APPENDIX A

# Getting Started with Alice

Alice is an innovative software system that allows you to create 3D animations and computer games while learning fundamental programming concepts. With Alice you place graphical objects such as people, animals, buildings, cars, and so on inside 3D virtual worlds. Then you create programming statements that make the objects perform actions. Alice's drag-and-drop program editor makes it easy to create animations with rich interactions between objects.

This appendix serves as a quick reference for using Alice version 2.0. If you need a complete text that teaches programming using the Alice software, see *Starting Out with Alice: A Visual Introduction to Programming*, also published by Addison-Wesley.

## Downloading and Installing Alice

Alice is free software, available from Carnegie Mellon University. You can download the latest version from http://www.alice.org. When you download Alice to your system, you get a file named *Alice.zip*. There is no installation wizard with Alice; you simply extract the contents of this file in the location where you want to install the software.

When you extract the contents of *Alice.zip* you will get a folder named *Alice*. Inside this folder you will find an executable file named *Alice.exe*. Double-click this file to run Alice.

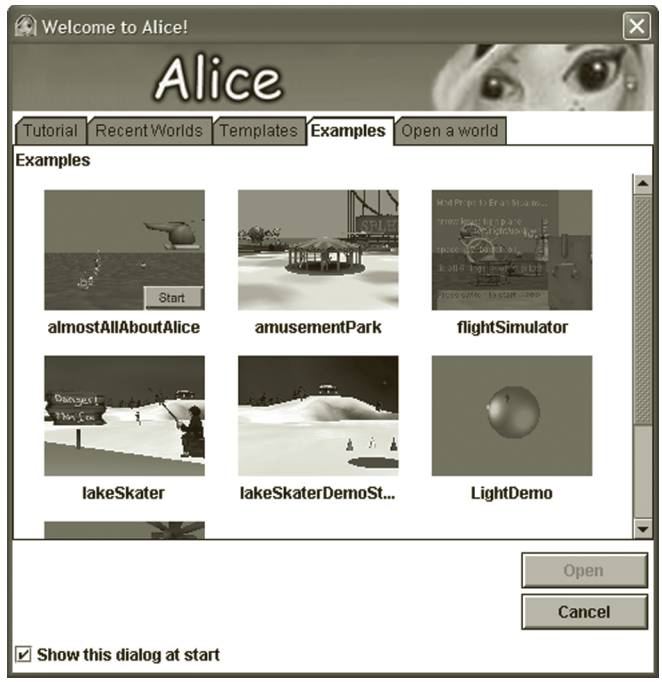
**TIP:** You will probably want to create a shortcut to the *Alice.exe* file on your desktop. Right-click the file and then select *Send To→Desktop (create shortcut)* from the menu. To start Alice double-click the shortcut that appears on the desktop.

## Using the *Welcome to Alice!* Dialog Box

When you start Alice the splash screen shown in Figure A-1 will display for a few seconds. When the software is fully loaded you should see the *Welcome to Alice!* dialog box, as shown in Figure A-2.

**Figure A-1** The Alice splash screen


**Figure A-2** The *Welcome to Alice!* dialog box




**NOTE:** If you do not see the *Welcome to Alice!* dialog box on your system, then Alice has been configured so it will not display the dialog box at startup, which might be the case in a shared computer lab. You can display the dialog box by clicking *File* on the menu bar, and then clicking the *New World* or *Open World...* menu items.

Note that at the bottom of the *Welcome to Alice!* dialog box there is a *Show this dialog at start* check box. Make sure this check box is checked so the dialog box will be displayed each time you start Alice.

Near the top of the *Welcome to Alice!* dialog box you will see a set of tabs labeled *Tutorial, Recent Worlds, Templates, Examples,* and *Open a world*. The following are brief descriptions of what you get when you click these tabs:

*Tutorial*—Click this tab and you will see a set of four Alice worlds that work as tutorials. These tutorial worlds guide you through the basic features of Alice. If you want to run the tutorials, click the *Start the Tutorial* button to execute them in order, or select and open any of the worlds individually.

*Recent Worlds*—Click this tab and you will see thumbnail images of the worlds that were most recently opened on your system. You can quickly open any world shown in this tab by selecting its thumbnail image and then clicking the *Open* button. You will not see any worlds listed here if you have not yet opened any worlds.

*Templates*—Click this tab and you will see a set of templates that you can use to create a new world. The templates are named *dirt*, *grass*, *sand*, *snow*, *space*, and *water.* Each template gives you a ground surface and a sky color.

*Examples*—Click this tab and you will see thumbnail images of example worlds that have been created by the developers of Alice.

*Open a world*—Click this tab and you will see a dialog box that allows you to open an Alice world. With this tab you can browse your local system or any attached network drive for Alice worlds. Note that Alice worlds are saved in files that end with the *.a2w* extension. (The *.a2w* extension signifies that the file contains an Alice version 2.0 world.)

## The Alice Environment

In Alice the screen that you work with is referred to as the *Alice environment*. The Alice environment is divided into the following areas: the Toolbar, the World View Window, the Object Tree, the Details Panel, the Method Editor, and the Events Editor. In addition, the toolbar area provides a trashcan icon and one or more clipboard icons. The locations of these different areas and icons are shown in Figure A-3. In the figure, *SnowLove*, one of the example worlds, is opened. Brief descriptions of each area in the Alice environment follow:

***Toolbar***—The toolbar provides a *Play* button that plays your virtual world, an *Undo* button that undoes the previous operation, and a *Redo* button that repeats the operation that was most recently undone.

***Trashcan***—Next to the buttons on the toolbar there is a trashcan icon. You delete items by dragging them to the trashcan.

***Clipboards***—The clipboard provides a place to store a copy of something. In Alice clipboards you can store copies of objects, instructions, methods, and events. To store a copy of an item in a clipboard, you click and drag the item to the clipboard. When a clipboard contains an item, it appears as if it has a white sheet of paper on it. In Figure A-3 the leftmost clipboard shows an example. To paste the item that is stored in a clipboard, you click and drag the clipboard icon to the location where you want to paste the item. If you want to empty a clipboard, you click and drag it to the trashcan.

By default, Alice shows only one clipboard. To change the number of available clipboards you click the *Edit* menu and then click *Preferences*. On the dialog box that appears, you click the *Seldom Used* tab and then change the number that appears next to *number of clipboards*.

***World View Window***—The World View Window shows a view of your virtual world. Each virtual world has a camera; the World View Window acts as the camera's viewfinder and also provides controls for moving and rotating the camera.

