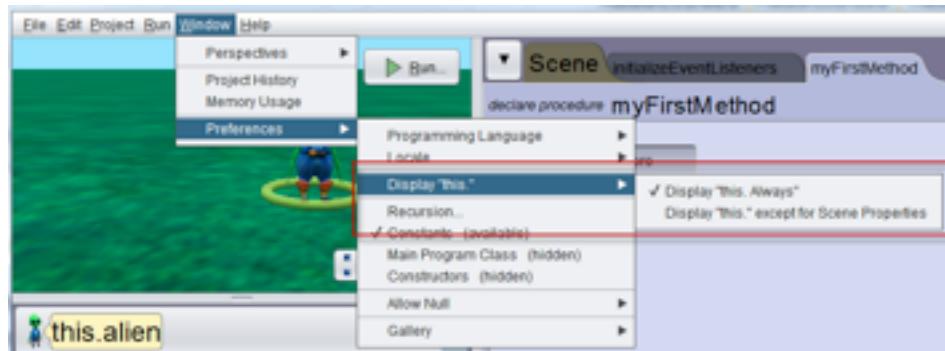


Figure 1.d.3 “this.” is always displayed

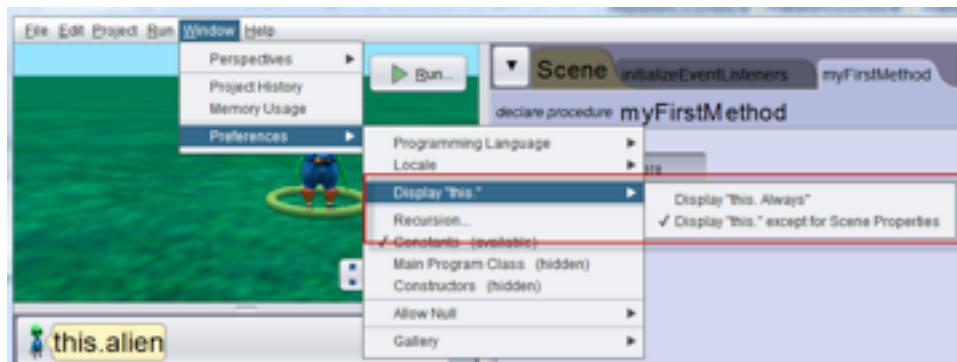


Figure 1.d.4 “this.” Is turned off in the Scene class

### *Setting multiple preferences*

Any combination of preferences may be set. Figure 1.d.5 shows two preferences (Constants and Constructors) we recommend 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.

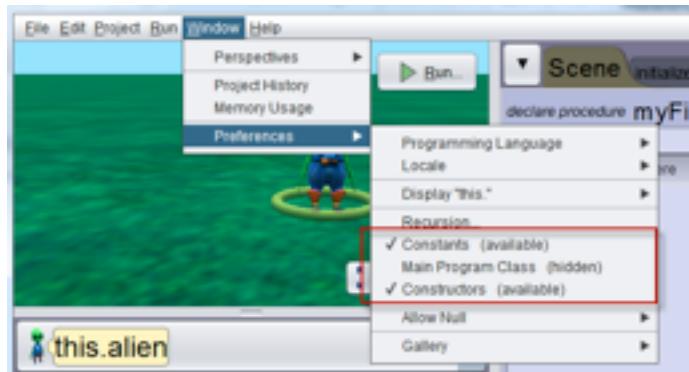


Figure 1.d.5 A selection of two preferences

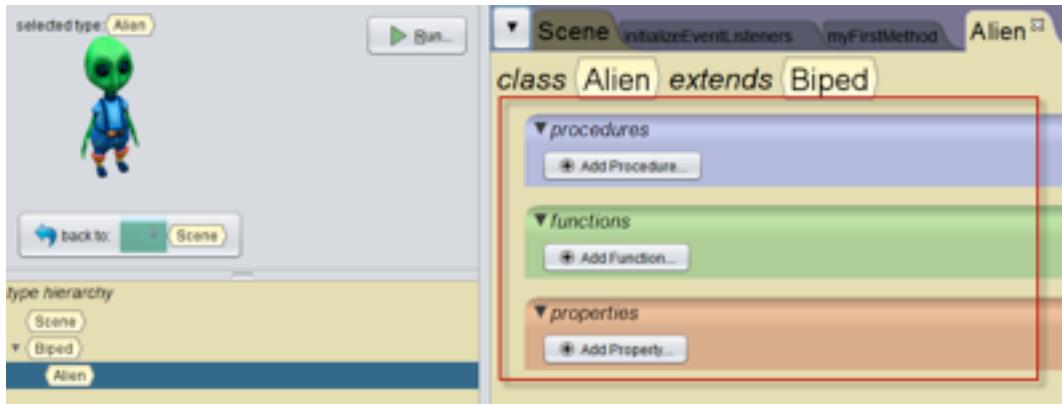
### Class selection button

When clicked, the class selection button (shown in Figure 1.d.6) will pull down a menu with a list of the classes currently used in this world. In this example, the list of classes includes: Scene, Biped, and Alien. Alice projects will 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. 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.



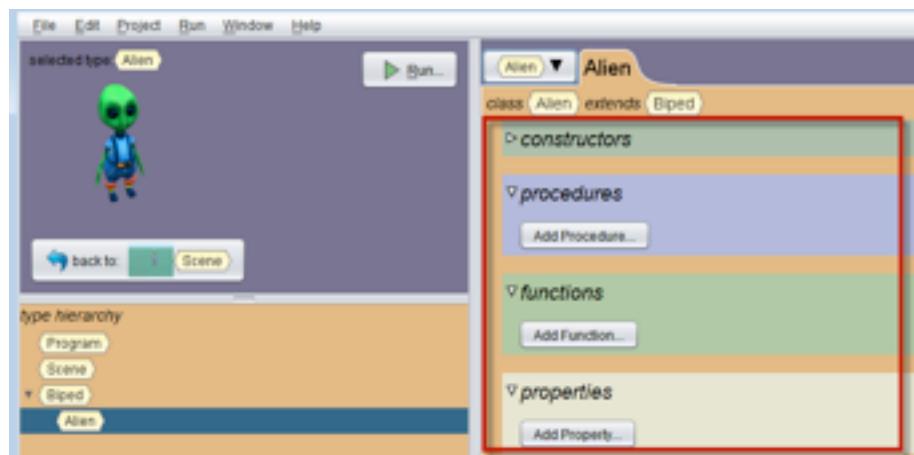
Figure 1.d.6 The class selection menu

Selecting one of the classes in the menu causes a class tab to be displayed in the Code editor. A class tab displays an outline of the methods defined in that class. For example, Figure 1.d.7 illustrates a class tab for Alien. Methods listed on a class tab are subdivided into three groupings (procedures, functions, and properties).



*Figure 1.d.7 Alien class tab in the Code editor*

If Constructors have been enabled in the preferences menu, a fourth group named constructors, is also displayed, as shown in Figure 1.d.8. A constructor is a special kind of method that contains instructions for creating a new object as defined by this class.



*Figure 1.d.8 Method categories on a class panel*

A Methods panel is normally displayed in the lower left corner of the Code editor window. But, when a class tab is selected in the Code editor, the Methods panel is replaced with a diagram showing where this class fits in the hierarchy of classes in this project. Figure 1.d.9 shows the hierarchy of classes in this example.

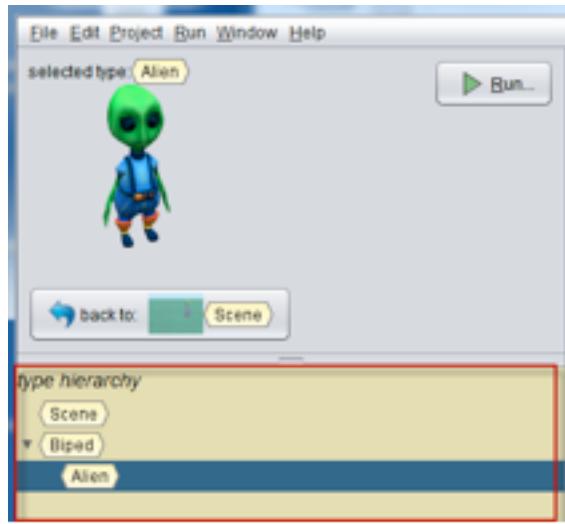


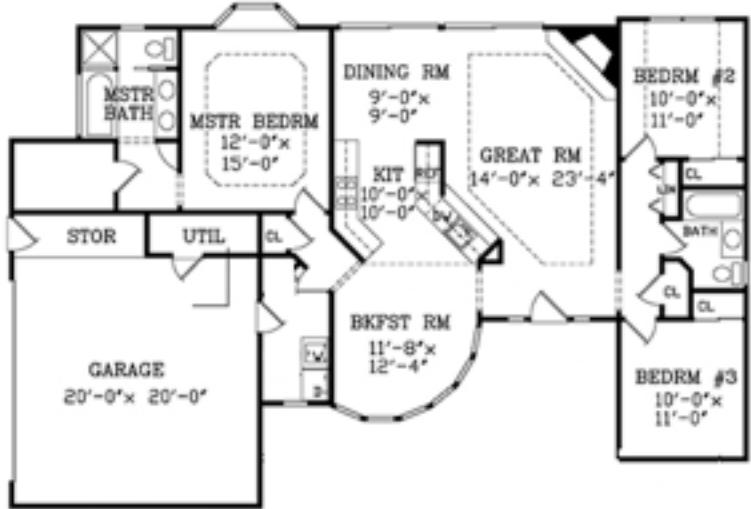
Figure 1.d.9 Hierarchy of classes in this project

## TOURING THE GALLERY

The purpose of this section is to introduce the collections of 3D models available in the Gallery. In particular, we look at why 3D models in the Gallery are in classes and how these classes are organized into collections. Our goal is to make it easy to find different kinds of models or to find a specific model in the Gallery. Please note that 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 document may occasionally vary from what is displayed in future releases.)

### 3D Models and Classes

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 2.a.1 illustrates a blueprint for house. This blueprint provides a design that tells a home-builder how to build the house but is not an actual physical instance of a house.



*Figure 2.a.1 A blueprint for constructing a house*

In animation film studios such as Disney, Pixar, and DreamWorks, a 3D model is a digital representation of an entity (someone or something). Animation adds motion to a model. In Alice, a class puts together a digital representation of an entity, a plan for constructing it, and instructions for animating it, all in the same jar. A more general definition is: A class defines a type of object (a modeled entity) and actions that can be performed by that object. The Gallery contains classes for creating and animating objects in an Alice virtual world.

Figure 2.a.2 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. Each object is an instance of a class in the Gallery. The scene is an object of the Scene class, the ground is an object of the Ground class, and the camera is an object of the Camera class.



Figure 2.a.2 A new Alice world with the default Object tree

## Gallery Organization

In an Alice world, many objects are added to a scene and then animated to tell a story, create a simulation, or play a game. This means that dozens of 3D models (classes) are needed in the Gallery. We expect that, over time, the Gallery will continue to grow to include hundreds of models.

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 2.a.3.



Figure 2.a.3 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 2.a.4.



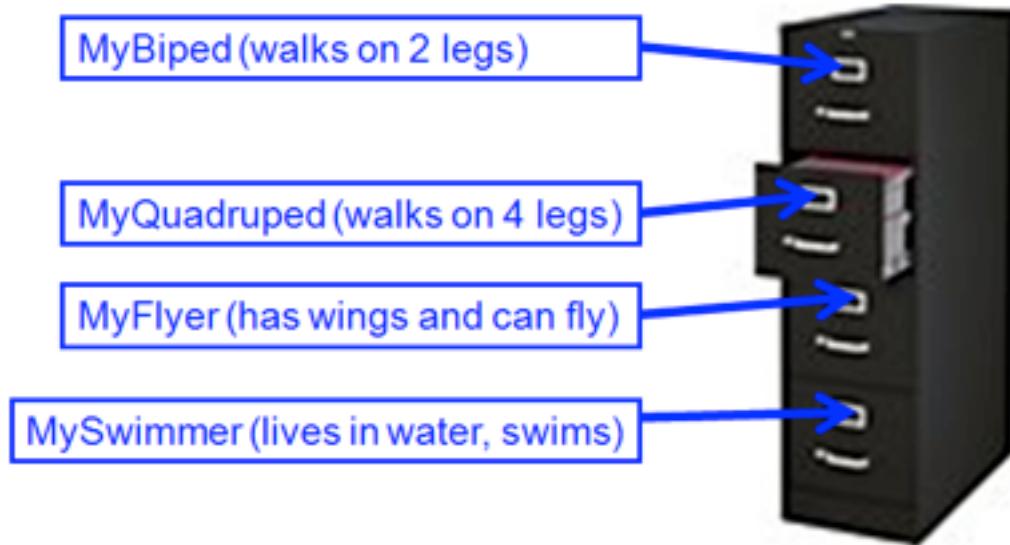
Figure 2.a.4 Browsing by Theme

**Group** – organized in common storytelling categories (for example, Animals, Characters, Scenery), as illustrated in Figure 2.a.5.



Figure 2.a.5 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 2.a.6. 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. (Props, not depicted in Figure 2.a.6 are stationary – do not move around on their own.)



*Figure 2.a.6 Common features used to categorize 3D models in Class Hierarchy tab*

To view the classes in a collection, single-click on the icon for that collection. In the example shown in Figure 2.a.7, we clicked on the Biped 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 Biped folder because they each represent an entity that has two legs for walking and moving around the scene. Notice, however, that each has its own unique properties. For example, the alien has green skin, the wolf has a long nose, and the bunny has long ears.

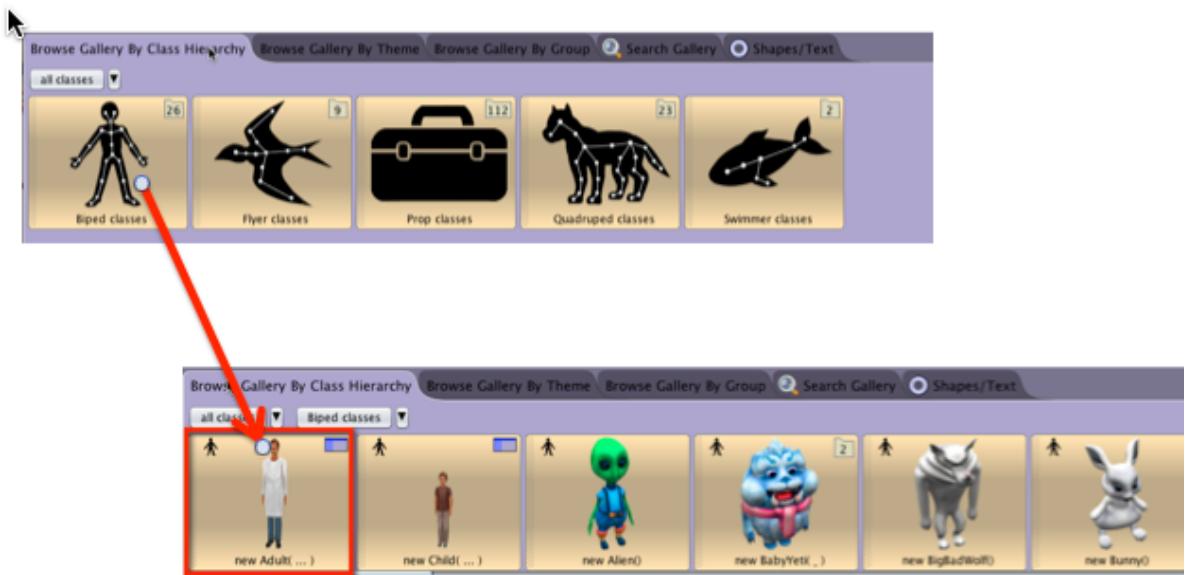


*Figure 2.a.7 3D models in the Biped folder*

### *How to find a model in the Gallery*

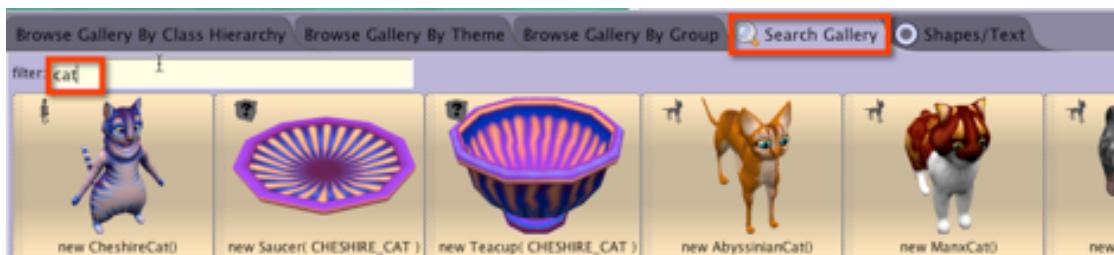
One way to find a specific 3D model in the Gallery is to take advantage of the organization system. In the Class Hierarchy tab, one would first think about how the desired object moves around...does it walk on two legs, walk on four legs, fly, or swim? Then, click that class folder and use the scroll bar to find the specific model. For example, to look for a Sims person, select the Biped folder because a person is likely to move around on two legs. Then, click the Person

thumbnail sketch, as shown in Figure 2.a.8. Sims people objects belong to the Person class in the Biped folder.



*Figure 2.a.8 Finding a model using Gallery organization*

An alternate way to find a specific type of model is to use the Gallery's Search tab, as shown in Figure 2.a.9. 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 2.a.9, 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.



*Figure 2.a.9 Finding a model 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.

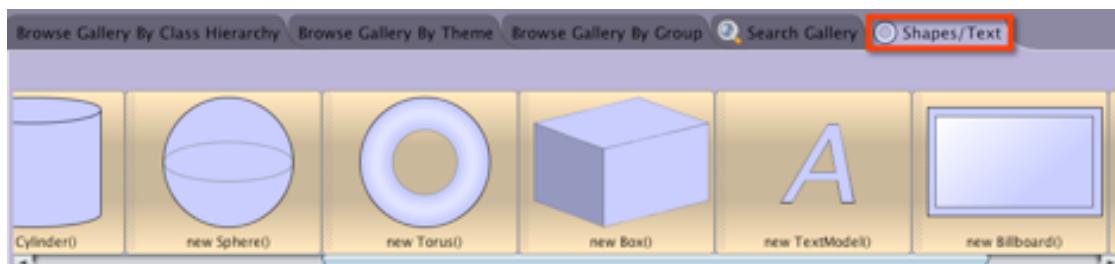


Figure 2.a.10 Shapes / Text in the Alice Gallery

# Setting up a Scene



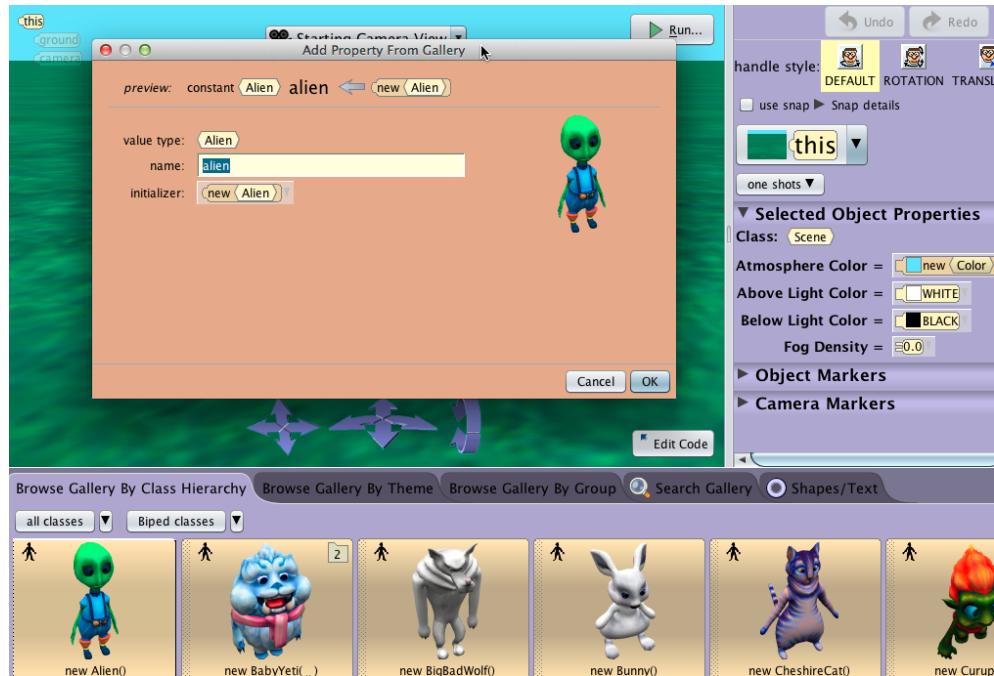
## ADDING AN OBJECT TO A SCENE

[Video: Adding Objects to a Scene](#)

The purpose of this section is to illustrate how to create an object and add it to a scene. The illustrations will include typical objects from the Browse by Class Hierarchy tab in the Gallery. In addition, we will illustrate how to add an object that is a Sims 2 person as well as objects from the Shapes/Text tab. Other tabs in the Gallery work in the same way as those illustrated here.

### *Add an object to a scene*

An object can be added to a scene in either one of two ways. One way is to single-click the thumbnail sketch of the desired object in the gallery. A dialog box is displayed where a name for the object can be entered (or a default name can be accepted), as shown in Figure 2.b.1. The name should be all one word (no spaces) and should begin with a lowercase letter of the alphabet. To use two or more words, use camelCase which starts with a lowercase letter for the first word and then uses a capital letter for each additional word. For example, the alien might be named greenAlien. Click OK when done.



*Figure 2.b.1 Enter a name for the new object*

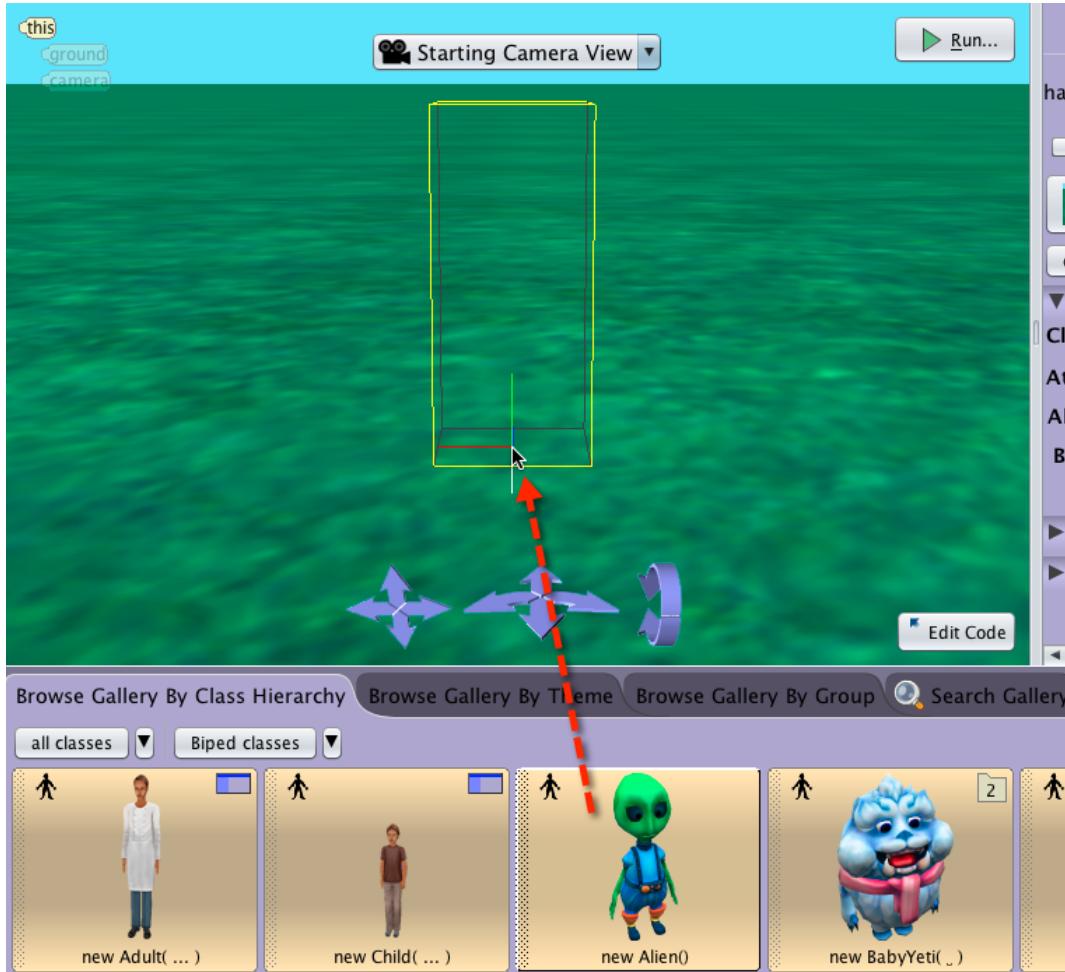
When OK is clicked, Alice displays the object in a default location in the scene and automatically adds the name of the object to the Object tree in the Scene Editor, as shown in Figure 2.b.2.



*Figure 2.b.2 A new object's name is automatically added to the Object tree*

The second way to add an object is to click and hold the left mouse button on the thumbnail sketch and drag it into the scene. The display of the mouse cursor will change to a box-like

outline, as shown in Figure 2.b.3. This is a bounding box that shows where the object will be located when the mouse button is released. This method of adding an object to a scene allows the user to control where the object will be positioned in a scene.



*Figure 2.b.3 Click and drag thumbnail sketch into the scene*

When the mouse cursor is released, a dialog box pops up where a name for the object can be entered (or accept the default name) in exactly the same way as described above.

### *Multiple objects*

It is possible to add more than one object of the same class. It is also possible to construct different objects from different classes in the same scene. Figure 2.b.4 shows four different objects in an Alice scene, each constructed from a different class.



*Figure 2.b.4 Objects of different classes in an Alice scene*

### *Add a Sims 2 person*

The first sketch (on the left) in the Biped collection is a Person Builder for the Person class of Sims 2 people. To open the Person Builder, single-click the Person thumbnail sketch, as shown in Figure 2.b.5.



*Figure 2.b.5 The Person class defines Sims 2 people objects*

As shown in Figure 2.b.6, the Person Builder provides options for: life stage (child, adult), gender (male, female), skin tone (lighter, light, dark, darker), body (dozens of costumes), an in shape ... out of shape slider, and head (hair color and style).



Figure 2.b.6 Sims 2 person-builder

Select features for each option and then click OK. A naming dialog box will pop up, as shown in Figure 2.b.7. Note that the features selected in the people-builder options are listed as “resources” to be used in constructing the person object. In this context, a resource is a painted image that is used to create the object’s appearance. In the naming dialog box, enter a name for the person and click OK. The person object will be added to the scene.

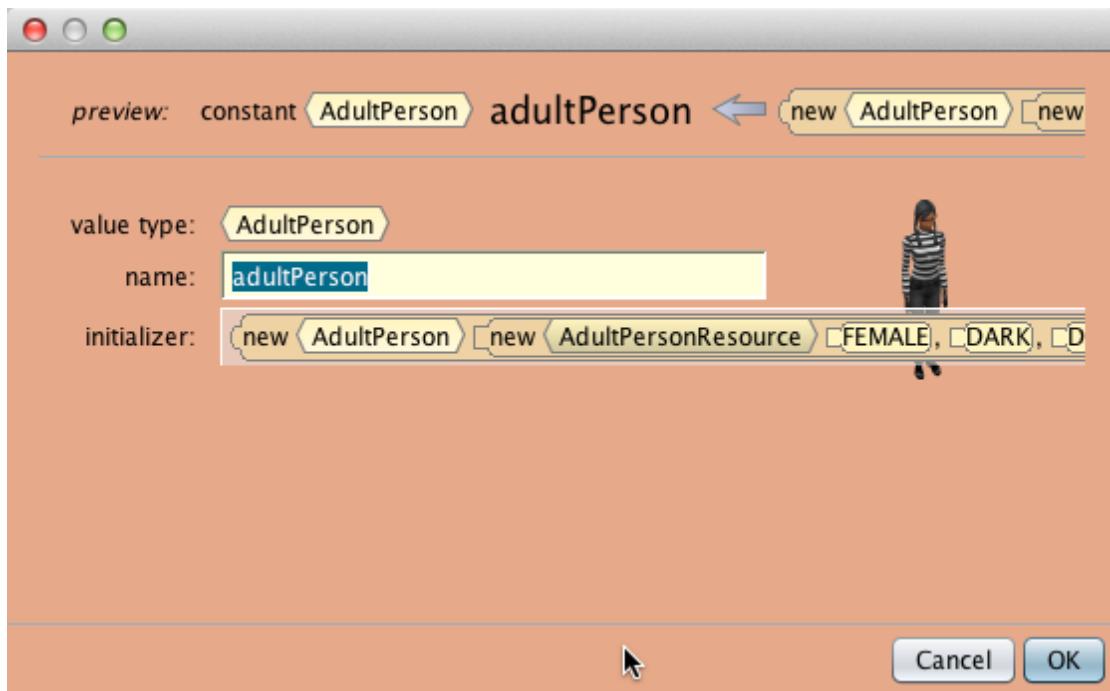


Figure 2.b.7 Naming a Sims 2 person

### Add a Geometric Shape object

In addition to the models in various collections, the Gallery also has a few basic, geometric shapes (disc, cone, cylinder, and sphere). To create a geometric shape, click one of the thumbnail sketches, as shown in Figure 2.b.8.

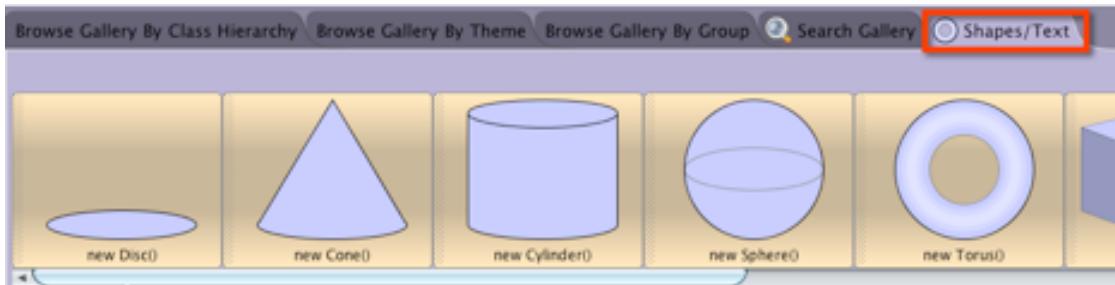
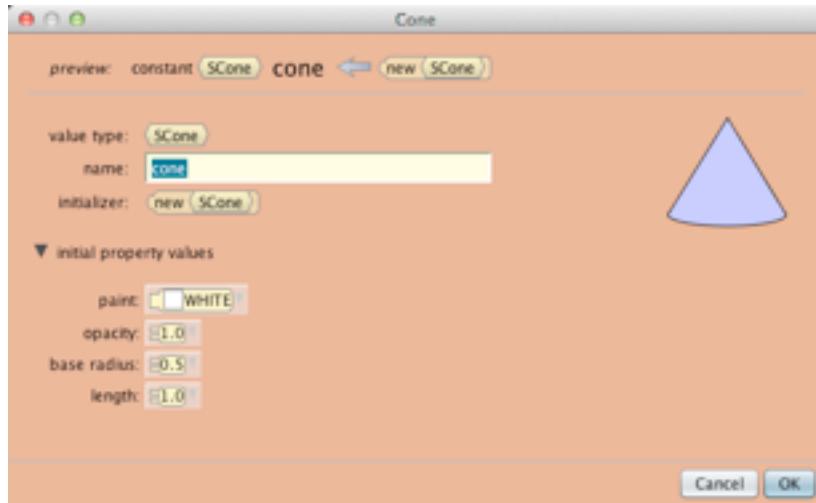


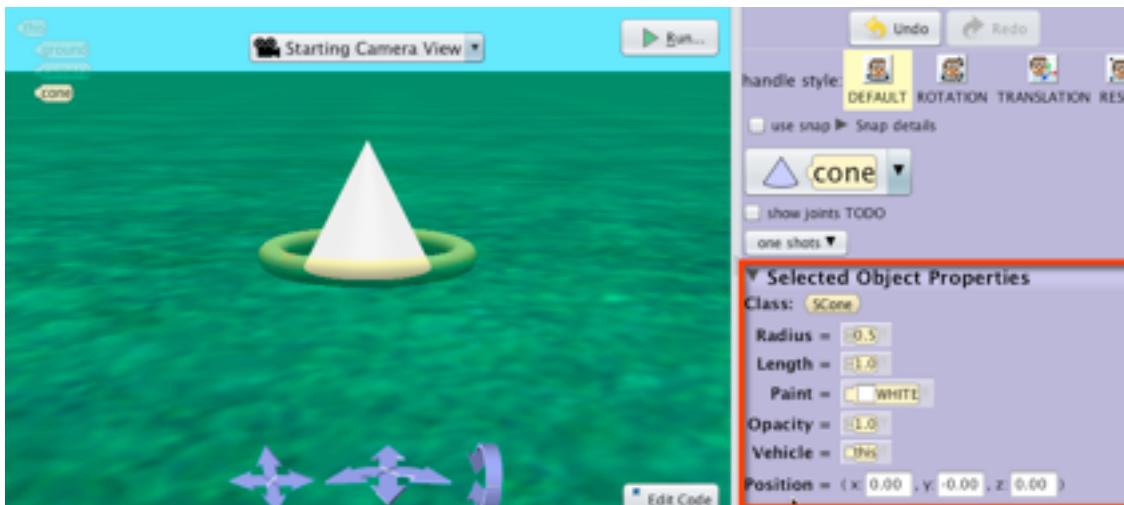
Figure 2.b.8 Geometric shapes in the Gallery

When a shape model is selected, a dialog box is displayed for entering a name for the new object, as illustrated in Figure 2.b.9.



*Figure 2.b.9 Naming a geometric shape object*

The new object can be positioned in the scene and properties can be set, as shown in Figure 2.b.10. Details for setting the properties of an object are provided in the next section of this guide.



*Figure 2.b.10 Properties can be set for painting, resizing, and other modifications*

### Add 3D text

To create an instance of the TextModel class, click the TextModel thumbnail sketch in the Gallery, as shown in Figure 2.b.11. 3D text is useful for displaying screen credits, a timer, or a scoreboard for a story or game.

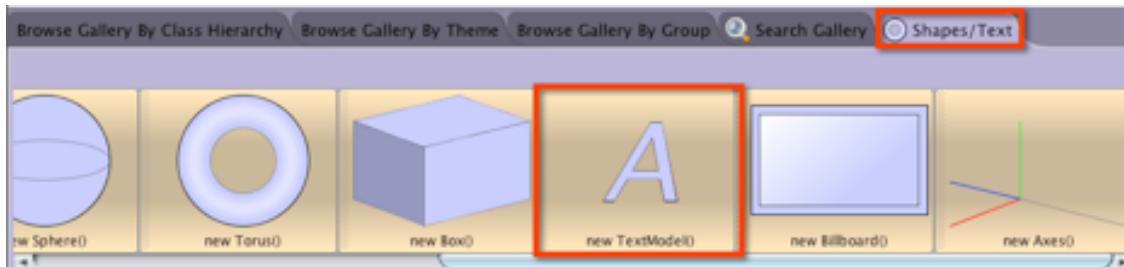


Figure 2.b.11 TextModel button to create 3D text object

When the TextModel thumbnail sketch is clicked, a dialog box is displayed where two items of information must be entered, as shown in Figure 2.b.12. The first item is a name for the 3D text object. The second item is a string of text characters that will be displayed by the text object. The drop down menu for a string of text characters (a STextModel object) allows a string to be empty (""), the default value "hello", or a Custom TextString entered using the keyboard.

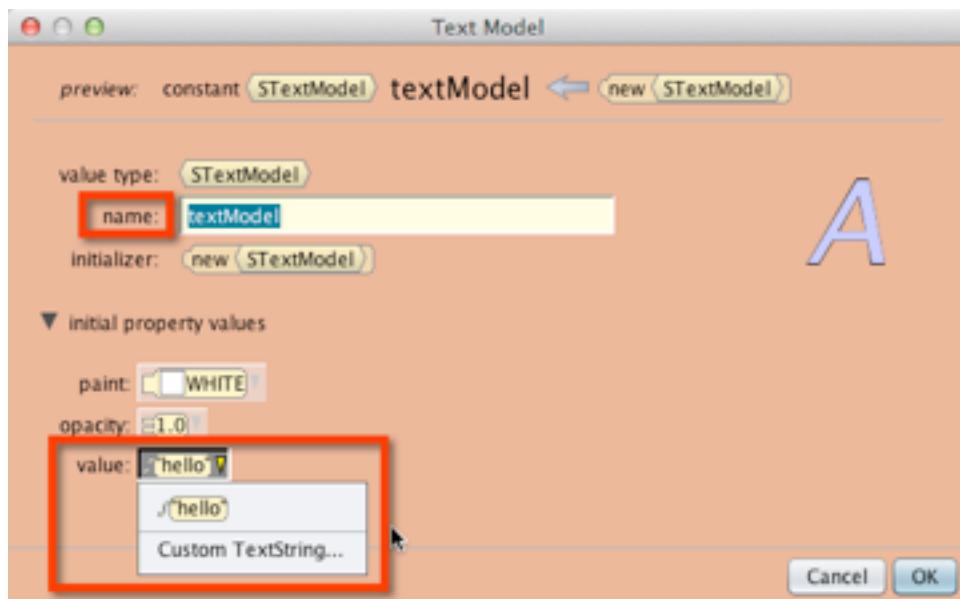


Figure 2.b.12 Text Model dialog box

An example is shown in Figure 2.b.13, where we entered Scoreboard as the name and selected Custom TextString. Then, in the pop-up Custom TextString box, we entered "0" as the value of the text to be displayed. Just to be clear, note that the name of the object is Scoreboard and the text string it displays is "0". That is to say, the name of a text object and the text string it displays are not necessarily the same.