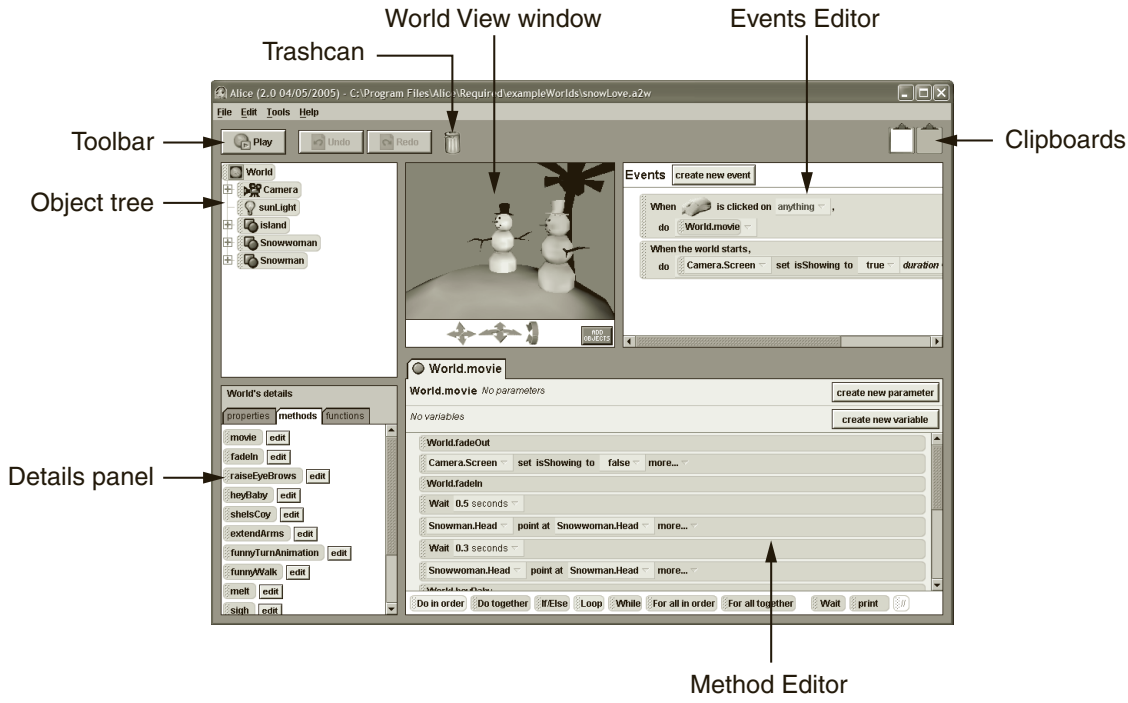
***Object Tree***—The Object Tree holds a list of all the objects in the world. Each object in the world is represented by a *tile*, which is simply a small rectangular icon. Tiles are used extensively in the Alice environment to represent numerous things.

***Details Panel***—The Details Panel shows detailed information about an object that has been selected in the World View Window or in the Object Tree.

***Method Editor***—The Method Editor is where you create methods (a set of instructions that causes some action to take place). You create methods by arranging tiles in the Method Editor.

***Events Editor***—An event is some action that takes place while the world is playing, such as clicking the mouse or pressing a key. Alice is able to detect when various events take place. You can use the Events Editor to specify an action that is to take place when a specific event occurs.

**Figure A-3** Parts of the Alice environment



# Playing a World

When you click the *Play* button, a separate *World Running...* window appears and the world's animation will play out in that window. For example, Figure A-4 shows the *SnowLove* example world playing.

**Figure A-4** The *SnowLove* world playing


Notice the toolbar at the top of the *World Running...* window. The following are brief descriptions of the items that appear on the toolbar:

***Speed Slider Control***—This controls the speed at which the world is played. When the slider is set to 1×, the world plays at normal speed. Moving the slider to the right increases the speed up to 10 times its normal speed.

***Pause Button***—Clicking the *Pause* button causes the world to pause.

***Resume Button***—Once a world has been paused with the *Pause* button, you can click the *Resume* button to resume playing.

***Restart Button***—Clicking the *Restart* button causes the world to start playing again.

***Stop Button***—Clicking the *Stop* button causes the world to stop playing and closes the *World Running...* window.

***Take Picture Button***—Clicking the *Take Picture* button captures an image from the world and saves it in a file. The dialog box that appears when you click the *Take Picture* button reports the name and path of the file containing the image.

## Creating a New World and Adding Objects to It

To create a new world, you click *File* on the menu bar and then click the *New World...* menu item. This displays the *Welcome to Alice!* dialog box, as shown in Figure A-2. (By default, this dialog box is also displayed when you start Alice.) Make sure the *Templates* tab is selected, as shown in Figure A-5.

The *Templates* tab shows a set of templates named *dirt*, *grass*, *sand*, *snow*, *space*, and *water* that you can use to create a new world. When you select a template from this dialog box and then click the *Open* button, Alice will create a ground surface and set the color of the sky. For example, Figure A-6 shows a world that was created with the sand template.

**Figure A-5** The *Welcome to Alice!* dialog box

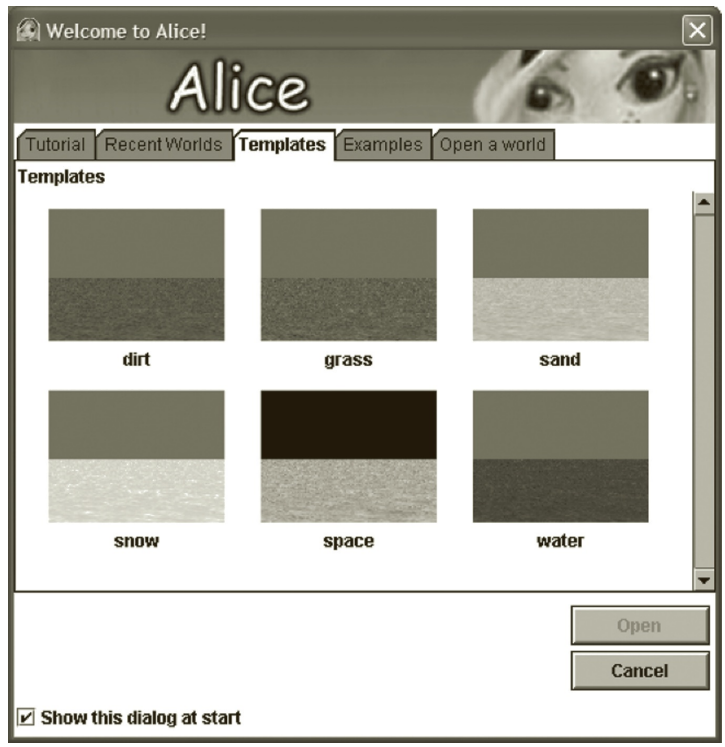



Figure A-6 shows the *Add Objects* button just below the World View Window. When you click this button the Alice environment changes to scene editor mode and opens a gallery, as shown in Figure A-7. A *gallery* is an assortment of different *types* of objects and is organized into various collections of objects such as animals, buildings, furniture, and people.

Alice provides two galleries: a local gallery and a Web gallery. The *local gallery* is stored on your computer and is installed with the Alice software. It provides a good sampling of object types and should be adequate for many of your projects. The *Web gallery* is maintained by the creators of Alice and may be accessed if your computer is connected to the Internet. It provides a much more extensive collection of object types than the local gallery.