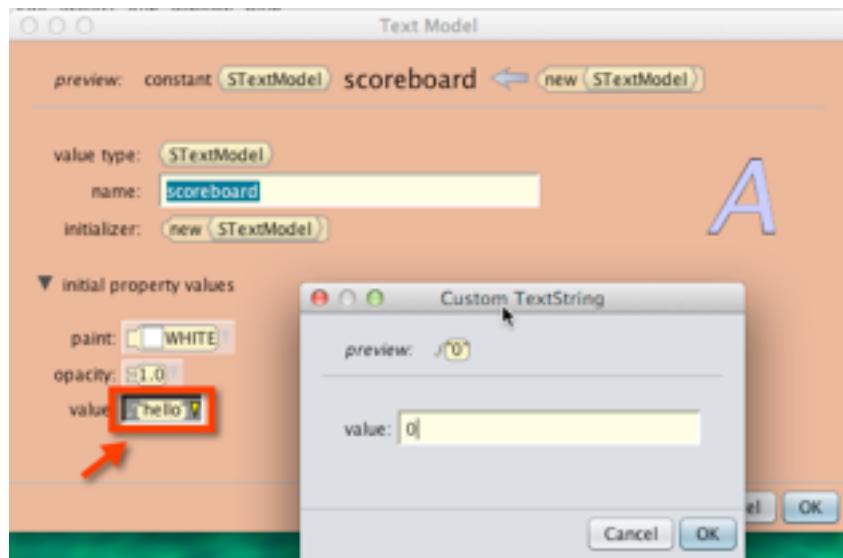


Figure 2.b.13 Naming a text object and initializing the text string (alphanumeric)

The new 3D text object will be displayed in the scene, as illustrated in Figure 2.b.14. The text object can be positioned in the scene, and its properties can be set in Setup.

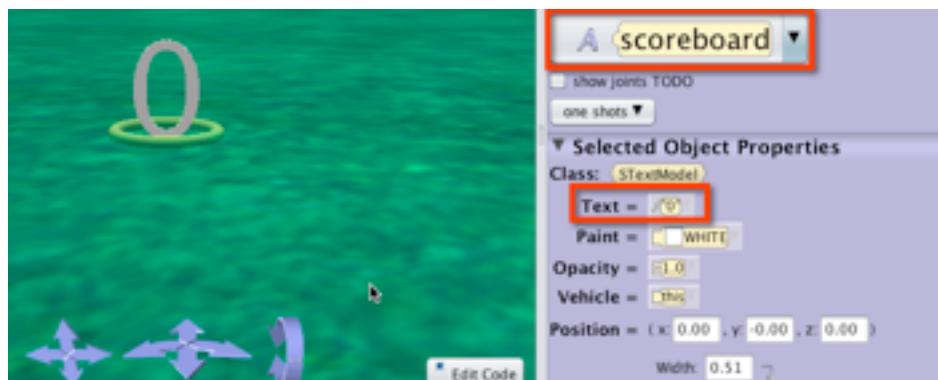


Figure 2.b.14 A 3D text object

### Add a billboard (2D image resource)

A 2D image may be added to a scene as a billboard. Billboards are useful as backdrops, a narrative element in a story, and for presenting instructions on how to play a game. To create a billboard from a 2D image, click the Billboard thumbnail sketch, as shown in Figure 2.b.15.

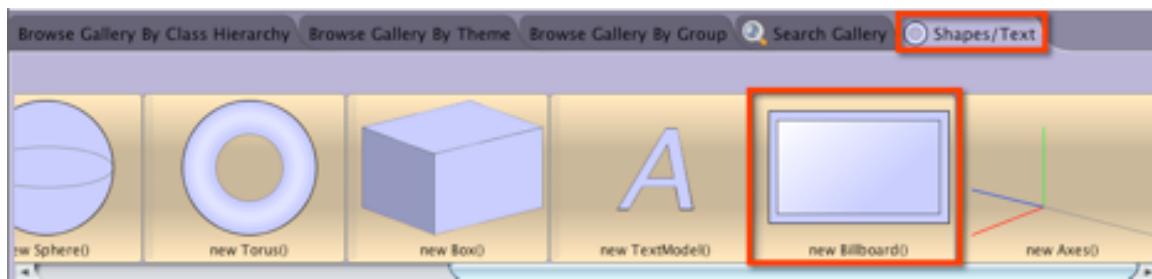


Figure 2.b.15 Billboard thumbnail sketch in the Gallery panel

When the Billboard is selected, a dialog box is displayed, as shown in Figure 2.b.16, where three items of information must be entered.

- The first entry item is a name for the object.
- The second item is a drop-down menu for finding and importing a 2D image or to create a billboard of a solid color. The file format of a 2D image must be .jpg, .png, .bmp, or .gif (must be all lower-case).
- The third item is a drop-down menu to select an image or color for the back of the billboard. (As a 2D object, a billboard has two sides: front and back.)

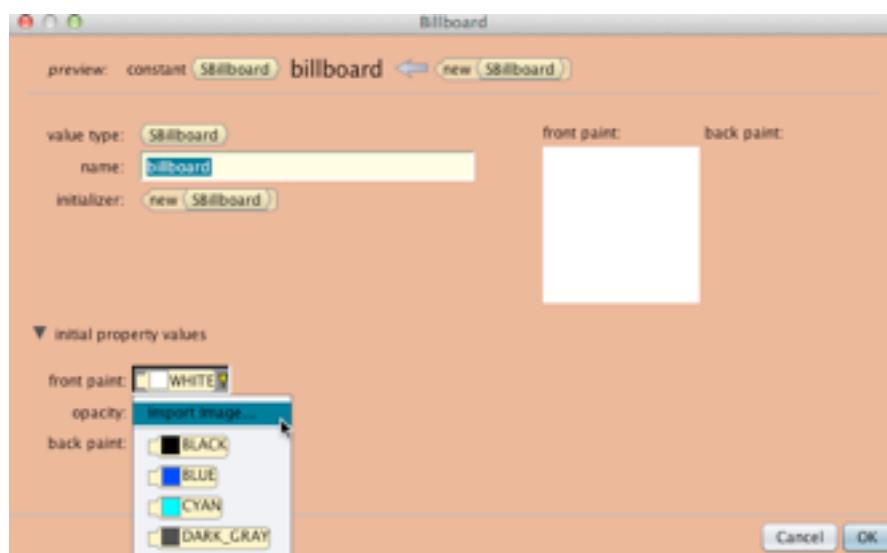


Figure 2.b.16 Billboard dialog box requires a name and an image or color for front and back

An example is shown in Figure 2.b.17, where we accepted the default name "billboard" as the name for the new billboard object, selected an image source (an Alice Team photo) and then, in the pull-down menu for the back of the billboard, selected a solid black color.

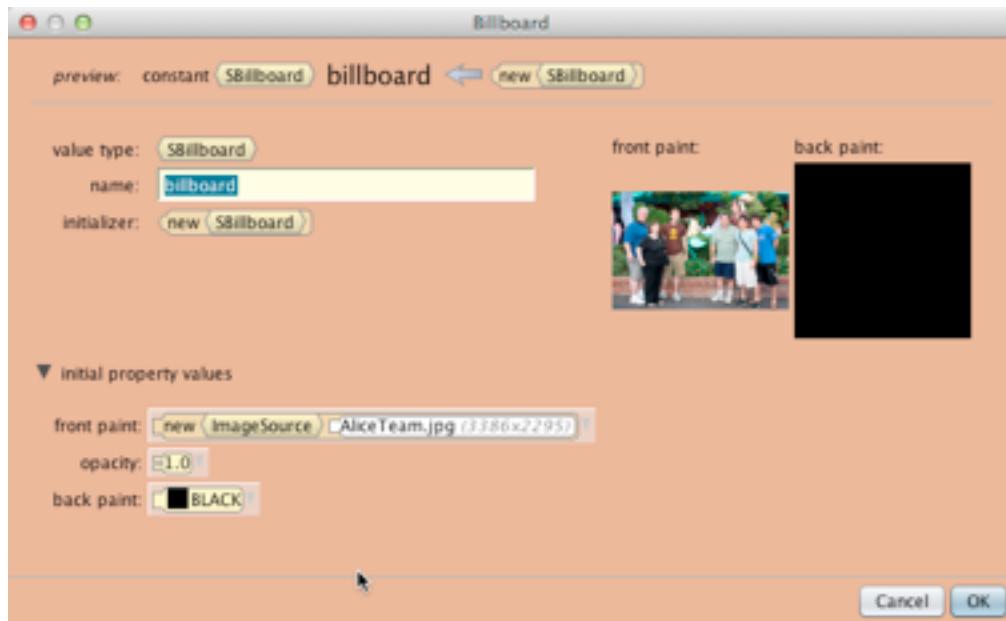


Figure 2.b.17. Example Billboard entries

When OK is clicked, the object is displayed in the scene, as shown in Figure 2.b.18. The image can be positioned in the scene and its properties can be set in Setup.



Figure 2.b.18 Alice team photo as a Billboard object in a scene

Once a 2D image is added to a scene as a billboard, the image will show up in the list of resources found in the Project menu under Manage Resources, as shown in Figure 2.b.19.

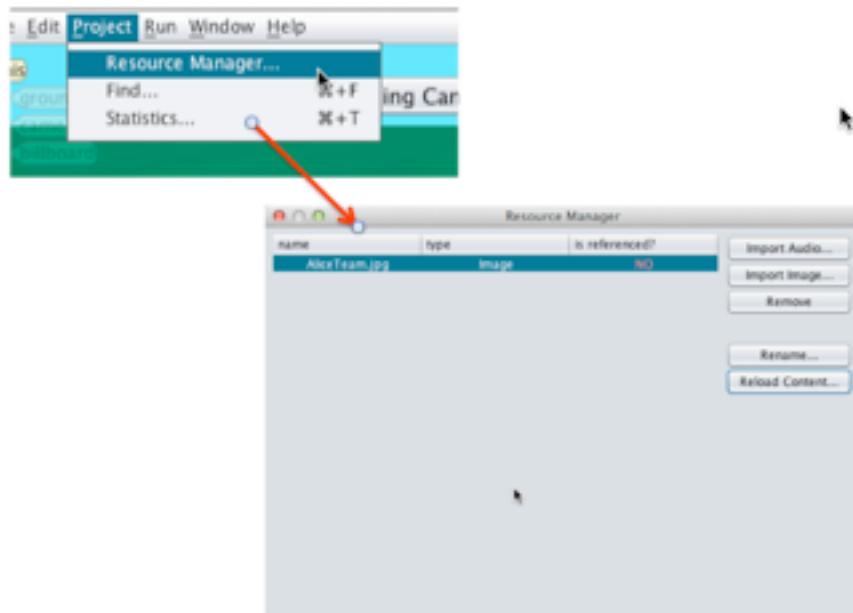


Figure 2.b.19 Billboard image is listed in Manage Resources

### Add an Axes object

To create an Axes object, click the Axes thumbnail sketch, as shown in Figure 2.b.20.

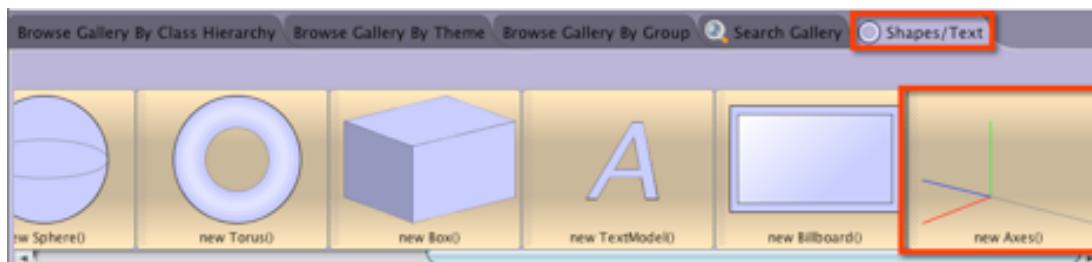


Figure 2.b.20 Axes button in the Gallery

When the Axes is selected, a dialog box is displayed, as shown in Figure 2.b.21, a name is entered for the object. Click OK. An Axes object will be added to the scene, as shown in Figure 2.b.22

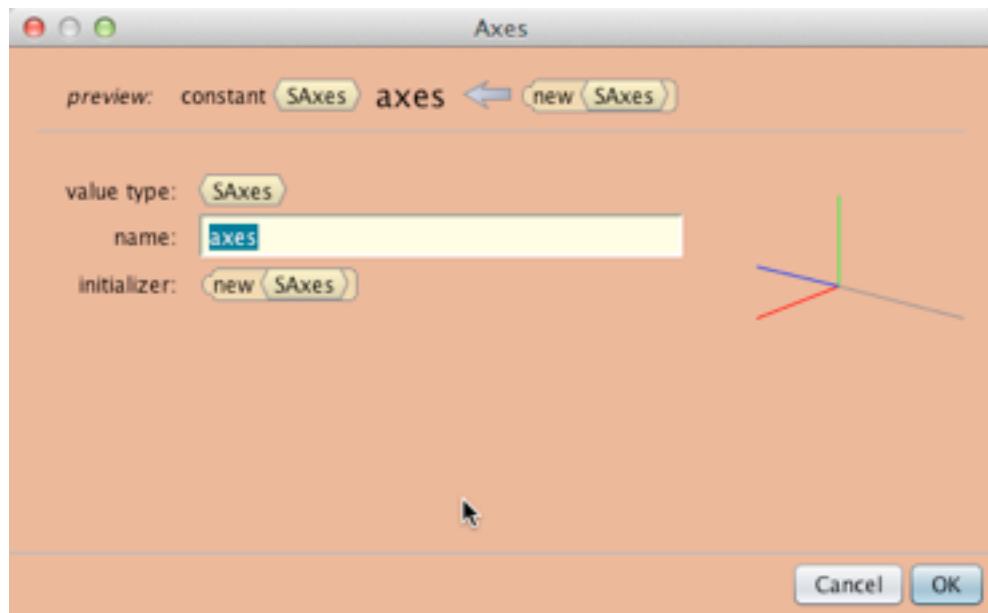


Figure 2.b.21 Naming a new Axes object

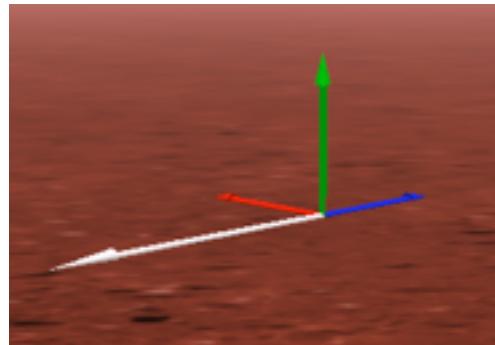


Figure 2.b.22 A new Axes object in a scene

### Using an axes for orientation

The orientation of an object and its skeletal joints is very important when working with 3D objects in a virtual world. One way to determine the orientation of an object (or a skeletal joint within an object) is to create an Axes object and align it to the orientation of the target object or a joint within the object.

To illustrate, we created a new Mars scene with an asteroid, as shown in Figure 2.b.23. Just looking at the asteroid, we have no way of knowing its forward direction. To determine the orientation of the asteroid, we selected the axes and used a moveAndOrientTo method to move

and orient the axes to the asteroid. Then, as shown on the right in Figure 2.b.23, the axes arrows are aligned with the orientation of the asteroid. The white arrow of the axes shows the forward direction for the asteroid, the red arrow shows the asteroid's right, and the green arrow shows the asteroid's up direction.

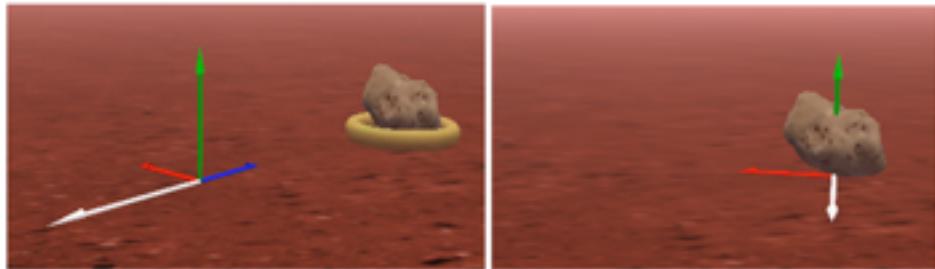


Figure 2.b.23 Original axes location (left) and aligned with android (right)

## HOW TO SET OBJECT PROPERTIES IN THE SCENE EDITOR

The purpose of this section is to demonstrate how to set a property of an object in the Scene editor. Properties are data that identify an object as an individual. A driver's license typically describes a person with a photo and identifying data such as first and last name, hair color, eye color, skin tone, height, and weight. In a similar way, identifying data about an Alice object include its Name, Paint, Opacity, Vehicle, Position, and Size (composed of Width, Height, and Depth) properties.

### Setup

To view property data about an object, click on the object in the scene. When clicked, the selected object will be surrounded by a ring-shaped handle, as shown in Figure 2.c.1. (The ring handle is a mouse control that can be used to turn the object left and right.) The selected object's properties are displayed in the Setup just to the right of the scene. In this example, the selected object is named seaweed2 which is an object of the Seaweed2 class.



*Figure 2.c.1 Selected object and its properties*

### *Set a property*

The phrase set a property means to modify a property. Setting a property may change its appearance, size, or position in the scene.

### *Set paint*

Paint includes both the texture map and the color of an object. An object has a wire mesh of polygons that creates the external appearance of the object. A texture map is applied to the mesh surface to create a "skin" coating that encloses the object. For people objects, the "skin" includes hair and eyes and for animal objects the color of fur, eyes, nose, ears, paws, and tail (if appropriate). An object's color is a coating that covers the texture map. By default, a WHITE color coating is simply a clear coating that does not change the colors on the skin (somewhat like a clear sugar-glaze on a doughnut).

To change the color coating, click the Paint's pull-down menu in Setup and select a color in the menu. In Figure 2.c.2, MAGENTA has been selected and the seaweed is painted with this color.



*Figure 2.c.2 Setting the color for painting an object*

### *Set opacity*

One way to think about opacity is that it is the opposite of transparency. By default in Alice, the opacity of an object is 1.0, which means the object is totally opaque (looks solid). Setting the opacity to 0.0 would mean that the object is totally transparent (can see right through it). The range of values for opacity, therefore, is from a low of 0.0 to a high of 1.0.

To set the opacity of an object, select the object in Setup and then click on the Opacity button. A drop-down menu allows the selection of opacity in a scale of 0.0 to 1.0, as shown in Figure 2.c.3. In this example, the seaweed2 object was selected and the opacity was set to 0.4. As can be seen by comparing the seaweed2 object with the other seaweed object beside it, the seaweed2 object has faded and is now partially transparent.

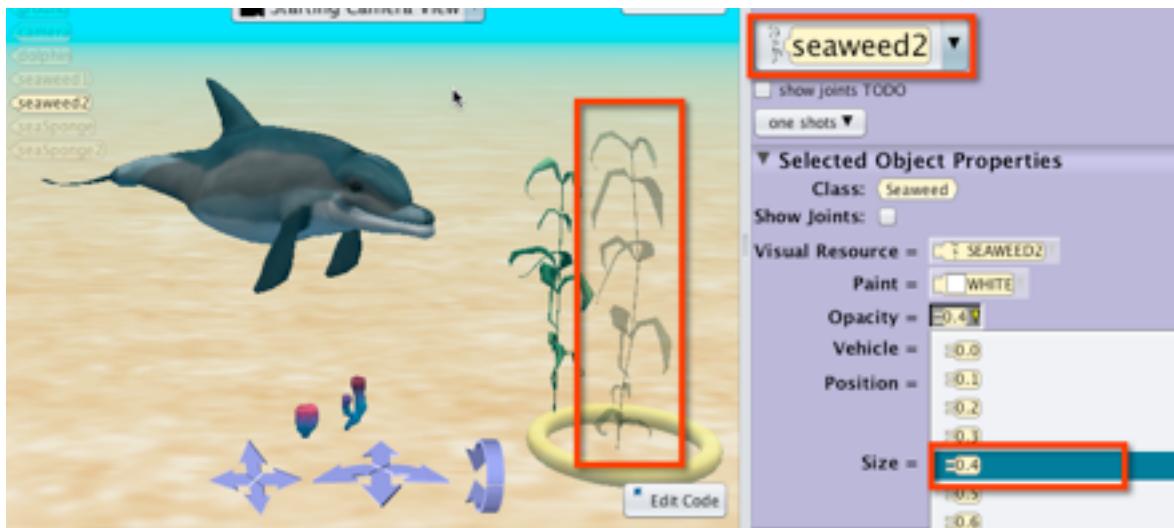


Figure 2.c.3 Setting opacity

### *Set vehicle*

In Alice, a vehicle is an object whose motions affect the motions of another object in the virtual world. As an analogy, consider a car as a vehicle. When a person is riding in a car and the car moves forward, the person moves forward with the car. In Alice, the current scene is, by default, the vehicle for all objects within it. So, if the scene moves left all objects within the scene would move left with it.

To set the vehicle of an object, first select the object for which the vehicle is to be changed. Then click on the Vehicle button. A drop-down menu allows selection of another object to be the vehicle, as shown in Figure 2.c.4. In this example, we set the vehicle of the seaweed to be the dolphin. Now, if the dolphin moves the seaweed will move with it, in the same direction and distance or if the dolphin turns, the seaweed will turn with it in the same angle of rotation.

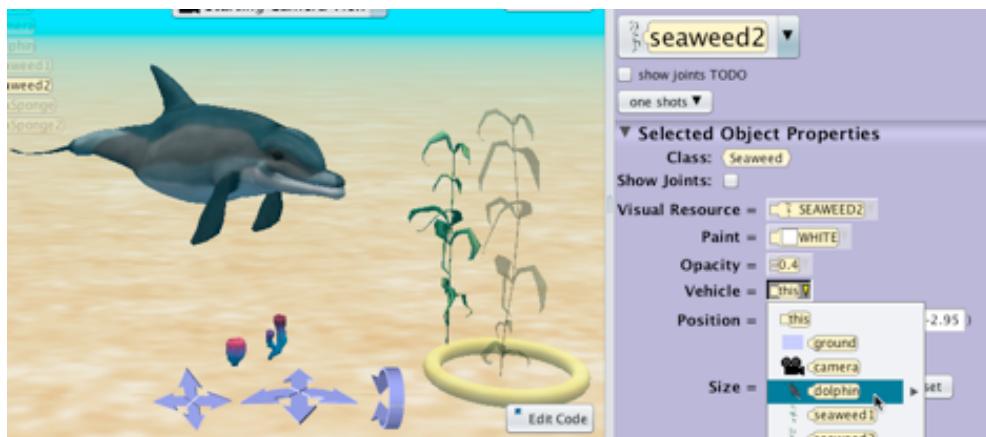


Figure 2.c.4 Set the vehicle of seaweedBack to be the dolphin

## *Set position*

The position of an object in a scene is relative to the center point of the scene. Using the mouse to drag the object around in the scene most common method of setting the position of an object in a scene. However, there may be some worlds in which it is important to position an object in an exact location in the scene.

The Setup panel of the Scene editor allows precise positioning of an object by setting its position coordinates. To set the position, click the mouse in one of three coordinate boxes and use the keyboard to enter a numeric value. As an example, in Figure 2.c.5 we added a blue cone, positioned at the center point of the scene (0,0,0). Then, the dolphin was positioned by entering numbers in the position boxes for x (-1.36), y (-0.04), and z (1.35). After the new values were entered and the Enter key was pressed the dolphin was immediately repositioned at that location in the scene.

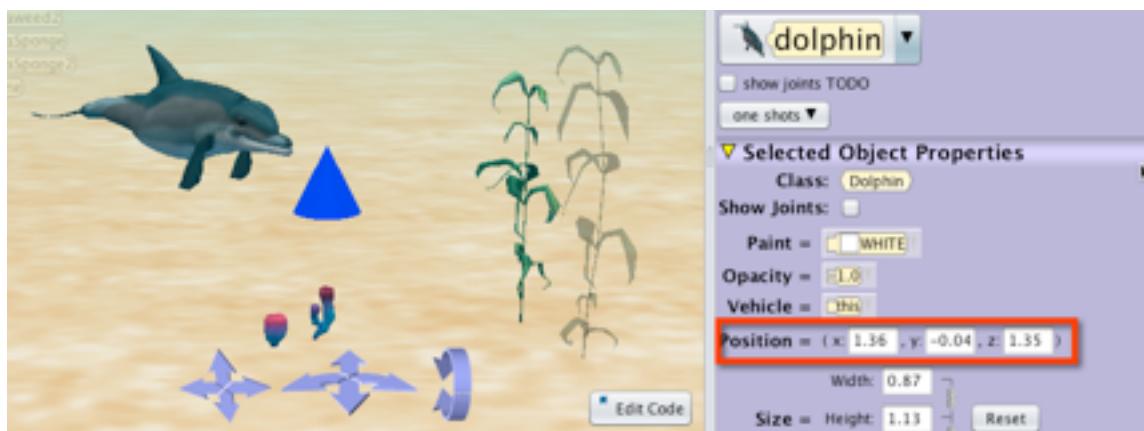


Figure 2.c.5 Setting the position coordinates of an object

## *Set size*

An object's size has three dimensions: width, height, and depth. The size of an object in a 3D world often needs some adjustment when added to a scene where other objects are much larger or smaller. For example, in Figure 2.c.6 the sea sponge's width is 0.12, height 0.24, and depth 0.10 m. Relative to the seaweed, the sea plant looks too small. To change the size, click the mouse in one of three dimension boxes and use the keyboard to enter a numeric value. By default, a change in size is locked to be proportionate in all dimensions.

To return an object to its original size, click the Reset button, immediately to the right of the Size dimension boxes.



Figure 2.c.6 Setting the size dimensions of an object

## HOW TO SET SPECIAL EFFECTS IN A SCENE

The purpose of this section is to illustrate special effects for creating a mood by using the scene's special effects properties: atmosphere color, lighting, and fog.

### *Example*

To illustrate atmosphere, light and fog special effects, we created a world with a brown ogre (of the Ogre class) in a green grass, blue sky scene, as shown in Figure 2.d.1. The rock and hedge objects are from the Props collection in the Gallery.



Figure 2.d.1 Example scene to illustrate special effects

We selected this scene in the Object tree, as shown in Figure 2.d.2.

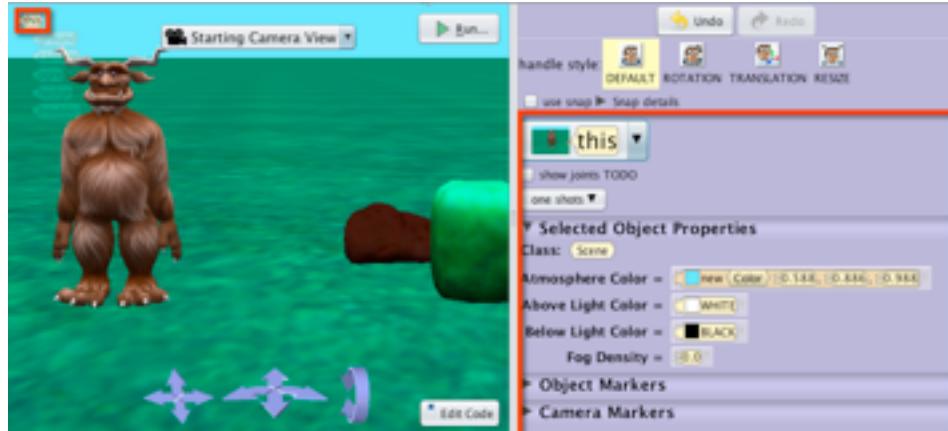


Figure 2.d.2 Select this scene in the Object tree

The default settings of Atmosphere, Above Light Color, Below Light Color, and Fog Density are as shown in the Setup, as shown in Figure 2.d.3.

**NOTE:** Examples of resetting these values are shown below in a progressive manner below. That is, each change carries over to the next so that the final result is cumulative.

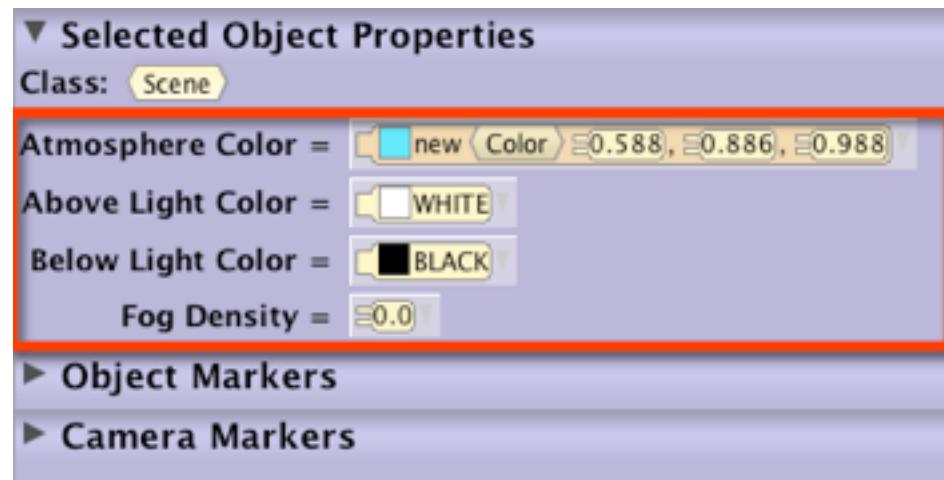


Figure 2.d.3 Default settings for lighting and fog in this scene

### *Set atmosphere color*

The Atmosphere Color determines the color of the sky. To set the sky to a different color, click the button for Atmosphere Color and select a different color from the drop-down menu, as shown in Figure 2.d.4. We selected dark blue, and the result is shown in the screen capture at the left.



Figure 2.d.4 Setting the color of the sky (atmosphere)

The pull-down menu for color provides a Custom Color option. Selecting Custom Color will pop up a dialog box to select from a grid of color swatches or use HSB or RGB color codes, as shown in Figure 2.d.5.

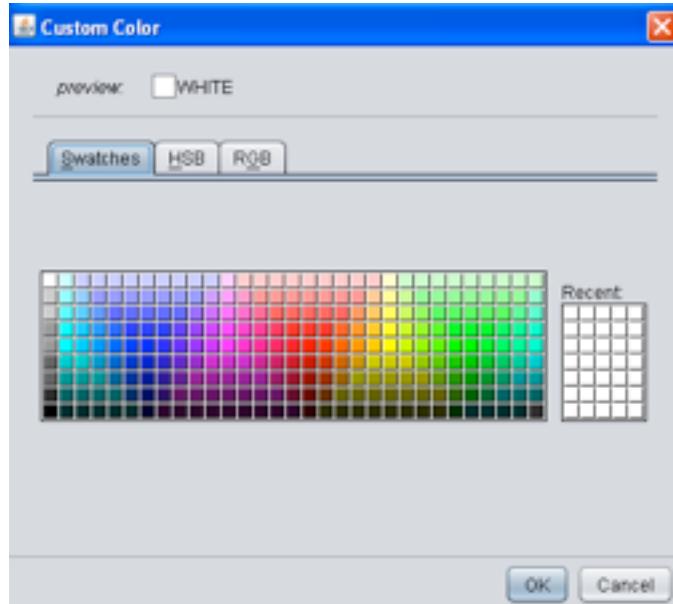


Figure 2.d.5 Custom color options

### *Set Above Light Color*

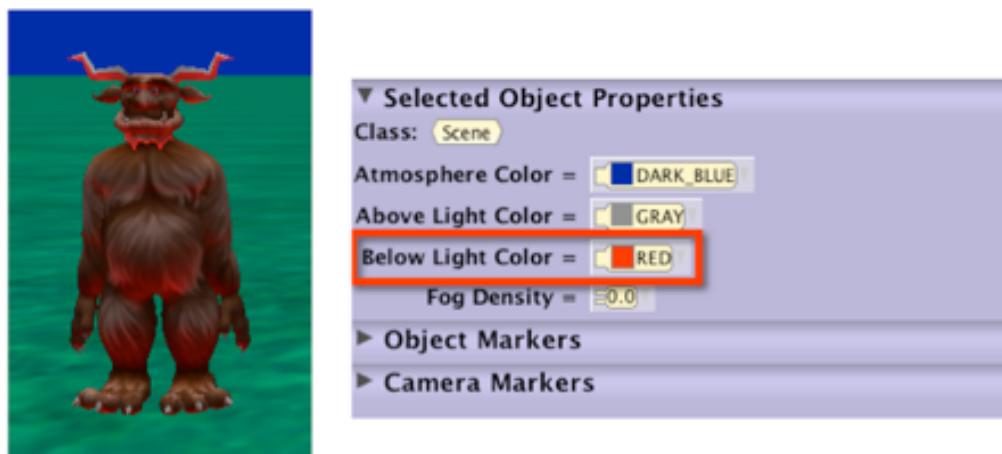
The lighting in a scene is projected from above the scene. By default, the lighting above the scene has a setting of WHITE, which looks like a clear day when the sun is shining. To change the lighting to other settings, click the Above Light Color button and select a different color, as shown in Figure 2.d.6. In this example, we selected GRAY and the result (similar to a cloudy day) is shown in the screen capture at the left.



*Figure 2.d.6 Above Light Color is set to GRAY*

### *Set Below Light Color*

In addition to light being projected from above the scene, it is also possible to project light from below. By default, the Below Light Color is set to BLACK, which is the equivalent of no lighting from below. To turn on lighting from below, click the Below Light Color button and select a different color, as shown in Figure 2.d.7. In this example, we selected RED for a fiery effect and the result is shown in the screen capture at the left.



*Figure 2.d.7 Below Light Color set to RED*

## *Set Fog Density*

Fog is used to create a misty effect in the scene, as shown in Figure 2.d.8. Fog can be used to allow objects to move into the scene from the back, gradually becoming more and more visible. By default, the fog density is set at 0.0, meaning that there is no fog. A fog density setting of 1 is the most fog that is possible for the scene (only the atmosphere is visible). To set the fog, click the Fog Density button and select a density value from the pull-down menu of values (in the range of 0.0 – 1.0). In this example, we selected 0.3 (approximately 30%) for a mild fog effect and the result is shown at the left in Figure 2.d.8.

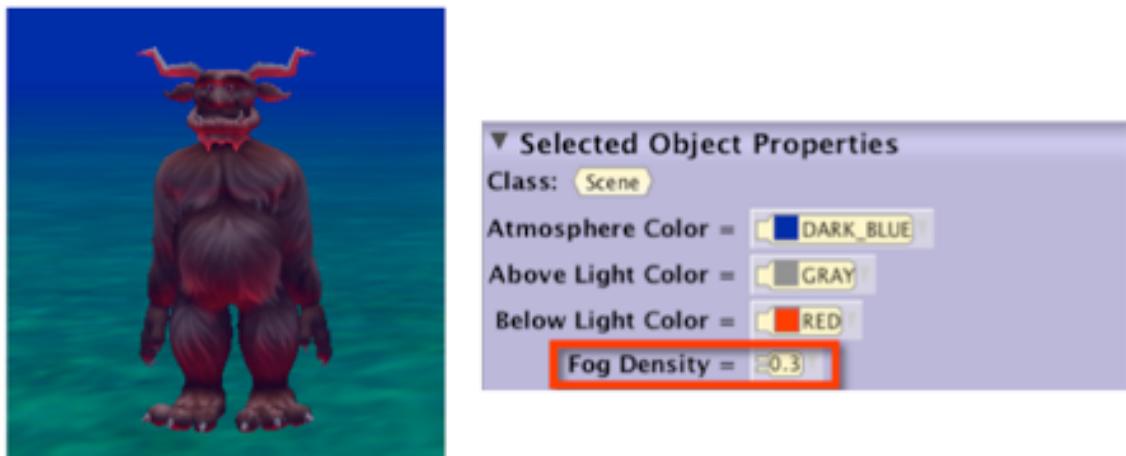


Figure 2.d.8 Fog Density set at 0.3

## MARKING AND CHANGING THE CAMERA'S POSITION

The primary purpose of this section is to illustrate how to change the camera's position. Two features of Alice 3 support changing the camera's position: (1) camera navigation controls, and (2) pre-set camera viewpoints. A secondary purpose of this section is to introduce markers as a way to create known camera positions in the scene.