**Figure A-6** A world created with the sand template

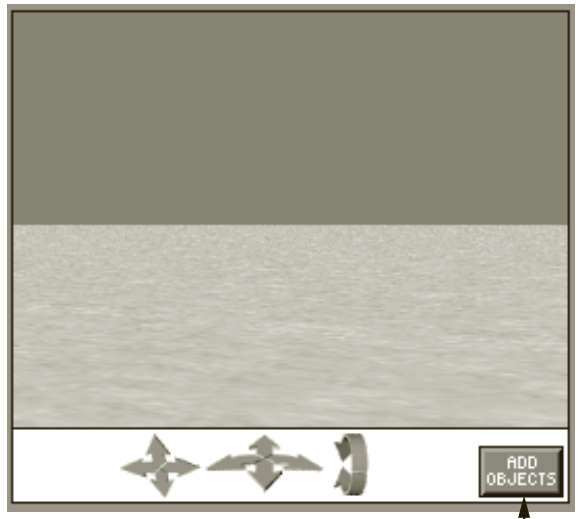



**Figure A-7** Alice in scene editor mode

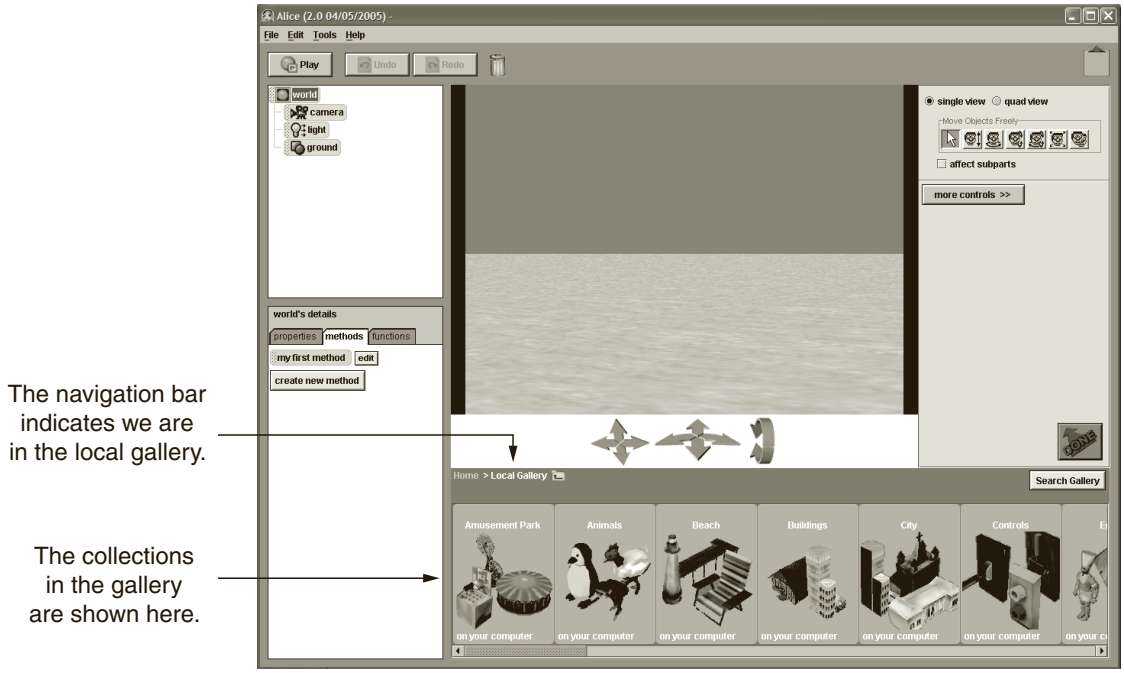
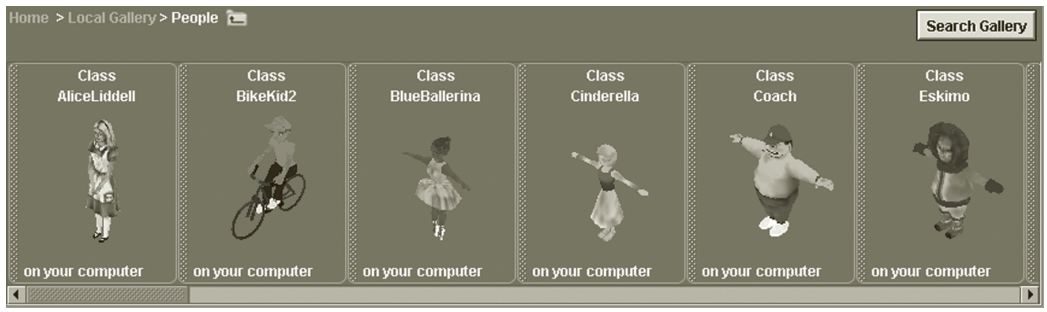


Figure A-7 points out a navigation bar that indicates which gallery and collection is currently displayed. Below the navigation bar are thumbnail images for the collections in the gallery. To open a collection and see the object types it contains, you click the collection's thumbnail image. For example, one of the collections is named *People*. It contains various types of people objects, as shown in Figure A-8.

**Figure A-8** Some of the object types in the `People` collection



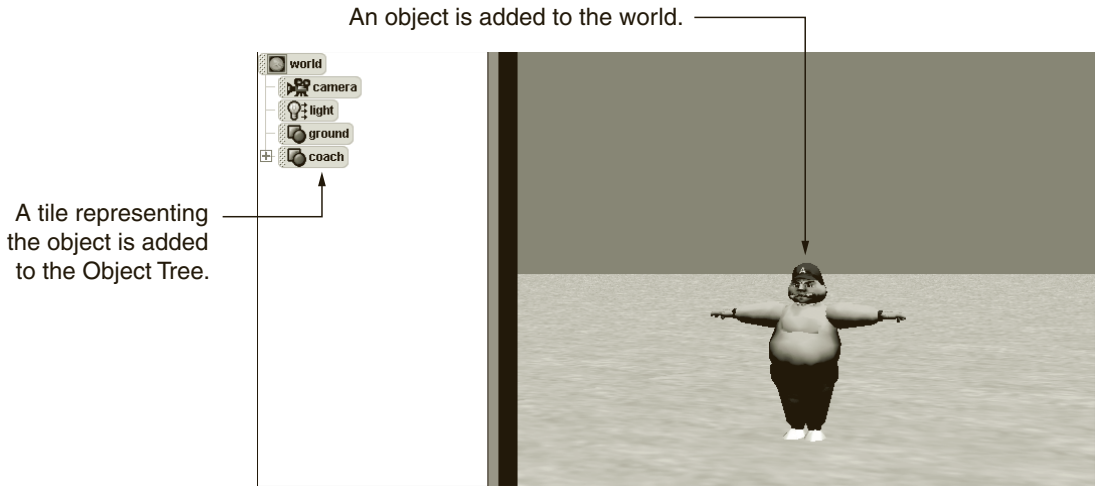

One way to add an object to the world is to click the thumbnail for that object type. You will then see an information window for the object. For example, if you click the thumbnail for the `Coach` object type, you will see the information window shown, as shown in Figure A-9. Click the *Add instance to world* button to add an object of this type to the world.

**Figure A-9** Information window for the `Coach` object type


Another way to add an object to the world is to click and drag the thumbnail for the object type into the World View Window. When you release the mouse button (with the mouse pointer inside the World View Window) an object will be created.

After you add an object to a world, you should see a tile for the object in the Object Tree, as shown in Figure A-10. Each object in a world has a name, and the object's tile will show the name that Alice assigned to the object. You can rename the object by right-clicking its tile and then selecting *rename* on the menu that appears.

**Figure A-10** An object is added to the world

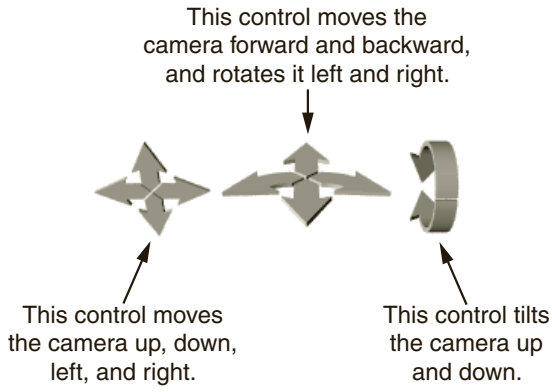



## Moving the Camera in the Alice Environment

The three camera controls shown in Figure A-11 appear just below the World View Window. You use these controls to move the camera around in the world and point it in different directions. The control on the left moves the camera up, down, left, and right. The control in the center moves the camera forward and backward, and rotates the camera left and right. The control on the right tilts the camera up and down.

Notice that each of the controls shows a set of arrows. You manipulate these controls by clicking and dragging the arrow that points in the direction that you want to move, rotate, or tilt the camera. You can make the camera move faster by dragging the mouse pointer away from the center of the camera control. The farther you drag the pointer away from the center of the camera control, the faster the camera will move.

**Figure A-11** Camera controls



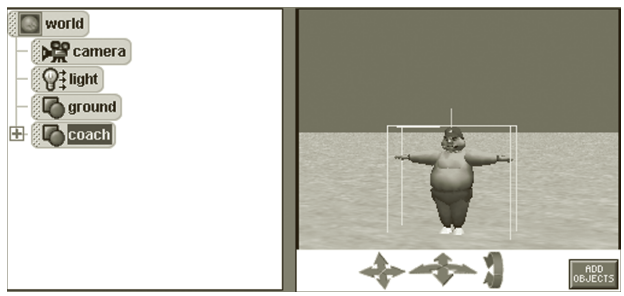
# Selecting Objects

To work with an object in the Alice environment, often you first have to select the object. The following are the ways to select an object:

- Click its tile in the Object Tree
- Click the object in the World View Window

When you select an object, a box appears around it in the World View Window, as shown in Figure A-12. (On your screen the box will be yellow.) This *bounding box* indicates that the object is selected. Also, the object's tile in the Object Tree will appear highlighted, as shown in the figure.

**Figure A-12** The `coach` object is selected




### Object Subparts

Objects are commonly made of other objects, which are referred to as *subparts*. When a plus sign appears next to an object tile in the Object Tree, it means that the object is made of subparts. For example, look at the Object Tree shown in Figure A-12 and notice that a plus sign appears next to the tile for the `coach` object. You can click the plus sign next to an object to expand the tree and see the tiles for the subparts. The plus sign then turns into a minus sign, which hides the inner objects when clicked.

Figure A-13 shows the Object Tree expanded to reveal that the `coach` object is composed of numerous subparts. One of these subparts, the `head`, is selected.

# Properties

Each object in an Alice world has *properties*, which are values that specify the object's characteristics. Once you have placed an object in an Alice world, you can adjust its properties until it has the characteristics you desire. To change an object's property you perform the following steps:

- Select the object
- In the Details Panel select the *properties* tab, as shown in Figure A-14
- Change the value of the desired property (to change a property's value, click the down-arrow that appears next to the property's value)

**Figure A-13** An object subpart selected

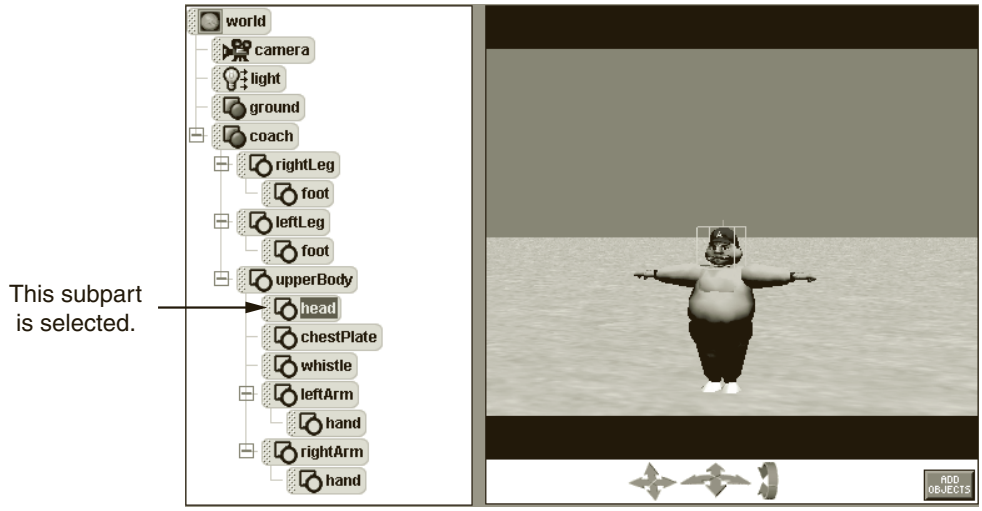



**Figure A-14** Properties displayed in the Details Panel


# Primitive Methods

A *method* is a set of instructions that causes some action to take place. In Alice all objects have a common set of built-in methods for performing basic actions. These methods, which are known as *primitive methods*, cause objects to move, turn, change size, and do other fundamental operations.

While you are creating an Alice world you can immediately execute an object's primitive methods by right-clicking the object in the World View Window or the object's tile in the Object Tree. Then you select *methods* from the menu that appears. Another menu appears showing a list of methods that you can immediately execute in the World View Window. Figure A-15 shows an example of these menus. Table A-1 describes each of the primitive methods shown on the menu.

**Figure A-15** Selecting a primitive method

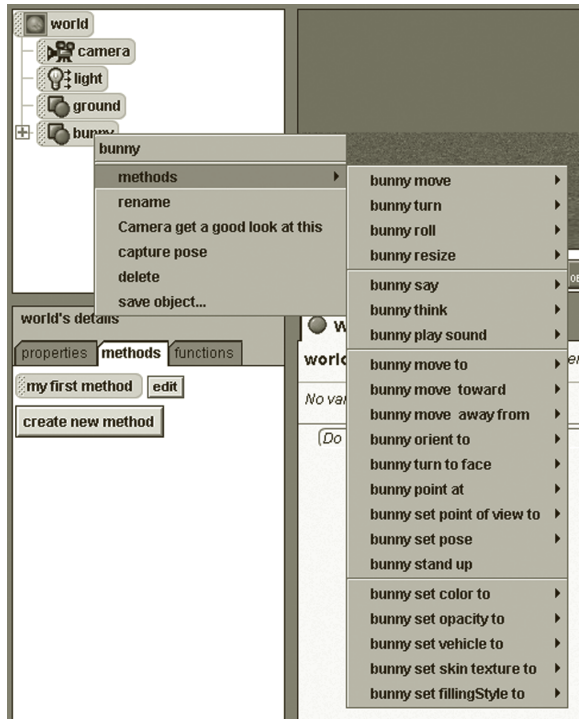



**Table A-1** Primitive methods

| Method Name | Description |
|---|---|
| `move` | This method causes the object to move up, down, left, right, forward, or backward. You specify the direction and distance that you want the object to move. |
| `turn` | This method causes the object to turn toward the left, right, forward, or backward. You specify the amount you want the object to turn in revolutions. |
| `roll` | This method causes the object to roll toward the left or the right. You specify the amount you want the object to roll in revolutions. |
| `resize` | This method changes the object's size by a specified amount. |
| `say` | This method causes a cartoon-like speech bubble containing a message to be displayed, as if the object were saying the message. |
| `think` | This method causes a cartoon-like thought bubble containing words to be displayed, as if the object were thinking the words. |
| `play sound` | This method plays a sound. You can specify one of the sounds that Alice provides or you can import any MP3 or WAV file. |