### *Marking the camera's position*

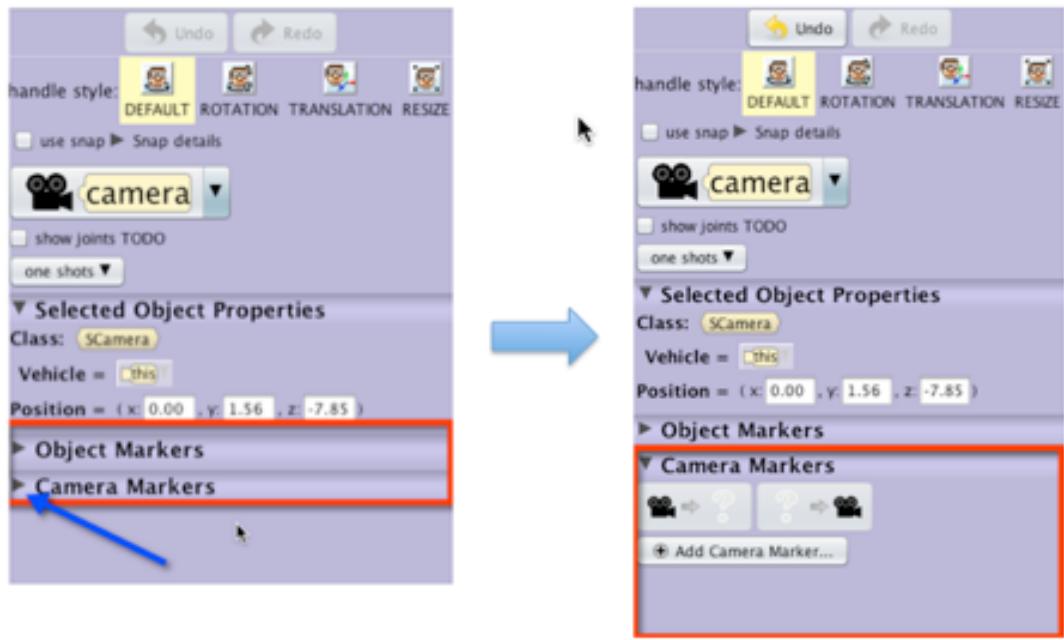
Alice has only one camera in a scene. The camera is moved around and repositioned for close-up views and special effects. Because moving the camera is a common action when setting up a scene in the Scene editor, Alice provides a way to mark a camera position so as to create known camera positions in the scene.

This is similar to Hollywood studios where a single camera film-style production is often used. Each scene and camera angle is setup and rehearsed until the director is happy with the arrangements. The camera positions (location and orientation angle) are marked before any actual filming begins.

In Alice, a camera marker is an object that remembers the position and orientation of the camera at the time the marker was created. The camera can then be moved or rotated to a different location and orientation, but the marker stays where it was created. When creating a camera marker in the Scene editor, the marker object is visible in the scene. There is no need to worry about camera markers cluttering up the scene, however. Later, when the user clicks the Run button to play an animation, markers will not be visible in the scene. (The markers are still there, just not visible.)

### *Markers section of the Setup panel*

The Camera Markers section of the Setup in the Scene editor is used to create markers for the camera's location and orientation. To view the Camera Markers section, click on the arrow next to the Camera Markers label at the bottom of the panel, as shown in Figure 3.a.1. The Camera Markers section should expand to show buttons for creating camera markers. Notice that the Camera Markers section has three buttons -- two small buttons having a dark gray camera icon and a question symbol and one button labeled Add Camera Marker ....



*Figure 3.a.1 Collapsed (left) and Expanded (right) Camera Marker section in Setup*

### *Creating a camera marker at the starting position*

We recommend marking the starting location of the camera before moving the camera around in the scene. The camera can then be moved freely around the scene and can always be returned to its original position, using the marker.

To create a starting location camera marker click on the Add Camera Marker... button, as shown in Figure 3.a.2. A dialog box will pop up, as shown in Figure 3.a.3. Enter a meaningful name for the marker, for example startPosition. When a name is entered, press the Enter key and Alice will automatically create a camera marker object at the current location of the camera. The marker remembers not only the location but also the camera's orientation (the direction and angle at which it is pointed). This information is commonly known as the camera's viewpoint.

If more than one camera marker is created, Alice will create the markers using a different color (red, green, blue, etc.) for each. As shown in Figure 3.a.4, we created two camera markers. One is red (startPosition) and the other is green (overheadPosition). In addition, the name of each camera marker is displayed in a matching color in the Markers section of the Setup panel.

Note: We pulled the camera way back in this scene in order to obtain a view of both markers for the screen capture in Figure 3.a.4. Otherwise, one of the markers would have been behind the camera.



Figure 3.a.2 Click Add Camera Marker ...

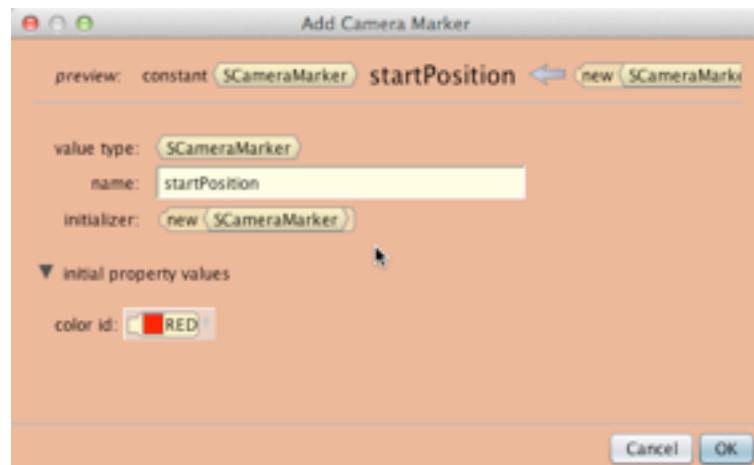


Figure 3.a.3 Enter a meaningful name for the camera marker

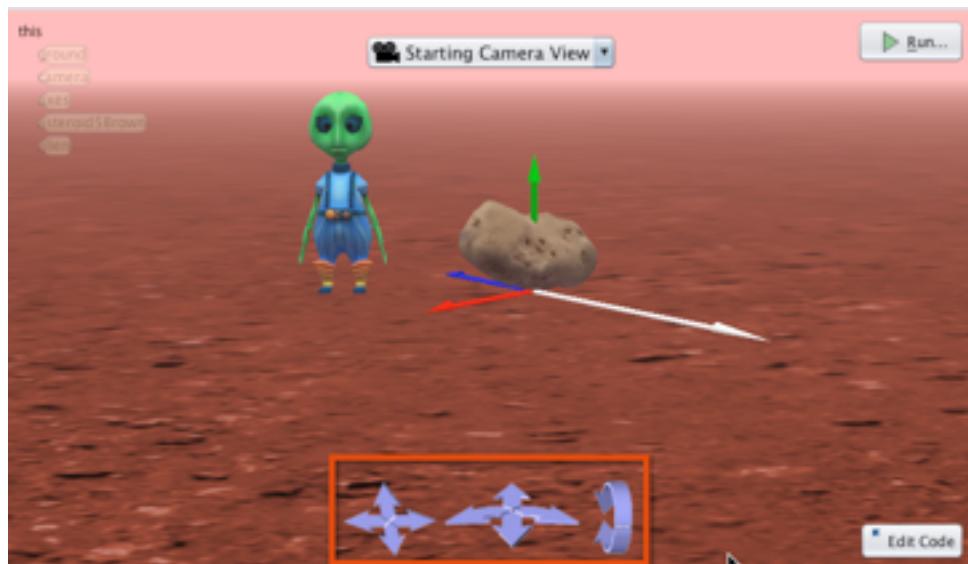


*Figure 3.a.4 Different markers have different colors*

### *Change the camera's position using navigation controls*

At the bottom edge of the scene view are three sets of camera navigation controls, as shown in Figure 3.a.5. The most common use of navigation controls is to set the camera's initial point of view for best effect in animation.

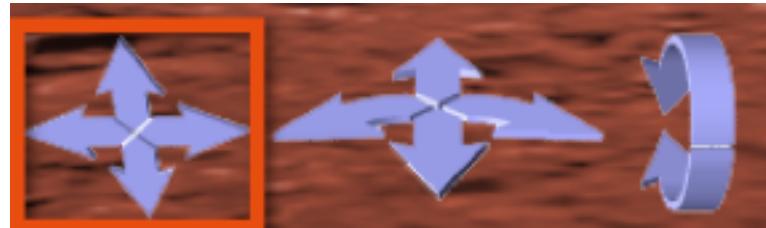
Clicking the arrow will manipulate the camera as described. Clicking and dragging in the direction of the arrow will speed up the camera movement, and clicking and dragging in between two arrows will combine the camera manipulation.



*Figure 3.a.5 Three sets of camera navigation controls*

### *Move camera up/down/left/right*

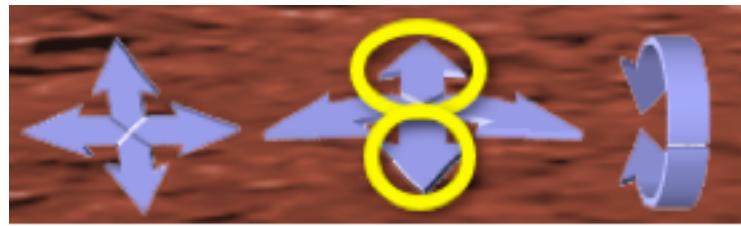
The set of four arrows on the left, as shown in Figure 3.a.6, move the camera up or down (vertically), and left or right (horizontally), from the camera's point of view. As with any move action in Alice, these buttons change the location of the camera in the scene, but not its orientation (the direction the camera is facing). Professional videographers refer to these actions as the camera being ped (up and down) or tracked (side to side).



*Figure 3.a.6 Move the camera up, down, left, or right*

### *Move camera forward/backward*

For purpose of clarity, the set of four arrows in the center are described here in two subsets. The two arrows pointing forward and backward (horizontally), as outlined in yellow in Figure 3.a.7, move the camera forward or backward (as seen by the camera). We refer to this action as the camera is zooming in or out relative to an object in a scene. (Note, however, that professional videographers often use the term zoom for changing the focal length of the camera's lens to give the illusion of moving the camera).



*Figure 3.a.7 Move the camera forward or backward*

### *Turn camera left/right*

The other two arrows in the center set, as outlined in yellow in Figure 3.a.8, turn the camera to the left or right, as seen by the camera. As with any turn action in Alice, a turn changes the orientation of the camera in the scene, but not the location of the camera. Professional videographers refer to this action as panning the scene.

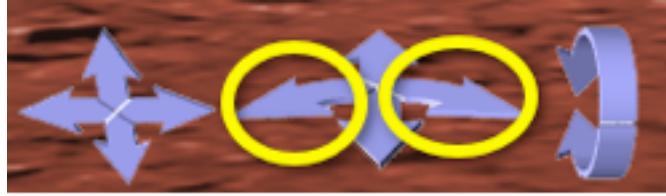


Figure 3.a.8 Turn the camera left or right

### Turn the camera forward/backward

The set of arrows on the right, as outlined in red in Figure 3.a.9, turn the camera forward or backward (a tilting action) in the scene. As with any turn action in Alice, a turn changes the orientation of the camera in the scene, but not the location of the camera. Professional videographers refer to this action as tilting.

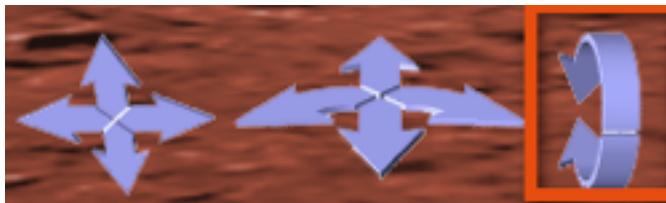


Figure 3.a.9 Turn the camera forward or backward

### How to position the camera at a marker

Let's assume that we have used the camera navigation controls to move the camera around the scene and it is no longer at the starting position. Now, we can take advantage of the camera markers we created earlier.

First, select (from the list of camera markers) a marker to which the camera will be moved. In Figure 3.a.10, the overheadPosition marker has been selected. Notice that the question symbols in the two camera marker buttons have been replaced with green camera icons because the overheadPosition camera marker in this example is green.

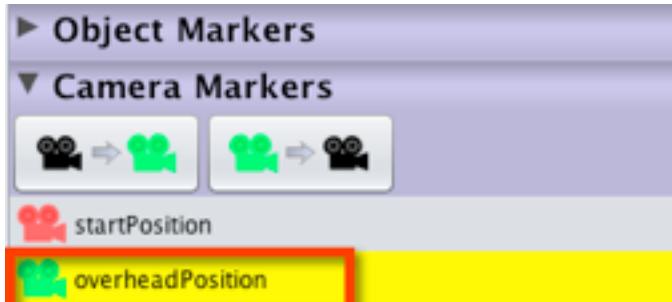


Figure 3.a.10 Step 1: Select the targeted marker

Secondly, click the camera => marker button (left of the two small buttons), as shown in Figure 3.a.11 to move the camera to the selected marker. The camera will immediately move and orient to the targeted marker.

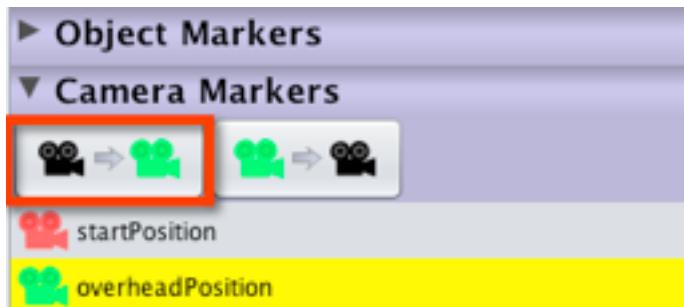


Figure 3.a.11 Step 2: Click camera => marker button (left)

### *Repositioning a camera marker*

Once in a while, a marker may have been created in the wrong place. Rather than deleting the marker and creating a new one, the existing marker can be repositioned. To reposition a marker, first position the camera in the desired new location and orientation. Then, select the marker to be repositioned in the list of camera markers. In the example shown in Figure 20.12, we selected startPosition (a red camera marker). Notice that the two small buttons now show the dark gray camera icon (current camera position) and a red camera icon (the selected marker).

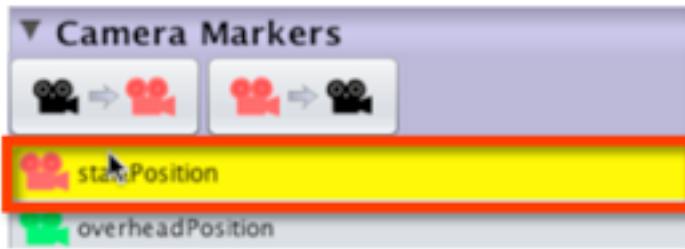


Figure 3.a.12 Select the marker to be repositioned

Now, click on the marker => camera button (outlined on the right in Figure 3.a.13). Alice repositions the selected marker to the current camera position.

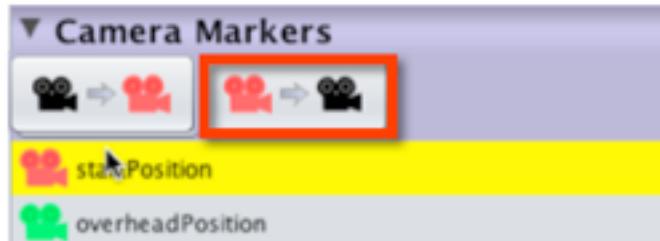


Figure 3.a.13 Click the marker => camera button (right)

## POSITIONING OBJECTS WITH MARKERS

In the previous section of this guide, we introduced camera markers. . The purpose of this section is to introduce object markers that remember the position and orientation of other kinds of objects.

### Markers

To better describe object markers, we use an analogy: a bookmark in a web browser (e.g., Firefox, Safari, IE, Chrome, or some other). To make it easy to find a favorite web site, a bookmark is created. Later, to return to that favorite web site, the bookmark in the browser is used to return to that website on the Internet. Object markers in Alice do a similar kind of thing. Markers are used to remember the position and orientation of an object at the time the marker was created. Then, later, after the object has moved or rotated to a different position, the object can be repositioned at the marker.

## Example

To illustrate object markers in this section, we have created a scene with the alien and an asteroid boulder in the Mars scene, as shown in Figure 3.b.1.



Figure 3.b.1 Example scene

## Open Markers in Setup

To open the Object Markers section of Setup in the Scene editor, click on the arrow next to the label Object Markers at the bottom of the panel, as illustrated in Figure 3.b.2 (left). The Object Markers section should expand to show buttons for creating object markers, as illustrated in Figure 3.b.2 (right).



Figure 3.b.2 Collapsed (left) and Expanded (right) Object Marker section in Setup

### Create an object marker

To create an object marker, first position the object in the desired location and orientation in the scene. Next, click on the Add Object Marker ... button. In the example shown in Figure 3.b.3, the object is the alien.

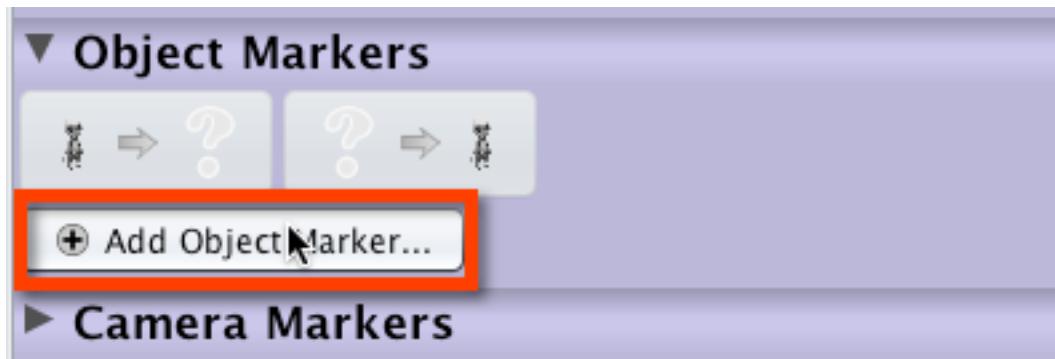


Figure 3.b.3 Add Object Marker ...

A pop-up dialog box provides an opportunity to give the marker a meaningful name, such as `firstPosition` as shown in Figure 3.b.4.