(*continues*)

**Table A-1** Primitive methods (*continued*)

| Method Name | Description |
|---|---|
| `move to` | This method causes the object to move to another object. When the method completes, both objects' center points will be in the same location. |
| `move toward` | This method causes the object to move in the direction of another object. You specify the distance to move in meters. |
| `move away from` | This method causes the object to move away from another object. You specify the distance to move in meters. |
| `orient to` | This method orients the object in the same direction as another specified object. When this method executes the object will turn so its up, right, and forward axes are aligned with the axes of the specified object. |
| `turn to face` | This method causes the object to turn so it is facing another object. |
| `point at` | This method is similar to the `turn to face` method, except the object will be tilted so its forward axis is "aiming" at the specified object's center point. |
| `set point of view to` | This method sets the object's point of view to that of another object. It is commonly used with the camera to move it to the location of another object, and give a view from that object's point of view. |
| `set pose` | Alice allows you to position an object and its subparts in a certain way and then capture that as a pose. This method causes the object to assume a pose that was previously captured. |
| `stand up` | This method makes the object "stand up" by aligning the object's up axis with the world's up axis. |
| `set color to` | This method sets the object's `color` property to a specified color, making the object appear in that color. |
| `set opacity to` | This method sets the object's opacity property, which determines the object's transparency. You set this property to some value between 0 percent and 100 percent, where 0 is completely invisible and 100 is completely opaque. |
| `set vehicle to` | This method sets the object's `vehicle` property. The `vehicle` property couples the object with another object. When the other object moves, this object moves with it. |
| `set skin texture to` | This method sets the object's `skin texture` property. The `skin texture` property specifies a graphic image to be displayed on the object. |
| `set fillingStyle to` | The `fillingStyle` property determines how the object is displayed. It has three settings: solid, wireframe, and points. The default setting is solid, which causes the object to be displayed as a solid. When the `fillingStyle` property is set to wireframe, the object is displayed as a wire skeleton that you can see through. When the `fillingStyle` property is set to points, the object is displayed as a set of points. |

Most of the primitive methods require that you specify additional pieces of information. For example, the `move` method causes the object to move, and it requires that you specify two pieces of information: a direction and an amount. These pieces of information are known as arguments—pieces of information that a method requires in order for it to execute.

## Deleting Objects

You can delete an object in an Alice world by performing any of the following operations:

- Right-click the object in the World View Window and then select *delete* from the menu that appears
- Right-click the object's tile in the Object Tree and then select *delete* from the menu that appears
- Click and drag the object's tile from the Object Tree to the trashcan

## Modifying Objects in Scene Editor Mode

When you click the *Add Objects* button, which appears below the World View Window, Alice goes into scene editor mode, in which you can use the mouse to modify the objects in your Alice world. For example, you can use the mouse to move objects, resize objects, rotate objects, and copy objects. Figure A-16 shows the location of the *mouse mode buttons*, which determine the action that can be performed with the mouse.

**Figure A-16** Location of the mouse mode buttons

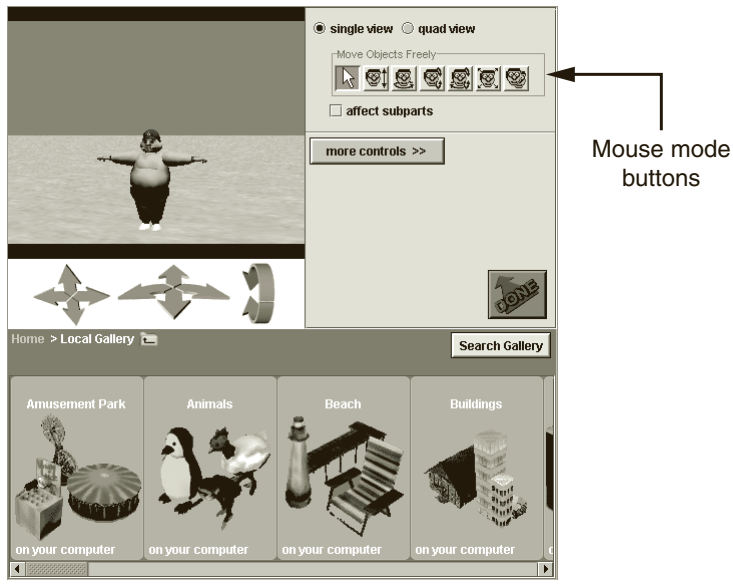


Figure A-17 shows the purposes of the buttons. The following are brief descriptions of each:

*Move Freely*—When this button is selected the mouse can be used to move an object freely in the world. Here are the actions that you can perform:

- To move an object horizontally within the world you simply click and drag it
- To move an object straight up or down, you hold down the Shift key while clicking and dragging the object
- To rotate an object left or right, you hold down the Ctrl key while clicking and dragging the object
- To tumble an object (rotate it left, right, forward, backward, or any combination of these directions), you hold down the Ctrl and Shift keys while clicking and dragging the object

*Move Up and Down*—When this button is selected you can move an object straight up or straight down by clicking and dragging the object.

*Turn Left and Right*—When this button is selected you can rotate an object toward the left or the right by clicking and dragging the object.

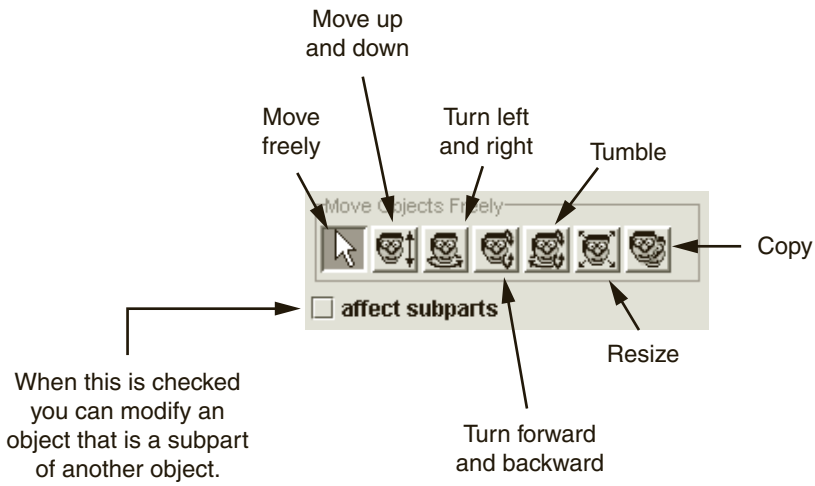
*Turn Forward and Backward*—When this button is selected you can rotate an object forward or backward by clicking and dragging the object.

*Tumble*—When this button is selected you can tumble an object by clicking and dragging the object. This means you can rotate the object right, left, forward, backward, or in any combination of these directions.

*Resize*—When this button is selected you can make an object larger or smaller by clicking and dragging the object.

*Copy*—When this button is selected you can make a copy of an object by clicking the object.

**Figure A-17** The purposes of the mouse mode buttons

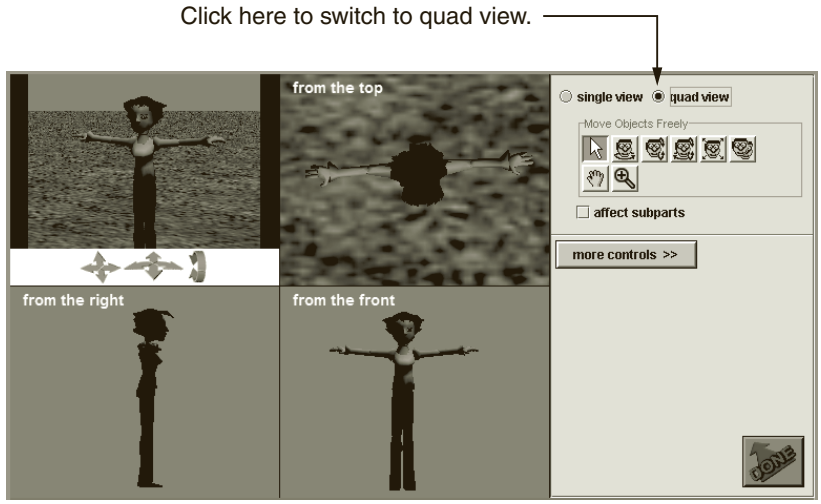


Notice that just below the buttons a check box labeled *affect subparts* appears. By default, this is not checked. When it is not checked the modifications that you make to an object using the *mouse mode* buttons are applied to the entire object. However, if you check the *affect subparts* check box, the modifications are applied only to one of the object's subparts.

## Single View and Quad View Modes

When Alice is in scene editor mode, you can switch the display of the world between single view mode and quad view mode. So far we have been using *single view mode*, which is the default display mode. In single view mode you have one view of the world—the World View Window. In *quad view mode* you have four views of the world: the World View Window, a view from the top, a view from the right, and a view from the front. Figure A-18 shows an example of these views and points out the *quad view* button, which you click to switch to quad view mode.

**Figure A-18** Quad view




You can use the mouse to modify objects in any of the views. If you look carefully at the *mouse mode* buttons while in quad view mode, you'll notice that the *Move Up and Down* button no longer appears because the right and front viewing windows support up and down movement. If you want to move an object up or down while in quad view mode, you simply select the *Move Objects Freely* button and then move the object up or down in either the right view or the front view.

You will also notice that two new buttons appear while in quad view mode: The *Scroll View* button and the *Zoom* button. Often, when you switch to quad view mode the objects in the world will not be fully visible in all of the views. To remedy this you can use the *Scroll View* button to scroll the top, right, or front view. To use the button, follow these steps:

1. Select the *Scroll View* button; the mouse pointer changes into a hand tool
2. Move the mouse pointer into the view you wish to scroll
3. Click and drag the view in the direction you wish to scroll

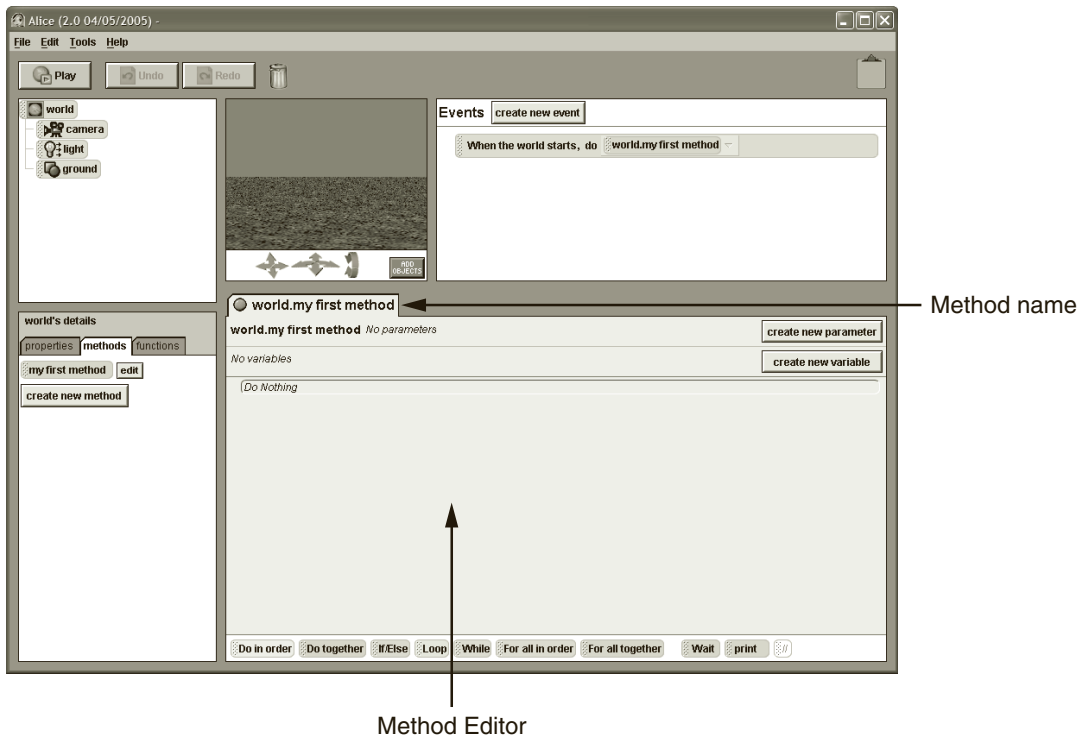
The *Zoom* button allows you to zoom into or out of the top, right, and front views. To use it, follow these steps:

1. Select the *Zoom* button; the mouse pointer changes into a zoom tool
2. Move the mouse pointer into the desired view and position it over the point that you wish to zoom into or zoom out from
3. Zoom by clicking and dragging; if you want to zoom in, drag down or to the right, if you want to zoom out, drag up or to the left

## Writing Methods in Alice

Recall that a method is a set of instructions that causes some action to take place. If you want an action to take place when an Alice world is played, you have to write a method. Figure A-19 shows the location of the Method Editor in the Alice environment, where you write the methods that perform actions in an Alice world.

**Figure A-19** The Method Editor




Notice that a *world.my first method* tab appears at the top of the Method Editor in Figure A-19. All methods have a name, and `world.my first method` is the name of the method

that is currently open in the editor. When you create a new world Alice automatically creates an empty method named `world.my first method`. By default, this method is automatically executed when you play the world.

In Figure A-19 notice that a group of tiles appears at the bottom of the Method Editor. Each of these tiles is an instruction that you can place in the method. Table A-2 describes the instructions represented by these tiles.

**Table A-2** Alice instructions

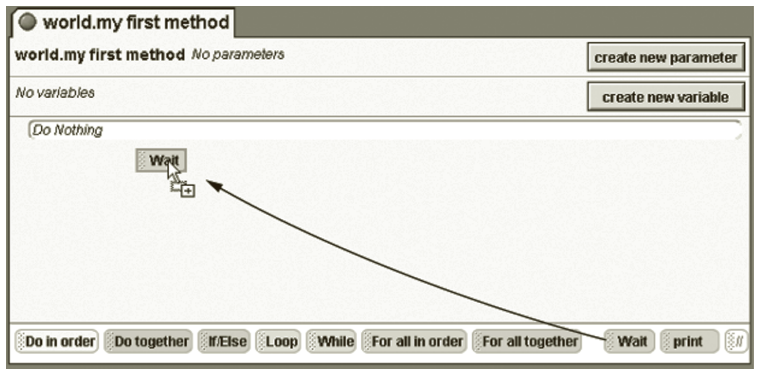
| Instruction | Description |
|---|---|
| `Do in order` | You place other instructions inside a `Do in order` instruction. The instructions that you place inside a `Do in order` instruction are executed in the order that they appear. |
| `Do together` | You place other instructions inside a `Do together` instruction. The instructions that you place inside a `Do together` instruction are executed simultaneously. |
| `If/Else` | The `If/Else` instruction tests a condition, which is anything that gives a true or false value. If the value is true, then one set of instructions is executed. If the value is false, then a different set of instructions is executed. |
| `Loop` | The `Loop` instruction causes one or more other instructions to repeat a specific number of times. |
| `While` | The `While` instruction causes one or more other instructions to repeat as long as a condition is true. |
| `For all in order` | The `For all in order` instruction steps through the items in a list, one item at a time, performing the same operation on each item. |
| `For all together` | The `For all together` instruction performs the same operation on all the items in a list simultaneously. |
| `Wait` | The `Wait` instruction causes the method to pause for a specified number of seconds. |
| `print` | The `print` instruction displays a message in a special area at the bottom of the *World Running…* window. |
| `//` | The `//` tile allows you to insert a comment into a method. |