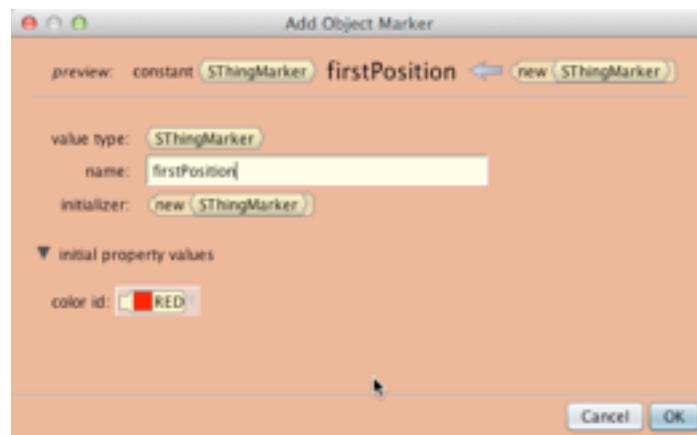


Figure 3.b.4 Enter a meaningful name for the object marker

When the name is entered, press the Enter key. Alice creates a set of axes to represent the object marker. The axes marker is automatically positioned at the pivot point of the object, as shown in Figure 3.b.5. The object marker has the same orientation as the object.



Figure 3.b.5 An axes object represents an object marker

### Moving an object to a marker in the Scene editor

In this example, we created a second marker at the top of the asteroid, as shown in Figure 3.b.6. To move an object from its current position to a marked position, first select (from the list of object markers) the marker to which the object will move. In Figure 3.b.6, the topOfAsteroid marker has been selected. Notice that the question symbols in the two object marker buttons have been replaced with an object (in this example, an alien) and an axes icons.



Figure 3.b.6 Step 1: Select the targeted marker

Secondly, click the object => marker button (left of the two small buttons), as shown in Figure 3.b.7, to move the object to the selected marker.

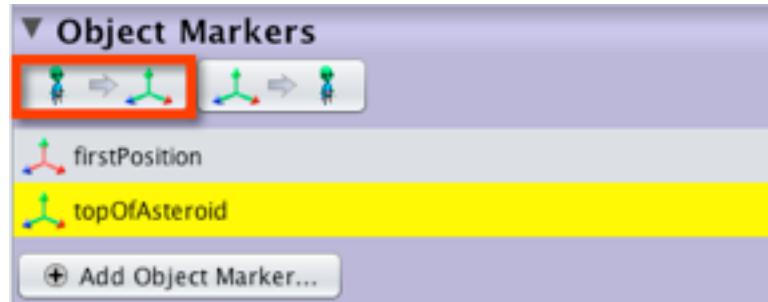


Figure 3.b.7 Step 2: Click object => marker button (left)

The object will immediately move and orient to the targeted marker. In this example, the alien moved to the top of the asteroid, as shown in Figure 3.b.8.



Figure 3.b.8 Result of moving an object to an object marker

**NOTE:** *The Undo button can be used to reverse an action with a marker, if necessary.*

## *Repositioning an object marker*

To reposition an object marker from its current position to the current location of an object, first select (from the list of object markers) the marker to be repositioned. In Figure 3.b.9, the firstPosition marker in the list has been selected. Then, click on the marker => object button to move the marker to the selected object, as shown in Figure 3.b.10.

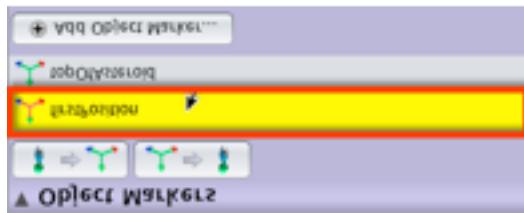


Figure 3.b.9 Select marker to be repositioned



Figure 3.b.10 Click marker => object button (right)

## HOW TO POSITION AND RESIZE AN OBJECT IN THE SCENE EDITOR

[Video: Using Handles to Position Objects](#)

The purpose of this section is to introduce the Scene editor's Undo and Redo buttons and to illustrate the use of handles (ring and arrow mouse controls). Three of the handles create rings and arrows for using the mouse to interactively position an object in a scene. A fourth handle is a mouse control for resizing an object.

***Hint: We recommend using a mouse for working with handles in the Scene editor. A touchpad on a laptop is also usable, but takes more patience.***

## *Position*

The position of an object in a scene has two components: (1) An object's (x, y, z) coordinate location relative to the center of the virtual world, and (2) An object's orientation. The (x,y,z) coordinate location was described in the previous section of this guide. In this section, the primary focus is on setting an object's orientation.

## *Orientation*

An Alice object lives in 3D space and thereby has a sense of direction in three dimensions. An object's senses of up and forward define its orientation. In Figure 3.c.1, an axes object has been embedded in the hare to illustrate the hare's sense of direction. The green arrow points up, the white arrow is forward, the blue arrow backward, and the red arrow right. Although we described the axes as though there were four separate arrows, this is not really true. The forward and backward arrows are actually just one continuous arrow but the two portions of the arrow are painted different colors to provide a better visual perspective.



*Figure 3.c.1 Orientation is defined by an object's sense of up and forward directions*

Orientation is important for an Alice object because motions such as move, turn, and roll are specified in terms of direction. For example, the hare may be told to move forward or move up. When an object performs a motion instruction, it does so relative to its own orientation. In this example, if harry is told to move left he will move to his left. To be clear, in the scene shown in Figure 3.c.1 above, harry would move to his left which is to the right of the scene as seen by the camera. As a rule of thumb, an object's motion is self-centric.

## *Undo and Redo buttons*

A sense of "freedom to play" when setting up a scene is provided by two buttons, Undo and Redo, in the upper right corner of the Scene editor, as shown in Figure 3.c.2.

Undo provides the ability to "erase a mistake." A click on the Undo button backtracks the most recent action and the state of the scene backs up one step, removing it. It is possible to click Undo repeatedly, backtracking all the way to the initial state of the project when it was first opened in this session (but not into previous sessions that were saved and later reopened).

Redo provides the ability to "change your mind." Click the Redo button to reverse the action of an Undo. Redo also provides the ability to repeat an action.



Figure 3.c.2 Undo and Redo buttons in the Scene editor

**Hint:** Use the Undo and Redo buttons to make it fun to set up a scene ... without tension or fear of breaking something.

## Handles

By default, the mouse can be used to click and drag an object forward/backward and left/right on the horizontal plane in a scene. Handle style controls create rings and arrows that can be used to modify the mouse's drag action in the Scene editor. Each handle action is summarized in Figure 3.c.3.



Figure 3.c.3 Handles change the drag action of a mouse on an object

## Single ring

When an object is first added to a scene, the Handle style displayed is usually a single rotation ring around the pivot point of the object, as shown in Figure 3.c.4. Using the mouse to click on the ring and drag the ring in a clockwise or counterclockwise direction causes the object to mimic the mouse action, rotating in the same direction as the ring is being turned.

Rotating an object with the single ring handle changes the orientation of the object by changing the forward and backward directions. (It is possible, however, for the object to end up facing in the same direction it was originally facing. In this case, the orientation is returned to its original value.)

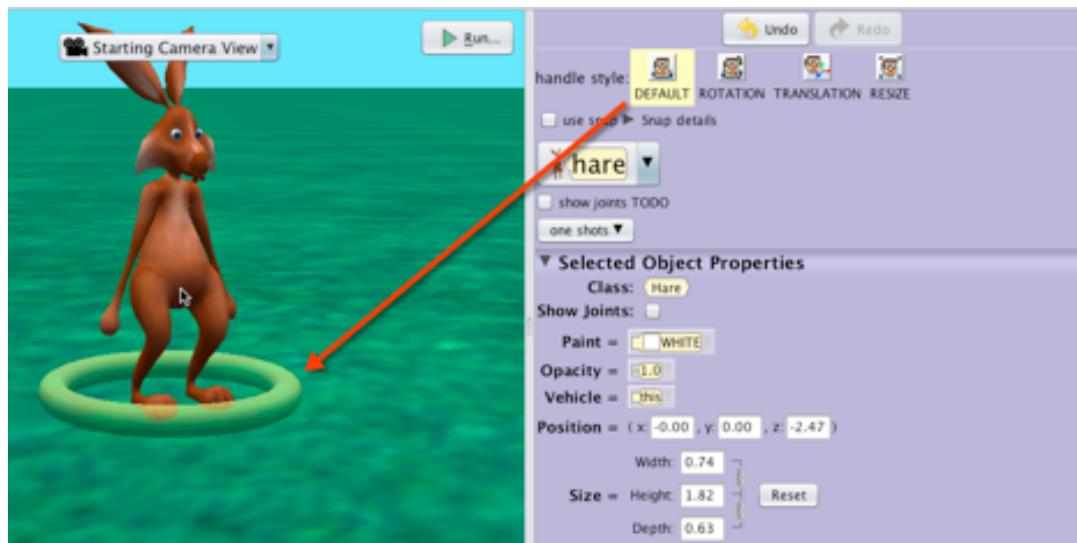


Figure 3.c.4 One ring to rotate an object left/right

## *Three rings*

The three rings handle can be used to turn an object left/right (turn around), turn an object forward/backward (tilt), or roll an object left/right (similar to a door knob), as shown in Figure 3.c.5. Rotating an object with any of the rings changes the orientation of an object. The turn ring changes the forward direction. The tilt ring changes the forward and up directions. The roll ring changes the up direction. (Once again, it is possible to rotate in such a way that the orientation returns to its original value.)

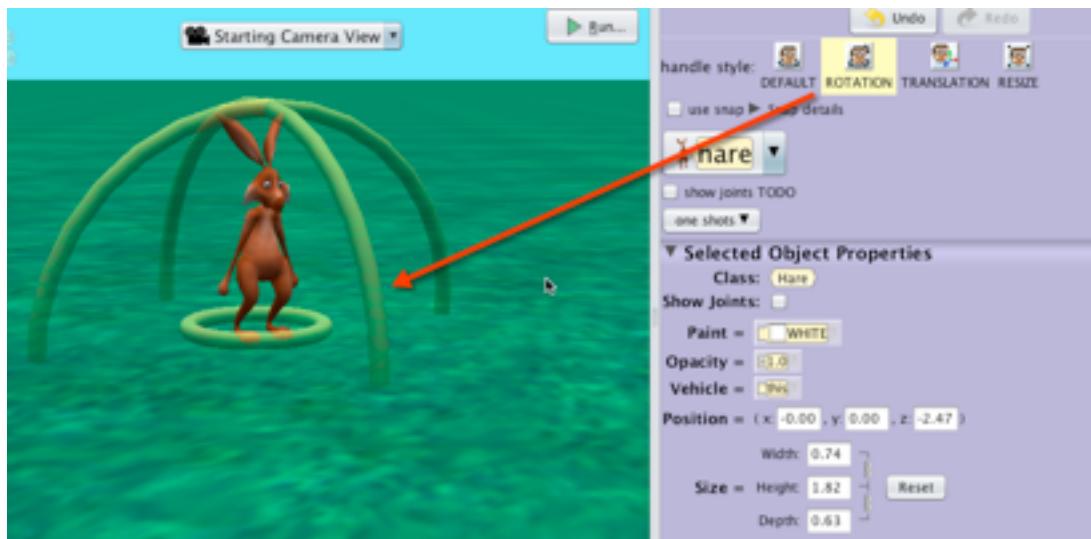


Figure 3.c.5 *Three rings* to turn an object left/right, or forward/backward, or roll left/right

As a short example of the usefulness of the ring handles, in Figure 3.c.6 we added a 3D text object to the scene. Note that the text is somewhat dark. The lighting in a scene is directly overhead. To get better lighting on the text, the text can be tilted slightly backward.



Figure 3.c.6 Text object was added to the example world

Look closely at the text object and the rotation handle button shown in Figure 3.c.7. When the rotation button is clicked, three rings are displayed around the pivot point of the text object.

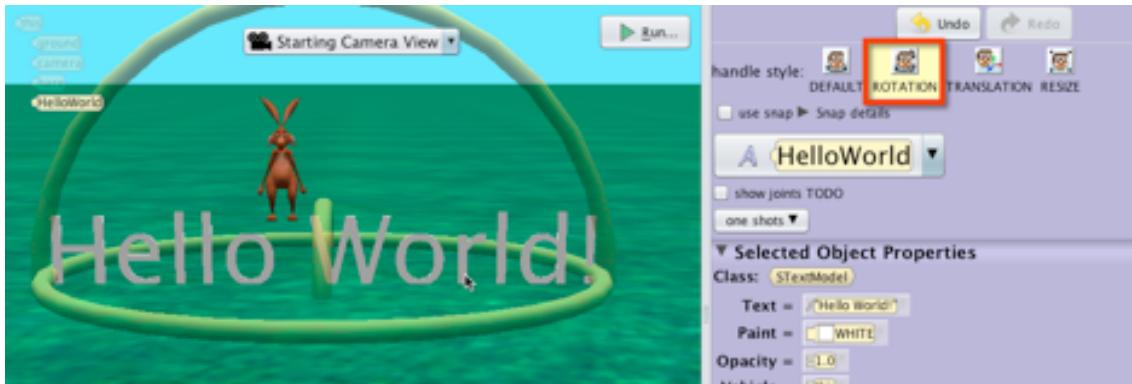


Figure 3.c.7 Three-ringed handle for rotating an object

The forward/back ring was used to tilt the text string slightly backward (toward the back of the scene). The text object appears brighter, as shown in Figure 3.c.8.



Figure 3.c.8 3D Text has better lighting

### Three arrows

The third handle button displays translation arrows (rather than rotation rings). The translation arrows can be used to move an object in any of six directions (up, down, left, right, forward, or backward), as shown in Figure 3.c.9.

The three translation arrows change an object's (x, y, z) coordinate location in the virtual world. However, the orientation of the object remains the same. (As with orientation, it is possible to move an object in such a way that it returns to its original location.)

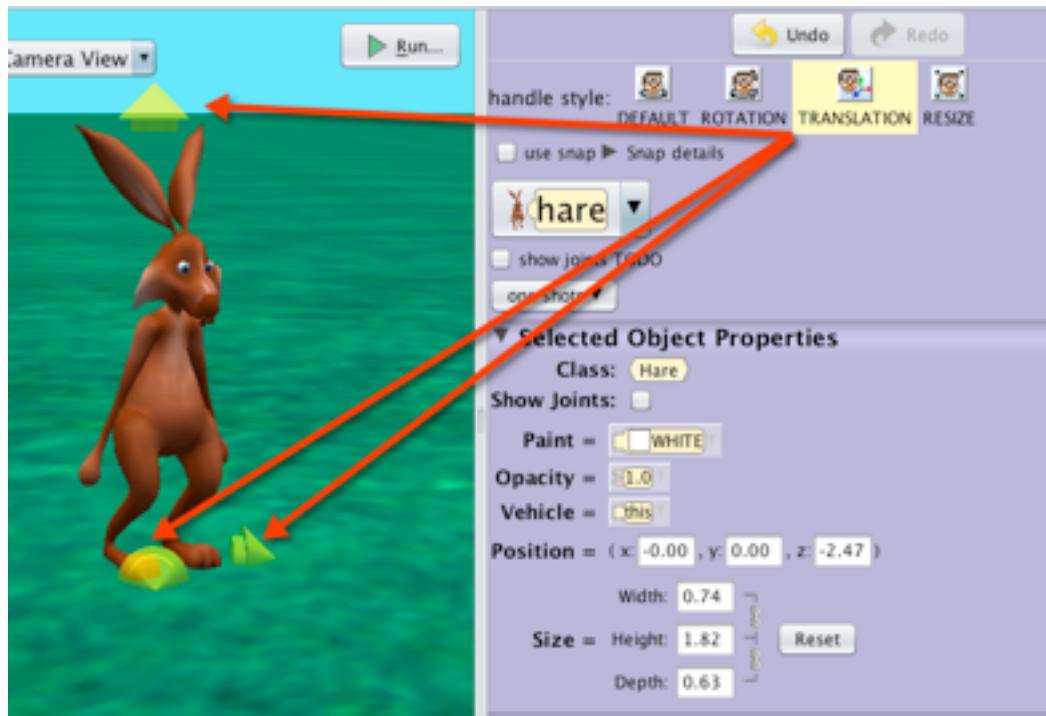


Figure 3.c.9 *Three arrows* to move object up/down, left/right, or forward/backward

The fourth handle style button displays a single translation arrow that can be used to resize an object, as shown in Figure 3.c.10. The single arrow changes the object's size in all directions proportionately. The single translation arrow offers a more free-styling control for resizing as compared to the specific accuracy of the Position (Width, Height, and Depth) property in the Setup.

The single resize arrow does NOT change the orientation of the object.

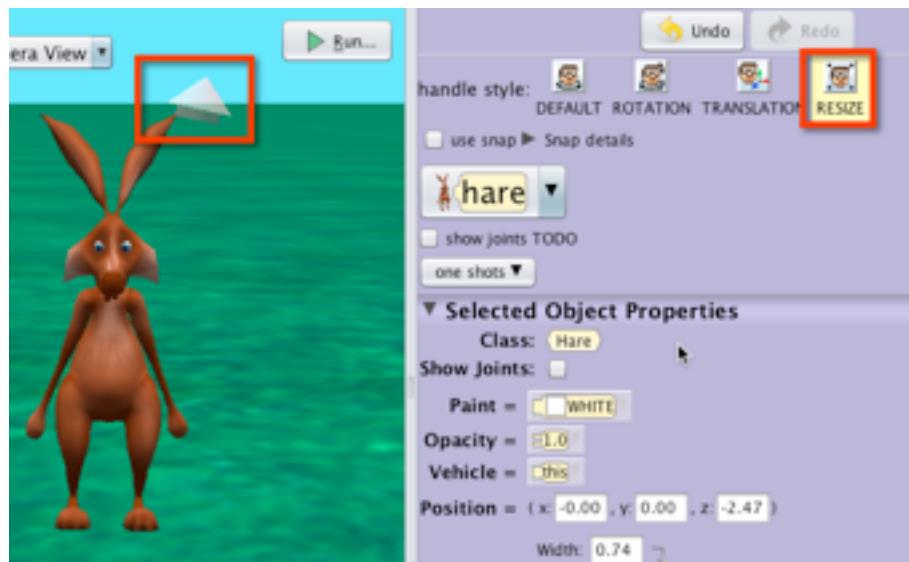


Figure 3.c.10 Single arrow resizes proportionately in all dimensions

A geometric shape has the additional capability of resizing in a single dimension. For example, the cone shape has three resize arrows, as shown in Figure 3.c.11. The upward arrow resizes the cone's height without affecting its width or depth. The arrow at the base resizes the cone's width without affecting its height. The third arrow (diagonally off to the upper side) resizes proportionately in all directions.