In Alice you place instructions in a method by dragging tiles into the Method Editor. For example, if you want to place a `Wait` instruction in the method that you are currently writing, you simply click and drag the `Wait` tile into the Method Editor, as shown in Figure A-20. When you drop the tile (by releasing the mouse button) the `Wait` instruction will be created in the method.

In addition to using the instructions that you see at the bottom of the Method Editor, you can also create instructions that execute an object's primitive methods. Once you have added an object to a world, you can see tiles for all of the methods that the object can perform by doing the following:

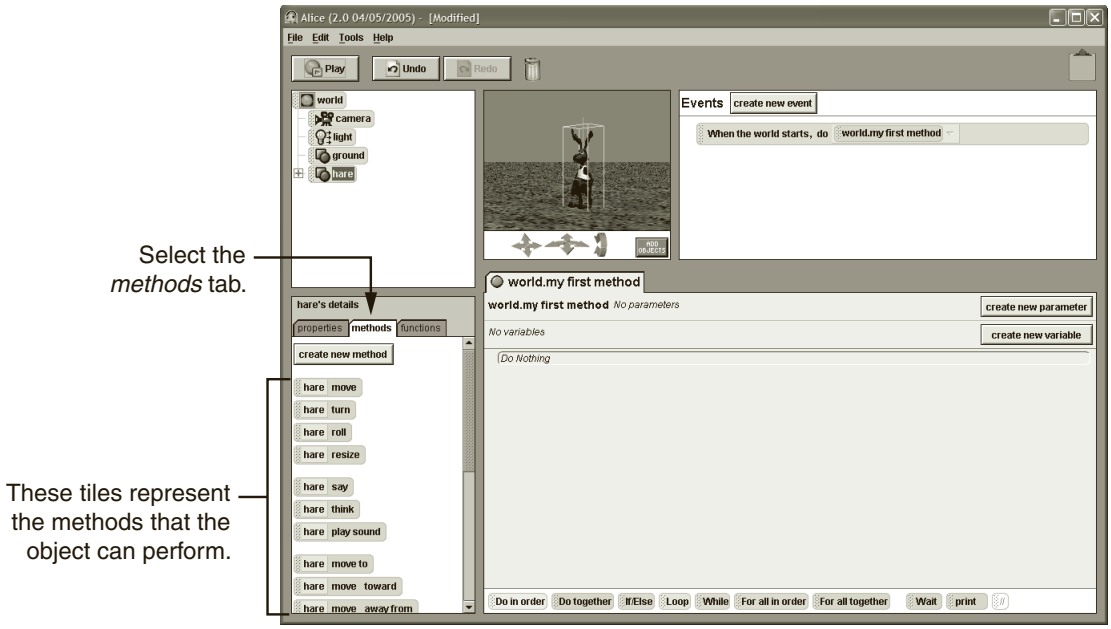
1. Select the object
2. In the Details Panel select the *methods* tab to display a set of tiles representing the object's methods

**Figure A-20** Dragging the `Wait` instruction into the Method Editor




For example, Figure A-21 shows an Alice world with an instance of the `Hare` class (which is in the *Animals* collection). The object, which is named `hare`, is selected. The *methods* tab is selected in the Details Panel, and a set of tiles for the `hare` object's primitive methods is displayed.

**Figure A-21** Methods displayed in the Details Panel




To create an instruction that executes a primitive method in the method that you are currently writing, simply drag the primitive method's tile and drop it into the Method Editor.

For example, Figure A-22 shows tile for the `hare` object's `move` method being dragged into the Method Editor.

Most of the primitive methods require that you specify arguments. For example, when you drop the tile for the `move` method into the Method Editor, a pop-up menu appears allowing you to select a direction. The allowable directions are up, down, left, right, forward, and backward. After you select a direction, another menu appears allowing you to select an amount, which is the distance that the object moves. In Alice distances are always measured in meters.

Figure A-23 shows an example of `world.my first method` after three instructions have been created. When the world containing this method is played, the `hare` object will move up 1 meter, then turn left 1 revolution, and then move down 1 meter.

**Figure A-22** Dragging the `hare.move` method tile into the Method Editor

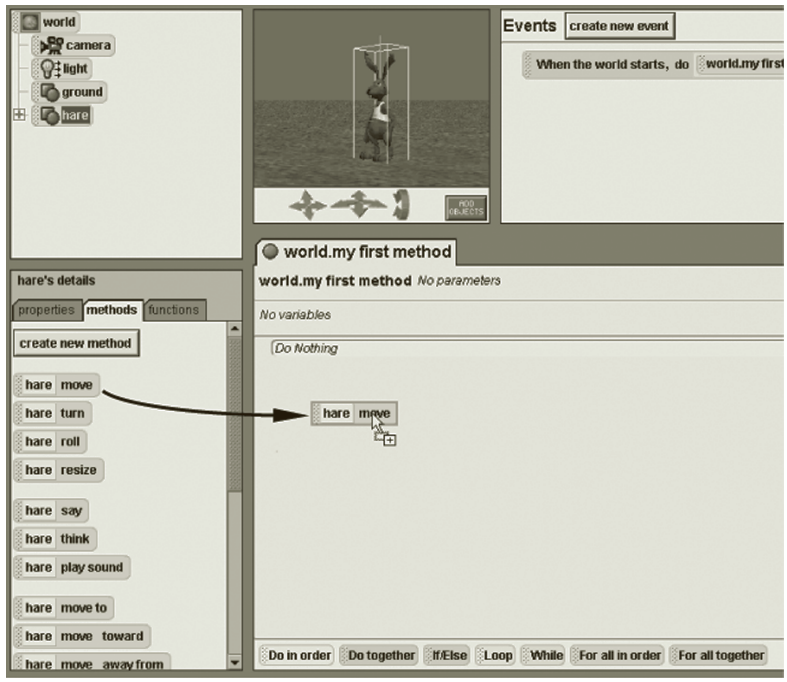
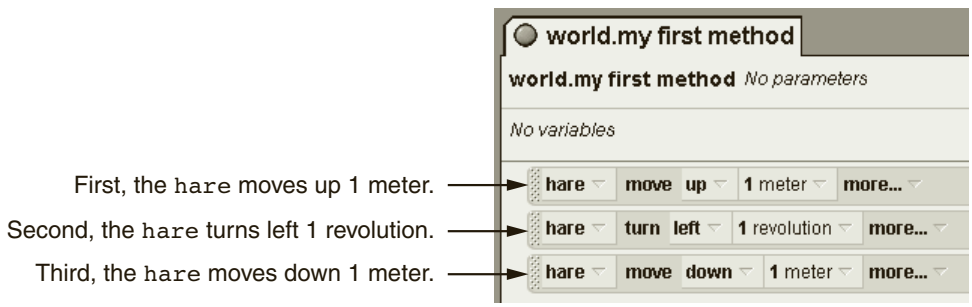



**Figure A-23** Three instruction tiles



## Copying and Deleting Instructions

To make a copy of an instruction tile within the same method, you right-click the tile and then select *make copy* from the menu that appears. To copy an instruction so that you can paste it into a different method, you drag the instruction to the clipboard. Then you open the method that you want to paste the instruction into, and click and drag the clipboard icon to the location where you want to paste the instruction. To delete an instruction tile that you have created in the Method Editor, you drag the tile to the trashcan.