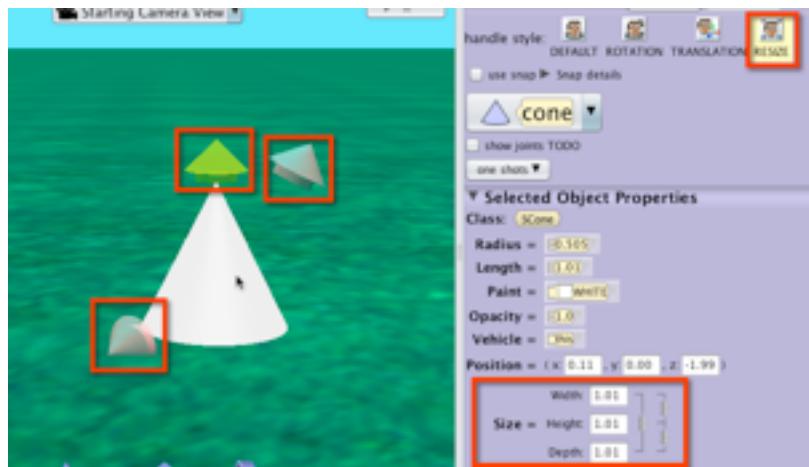


Figure 3.c.11 Three resize buttons for a geometric shape

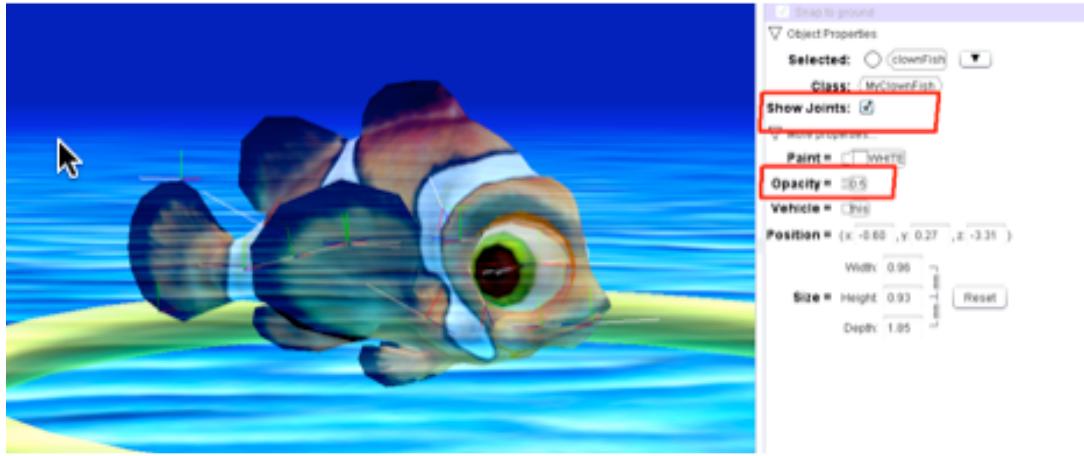
## POSITIONING SUB-PARTS IN SCENE EDITOR

The purpose of this section is to illustrate how to position sub-parts of an object in the Scene editor. In Alice 3, the 3D model classes define objects having an internal skeletal system consisting of joints. Sub-parts (for example an object's head, arms, legs, tails, and other parts) are connected to one another and to the body by these joints. Therefore, a sub-part of an object is positioned by rotating the joints of the skeletal system.

### *How to view the skeletal joints*

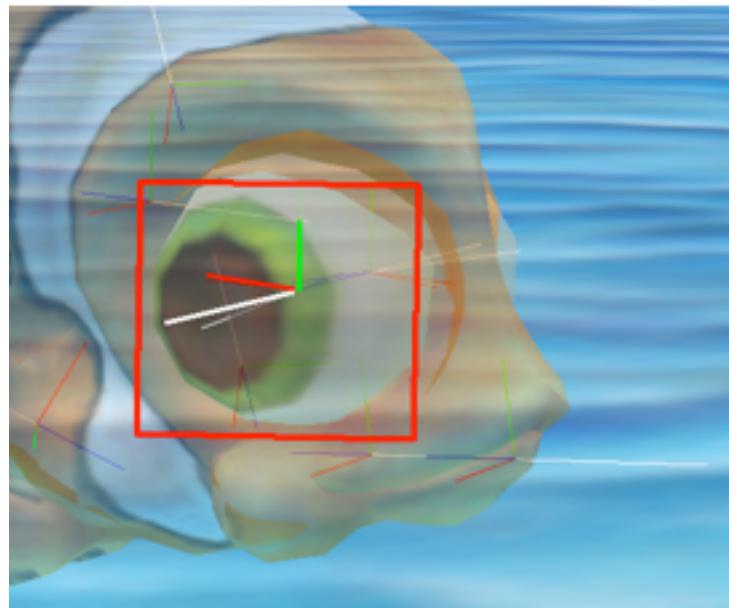
In the real world, joints in a person's skeletal system are not usually visible to the human eye. For example, we have shoulder joints and elbow joints but these skeletal joints are enclosed within the body's skin and muscular tissue. The joints can only be seen by taking an X-ray or by some other medical procedure.

Similarly, Alice object joints can only be seen by using an X-ray-like view. To view the joint positions of an object, select an object in the Object tree and then check the box for the Show Joints option in the Setup. Next, reduce the object's opacity property to a low value such as 0.5. Figure 3.d.1 illustrates an example X-ray-like view of the skeletal system. Notice that the location of each joint is marked with a small axes object.



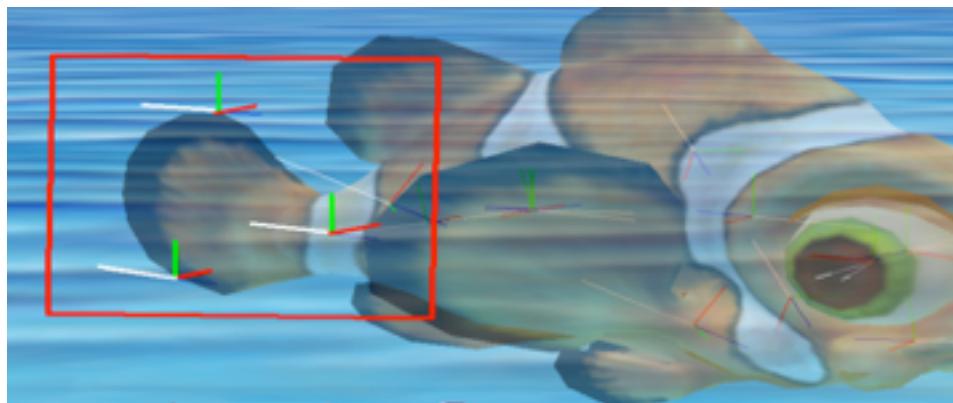
*Figure 3.d.1 An X-ray-like view of the skeletal joint system*

The axes object is for the purpose of showing the location and orientation of each joint. White is forward, red is right, and green is up (as seen by the joint at that location). For most skeletal joints, the orientation is consistent with the functioning of an attached sub-part. For example, Figure 3.d.2 shows a close-up view of the fish's right eye. The important thing to understand is that this fish's eyes face outward (to the side). The eye's joint axes object has a white axis pointing in the direction the eye is “facing” (which is forward as seen by the eye), the green axis is the upright position of the eye within the fish's body (up), and the red axis is to the right of the eye.



*Figure 3.d.2 Orientation is consistent with the action of an attached sub-part*

Some skeletal joints are located in an extended limb (for example, an arm, leg, wing, fin, or flipper). A limb often contains numerous joints which must share the same orientation. For example, Figure 3.d.3 shows a close-up view of the fish's tail. The tail is a limb that is “facing” outward (similar to the fish's eye). The tail is one sub-part but has three joints to provide some flexibility for animation. The three joints share the same orientation, as seen in the axes at each joint. The white axis of each joint is facing outward (forward for the tail), the green axis is the upright position of the tail sub-part as attached to the fish's body (up), and the red axis is the right of the tail as the tail is facing outward.



*Figure 3.d.3 Multiple joints in a limb have consistent orientation, facing away from the body*

### *How to select a skeletal joint*

Sub-parts of an object can be positioned in the Scene editor by selecting the appropriate joint from the Object Parts menu. To view the Object Parts menu, click the Selected tile in the Setup, select the object in the list of objects and pull the mouse cursor over the right arrow to open a cascading menu of joints, as shown in Figure 3.d.4. In this example, the tail was selected for a clownFish object.



Figure 3.d.4 Selecting a joint/part of an object

When a joint is selected, Alice automatically selects the rotation handle and rings are displayed around the selected joint. In this example, the rings are displayed with the fish's tail joint as the pivot point of the tail, as shown in Figure 3.d.5.

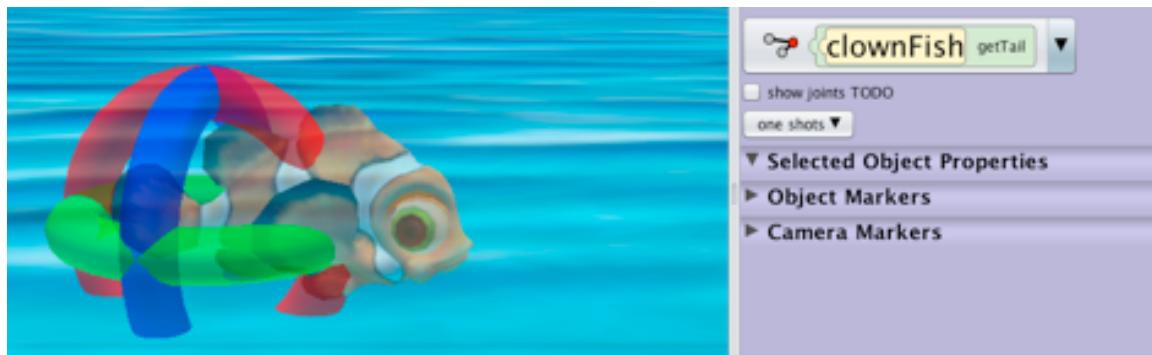


Figure 3.d.5 Three rings for rotating the tail joint

Now the rings can be used to rotate the tail into the desired position, as shown in Figure 3.d.6. The same process can be used to position other joints (and associated subparts) in the object.



Figure 3.d.6 Using a ring to rotate the tail

## RELATIVE POSITIONING WITH CAMERA VIEWPOINTS

[Video: Using Camera Views](#)

The purpose of this section is to demonstrate how to position two or more objects at locations relative to one another in a scene. Alice provides multiple camera viewpoints for relative positioning.

### *Example*

To illustrate, we added a hare (named harry), a Cheshire cat (named chessy), a tiger (named tiggerrr), a tea tray, and a teapot to the example scene shown in Figure 3.e.1. In this example, the goal is to put the teapot on the center of the tea tray.



*Figure 3.e.1 Tea tray and teapot, in original positions*

### *Camera viewpoints*

Positioning the teapot on the center of the tray looks simple. Just drag the teapot onto the center of the tray, as shown in Figure 3.e.2. However, the actual position of one object relative to another object can be deceptive because our view of the scene is only what we see through the camera's lens (the camera viewpoint). In this example, the camera viewpoint is from the front of the scene and it is difficult to see whether the teapot is actually at the center of the tray.



Figure 3.e.2 Teapot is on the tray, but is it in the center?

A Camera viewpoints menu is located at the top center of the scene view. To open the Camera viewpoints menu, click on the down-arrow at the right edge of the button. The menu should drop down to show a list of pre-set camera viewpoints, as illustrated in Figure 3.e.3.



Figure 3.e.3 Camera viewpoint pull-down menu

When an item in the menu is selected, Alice "automagically" takes care of positioning the camera at the selected viewpoint. The Layout Scene View positions the camera upward and at an angle, as shown in Figure 3.e.4. From this viewpoint, it is easy to see that the teapot is not quite on the center of the tea tray.



Figure 3.e.4 Layout Scene View

As shown in Figure 3.e.5, all the handle style tools and the camera navigation controls are available in this view and can be used to reposition objects in the scene.

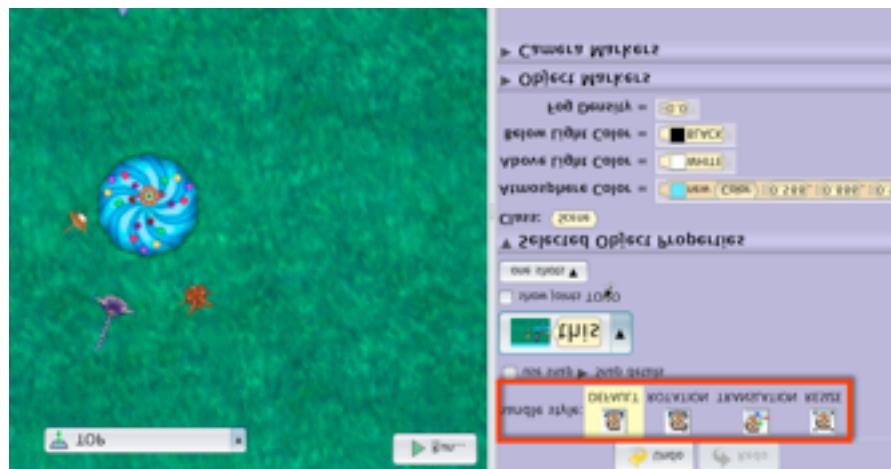


Figure 3.e.5 Camera navigation tools and handles can be used to reposition objects

Figure 3.e.6 shows the result of using the mouse to carefully position the teapot on the center of the tray. Use the Camera Viewpoints menu to put the camera back to the Starting viewpoint.



*Figure 3.e.6 The teapot is now on top and at center of the tea tray*

### *Other Camera viewpoints*

In the example above, the Layout Scene view is all that was needed. However, the camera viewpoints menu offers other options:

#### **TOP view**

The TOP view presents an overhead view of a scene, as shown in Figure 3.e.7. The camera is hovering over the scene and is pointing straight toward the ground in the scene.



Figure 3.e.7 TOP view

In the TOP view, the camera navigation arrows are limited to those used for moving the camera (forward, backward, left, right, up and down), as shown in Figure 3.e.8. The four navigation arrows to the left in Figure 3.e.8 allow the camera to be moved left, right, up, and down, as seen by the camera. The two arrows on the right move the camera forward and backward, as seen by the camera. The SIDE and FRONT views, as described below, also have this limitation on available camera navigation arrows.



Figure 3.e.8 Camera navigation arrows in TOP, SIDE, and FRONT views

### Important concepts:

The motion of the camera that results from using camera navigation arrows is always “as seen by the camera.” This can be surprising when the camera is in TOP, SIDE, or FRONT view. For example, in TOP view, the camera’s “forward” orientation is looking straight down toward the ground. So, moving the camera forward in TOP view actually zooms in closer to the ground.

### SIDE view

The SIDE view presents a camera viewpoint that faces the center point of the ground, from the ground's right side, as shown in Figure 3.e.9. All the handles are available. The camera navigation arrows allow moving the camera forward, backward, left, and right, up and down. But it is not possible to turn the camera to the left or right, forward or backward.



Figure 3.e.9 SIDE view

### FRONT view

In the FRONT view the camera viewpoint faces the center point of the ground, as shown in Figure 3.e.10. The camera navigation arrows allow moving the camera forward, backward, left, and right, up and down. But it is not possible to turn the camera to the left or right, forward or backward.



*Figure 3.e.10 FRONT view*

The black shape (highlighted in a red box) in Figure 3.e.10 above, is a starting camera marker (used internally by Alice to remember the starting position for the camera). The camera marker can often be seen in the other camera viewpoints as well, and the camera marker can be moved, turned, rolled and oriented in the same way as any other Alice object in a scene. Remember, however, that changing Alice's internal camera marker will change the Starting Camera View.

#### **Important concepts:**

It must be emphasized that the Camera viewpoints menu is only available in the Scene editor, for convenience in setting up a scene. The Camera viewpoints listed in the viewpoints menu are not available in the Code editor and cannot be used for creating program code.

## HOW TO ALIGN OBJECTS USING A SNAP GRID

The purpose of this section is to demonstrate how to align two or more objects. Alice provides a grid and one shot methods for alignment and positioning.

*Example*

To illustrate, we will continue with the scene created in the previous section where the teapot was positioned on the center of the tea tray. The current state of this example scene is shown in Figure 3.f.1. In this continuing example, the goal is to position tiggerrr, chessy, and harry all in a straight line behind the tray.



*Figure 3.f.1 Current state of example scene*

To align objects in a scene, activate the Snap grid in the Setup Panel, as shown in Figure 3.f.2. The Snap grid option displays a grid on the ground or water surface in a scene. By default, the grid is set to display grid blocks that are 0.5 meters on a side. In addition, using the mouse to drag-and-drop an object will cause the object to snap into position at the nearest grid point. Rotating an object will cause the object to snap into position at the nearest 30 degree angle. The grid and angle snap values may be set to other values.



Figure 3.f.2 Grid is displayed and snap is active

To use the grid for positioning an object, click and drag the object with the mouse. Alice automatically creates extended, highlighted grid lines for the clicked object, as shown in Figure 3.f.3.

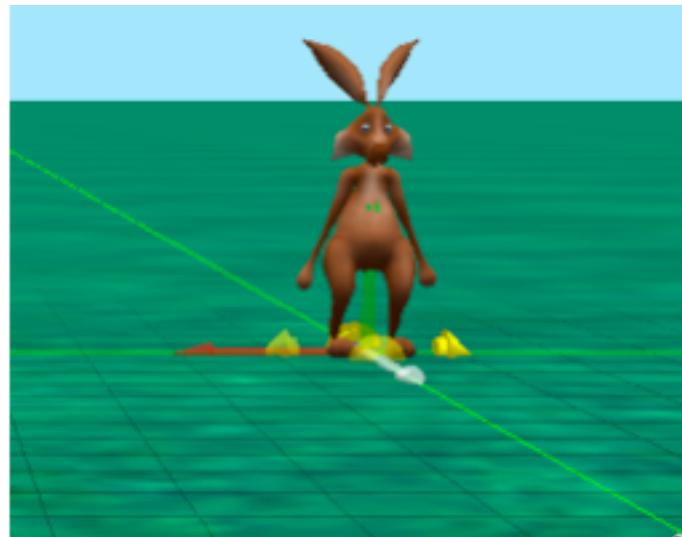


Figure 3.f.3 Highlighted grid lines for the selected object

To align the three objects along one line of the grid, click and drag each object so as to snap to a grid point along the same line, as shown in Figure 3.f.4.



Figure 3.f.4 Using Snap grid lines for alignment

## PRECISE POSITIONING WITH ONE-SHOTS

[Video: Using One-Shot Procedures](#)

The purpose of this section is to illustrate how to use a one-shots menu for alignment and precise positioning of objects and object sub-parts in the Scene editor.

### *Example*

To illustrate precise positioning of objects and sub-parts, we will use the scene shown in Figure 3.g.1. In this example scene, the alien is on a moon surface with his pet robot.



*Figure 3.g.1 An alien and his pet robot*

### *One-shot methods*

One-shot methods are listed in a drop-down menu in the Scene editor. A one-shot method is an action performed “right now” and only once (a “one-shot”) by an object in the scene. There are three techniques for opening a one-shot menu. One technique is to right-click on the name of an object in the Object tree, as shown in Figure 3.g.2, and then select the word “procedures” from the drop-down menu. Procedures are methods that perform an action. One-shot methods, therefore, are procedures.



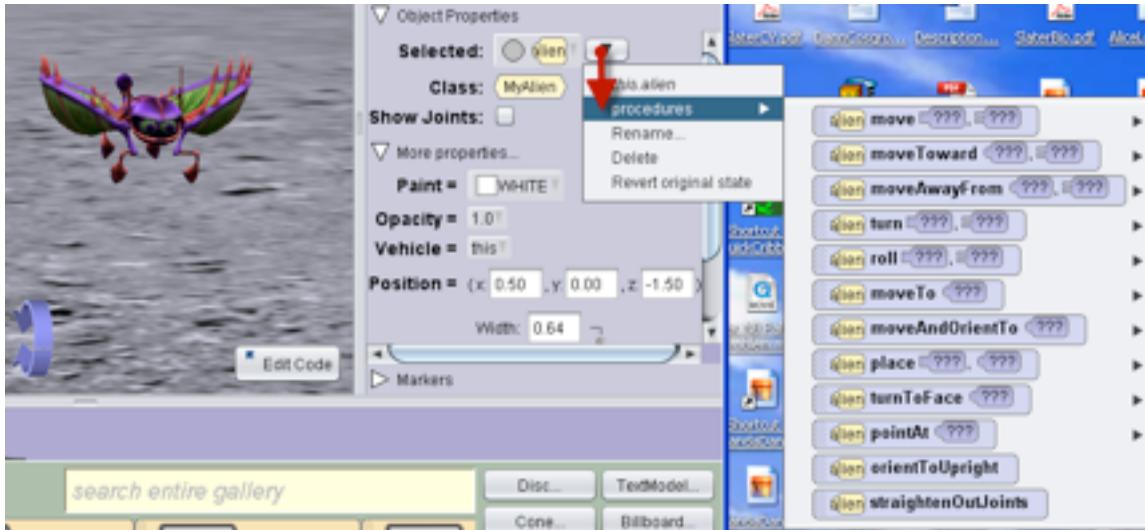
*Figure 3.g.2 Opening one-shots menu by a right-click in the Objects tree*

A second technique is to right-click on the object itself, as shown in Figure 3.g.3.



*Figure 3.g.3 Opening the one-shots menu by right-clicking on the object*

The third technique is to left-click on the pull-down menu button in the Setup, as shown in Figure 3.g.4. Notice that the pull-down menu cascades beyond the right of the Alice 3 window (the monitor's wallpaper can be seen in the background). On computer systems where the monitor is not wide enough, the menu will wrap to the left instead.



*Figure 3.g.4 Opening the one-shots menu by clicking a button in Setup*

As an illustration of using one-shots, let's walk through the steps of precisely positioning alien and buddy (the pet robot) exactly 2 meters apart. The first step is to position the two objects in

the exact same location and orientation. Right click on the alien in the Object tree, select procedures, and then select the moveAndOrientation tile as shown in Figure 3.g.5. In this example, we selected buddy as the target object.

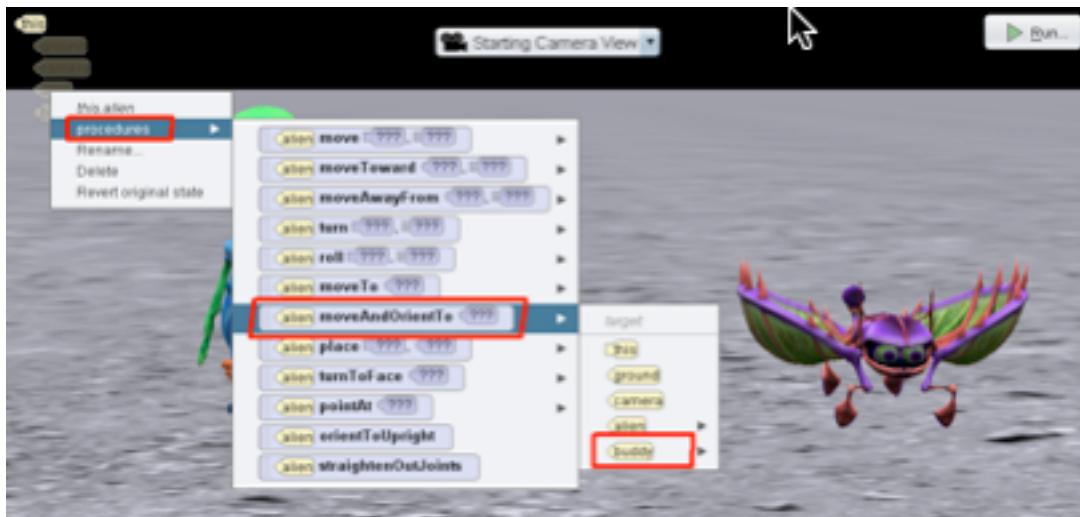


Figure 3.g.5 Select alien.moveAndOrientationTo buddy

The alien will immediately move to the exact same location and orientation as buddy, as shown in Figure 3.g.6.



Figure 3.g.6 Two objects in same location and orientation

The second step is to select a one-shot to move the alien 2 meters to its right. In Figure 3.g.7, we right clicked on the alien tile in the Object tree, selected procedures, the alien.move tile, RIGHT as the direction, and 2.0 meters as the amount.

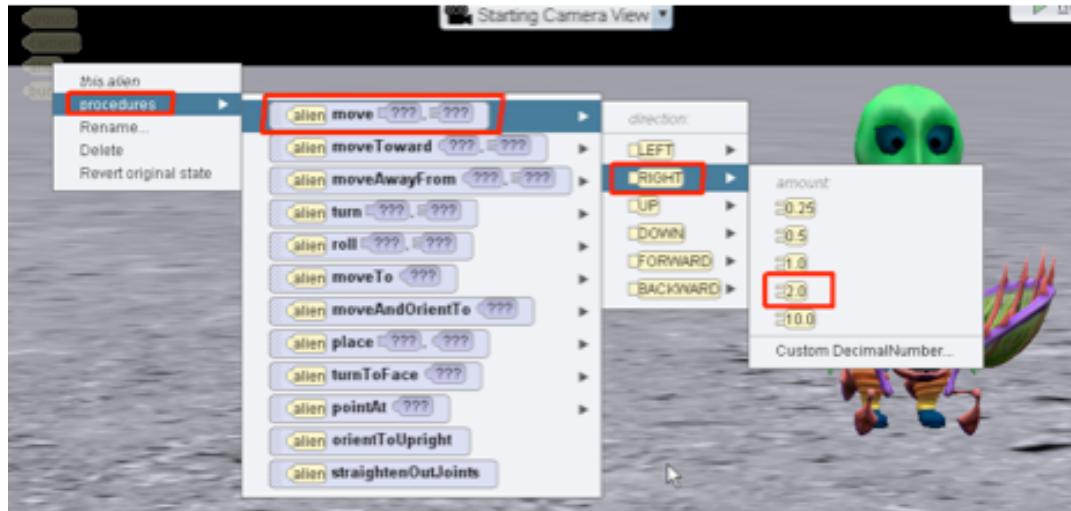
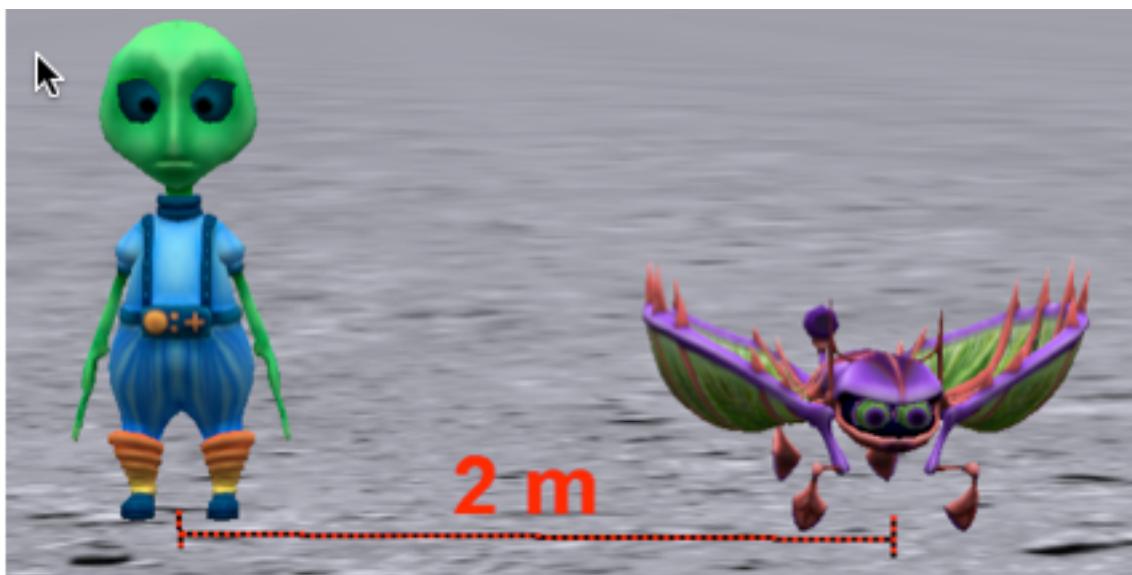


Figure 3.g.7 Positioning the alien exactly 2 meters from the robot

As seen in Figure 3.g.8, the alien and robot are now precisely 2.0 meters apart. It is important to note that the distance is measured as the shortest distance from the center of one object to the center of the other object. The center of an object is its pivot point as it moves, turns, and rolls in animations.



*Figure 3.g.8 Distance is measured center to center*

### *Using a one-shot for positioning an individual joint*

To illustrate using one-shots for positioning an individual joint, we selected the alien's right shoulder joint in the Setup, as shown in Figure 3.g.9.



*Figure 3.g.9 Select the alien's right shoulder joint*

With an object's joint selected, the one-shots menu can be displayed by clicking the down-arrow immediately to the right of the selected joint name, as shown in Figure 3.g.10. Note that the one-shots menu for a joint has only five methods. This is because an object's skeletal joints can be turned and rolled but cannot be "moved."



Figure 3.g.10 Opening one-shots menu for a skeletal joint

In this example, we selected a one-shot to turn the alien's right shoulder backward 0.125 revolutions, as shown in Figure 3.g.11.

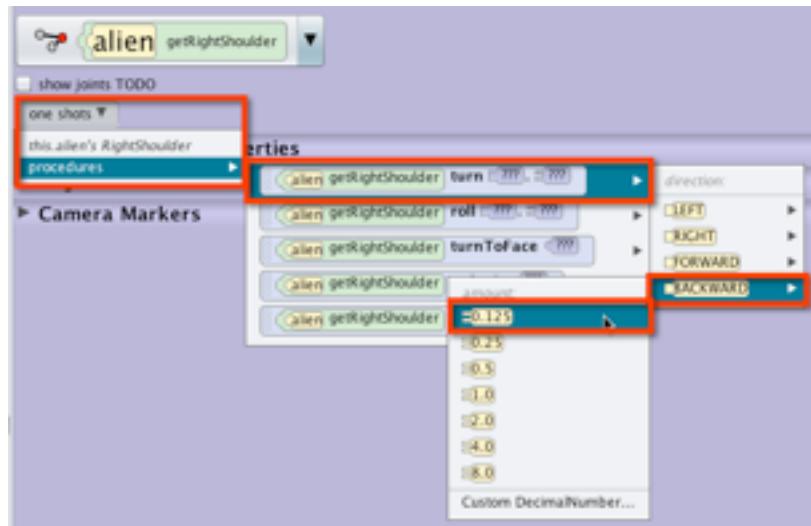


Figure 3.g.11 Turn the alien's right shoulder 0.125 revolutions backward

Turning, rolling, and orienting a skeletal joint has an effect on associated subparts of an object. Figure 3.g.12 shows the result of turning the right shoulder 0.125 revolutions (45 degrees).



Figure 3.g.12 Position of the alien's right arm after turning the right shoulder

## HOW TO CUT, COPY, AND PASTE WITH THE CLIPBOARD

The purpose of this section is to demonstrate using the clipboard to perform cut, copy, and paste in a drag-and-drop Code editor.

Cut, Copy, and Paste items appear in the Edit menu, as shown in Figure 3.h.1. However, these items are placeholders...allowing for possible future implementation.



Figure 3.h.1 Edit options in the Menu bar

Actually, the traditional cut, copy, and paste actions are useful in a text-based editor but are of limited use in a drag-and-drop editor. In Alice, a clipboard is far more useful as a way to store a single graphic tile (one statement) or a block of graphic tiles (multiple statements) for cut, copy, and paste actions.

## Cut

To cut, use the mouse to drag a single graphic tile or a block of graphic tiles into the clipboard, as shown in Figure 3.h.2. In the example shown here, an entire block of tiles is dragged to the clipboard.

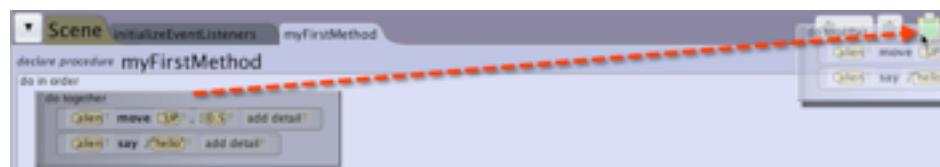


Figure 3.h.2 Drag code to cut to the clipboard

By default, when the mouse is released, the clipboard turns white and the block of tiles is erased from the editor, as shown in Figure 3.h.3. In other words, the default clipboard action is cut.



Figure 3.h.3 Cut removes graphic tiles from a method

## Copy

To copy the statement tiles (instead of cutting), press and hold the Ctrl key (the Option key on Mac) while using the mouse to drag the code into the clipboard, as shown in Figure 3.h.4.

**Hint: release the Ctrl (OPTION) key only after releasing the mouse button.**

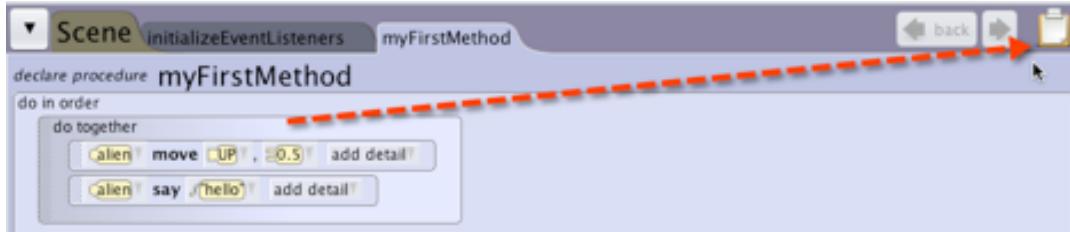


Figure 3.h.4 Drag with Ctrl (Option on Mac) to copy to clipboard

When the mouse is released, the clipboard turns white and the original tiles are NOT erased from the editor, as shown in Figure 3.h.5.



Figure 3.h.5 Copy does not remove code that is saved to the clipboard

### Paste (and remove)

Once a graphic tile has been stored on the clipboard, it can be pasted elsewhere in the editor. In this example shown in Figure 3.h.6, we dragged the code from the clipboard into a method named encounter. Note, in Figure 3.h.6, that the color of the clipboard has returned to its usual brown. By default, dragging a graphic tile out of the clipboard removes the tile from the clipboard. In this example, the clipboard is now empty.

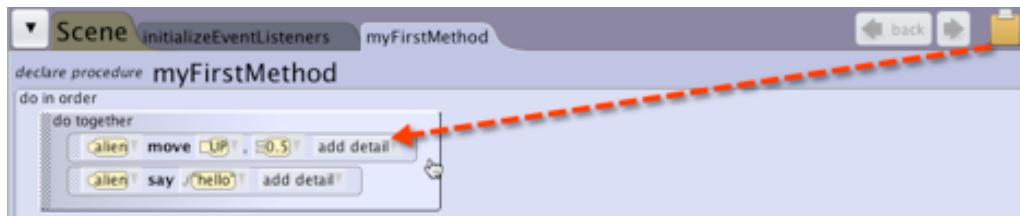


Figure 3.h.6 Paste the code from the clipboard into the editor

### Paste (without remove)

To paste without removing a graphic tile from the clipboard, press and held the Ctrl (Option on the Mac) key while dragging from the clipboard into the editor, as shown in Figure 3.h.7. Note that the color of the clipboard has remained white. This means the clipboard still holds a copy of the tile, allowing it to be pasted more than once.

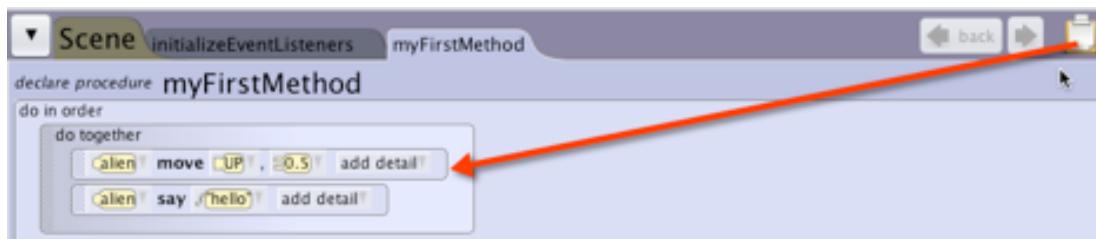


Figure 3.h.7 Paste with Ctrl (Option on Mac) to copy from clipboard to the code editor

### Important Concepts:

Cut, copy, and paste actions can result in scope errors. In the examples used here, we worked with `myFirstMethod` and `encounter` methods -- both of which belong to the `Scene` class. Because this scene contains all other objects in the virtual world (in this example, the `dolphin` and `seaPlant1`), we had no scope errors.

We wish to caution the reader, however, that if code is cut or copied from a method belonging to one class and then pasted into a method belonging to a different class, a scope error may occur. This is not unique to Alice. This is standard protocol for scope-enabled programming languages, whether working in a text editor or a drag-and-drop editor.