## Creating Methods

When you first create an Alice world, a method named `world.my first method` is automatically created in the `world` object. You are not limited to this one method in the world, however. Follow these steps to create a new method in the `world`:

1. Select the `world` in the Object Tree.
2. In the Details Panel, under the *methods* tab, click the *create new method* button, as shown in Figure A-24.
3. A dialog box will appear asking for the new method's name. Enter a name in the dialog box and click the *OK* button. A tile for the new method will appear in the Details Panel, above the *create new method* button. For example, the Details Panel in Figure A-25 shows three world-level methods.
4. Create the instructions for the method in the Method Editor.

**Figure A-24** The *create new method* button

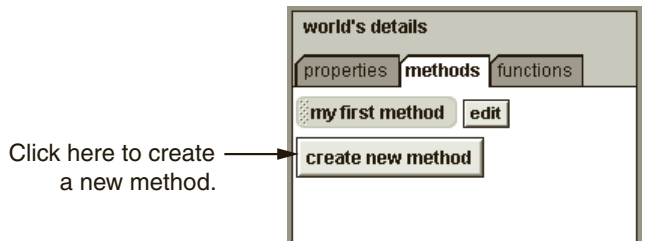
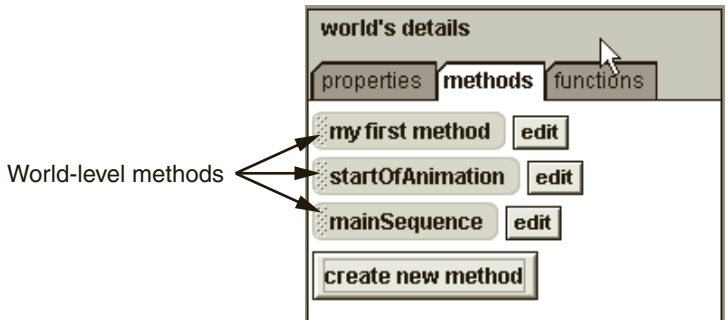



**Figure A-25** An example of a world with three world-level methods



Once you have created the new method, you can call it from other methods by dragging the new method's tile from the Details Panel into the Method Editor and dropping it at the point where you wish to call the method.

You can also create your own custom methods in the objects that you place in your world. In Alice the methods that are part of an object are referred to as *class-level methods*. If an object doesn't provide all of the methods that you need, you can easily add your own methods for that object. You write custom class-level methods in Alice by following these steps:

1. Create the desired object.
2. Select the object.
3. In the Details Panel, under the *methods* tab, click the *create new method* button.
4. A dialog box will appear asking for the new method's name. Enter a name in the dialog box and click the *OK* button. A tile for the new method will appear in the Details Panel, above the *create new method* button.
5. Create the instructions for the method in the Method Editor.

Once you have created the new method, you can call it from other methods in the usual way: by dragging the new method's tile into the Method Editor and dropping it at the point where you wish to call the method.

## Renaming Methods

To rename a method, you simply right-click the method's tile and select *Rename* from the menu that appears. After you do this, you will be able to edit the name that appears on the method's tile directly.

## Creating Variables and Parameters

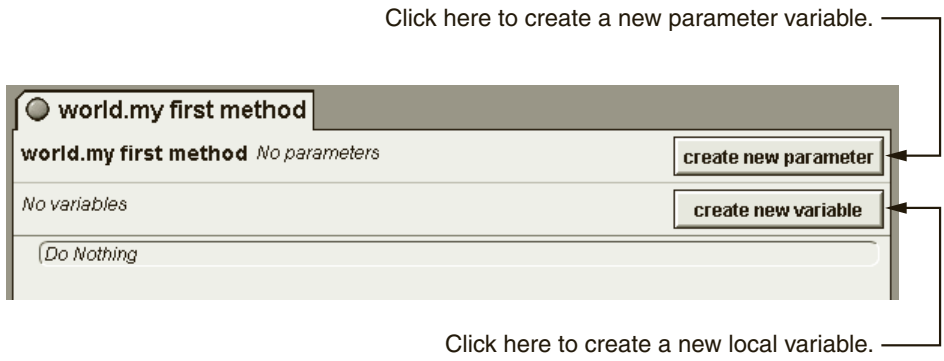
A variable is a storage location that is represented by a name. Like traditional programming languages, Alice allows you to use variables to store data. The following variable categories are available in Alice:

- **Local Variables**—A *local variable* belongs to a specific method and can be used only in the instructions in that method. When a method stops executing, its local variables cease to exist in memory.
- **World-Level Variables**—A *world-level variable* belongs to the `world` object, and exists as long as the world is playing.
- **Class-Level Variables**—A *class-level variable* belongs to a specific object, and exists as long as the object exists. Class-level variables are like properties.
- **Parameter Variables**—A *parameter variable* is used to hold an argument that is passed to a method when the method is called. Once you create a parameter variable in a method, you must provide an argument for that parameter whenever you call the method.

Before you can use a variable, you have to create it. To create a local variable or a parameter variable in a method, you open the method in the Method Editor and then you click

the *create new variable* button or the *create new parameter* button. Figure A-26 shows the locations of these buttons.

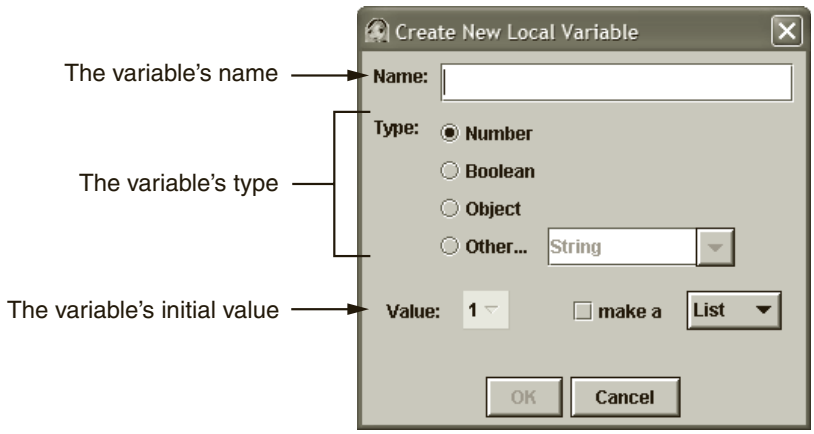
**Figure A-26** The *create new variable* button




When you click either of these buttons, a dialog box appears requiring you to enter more information about the variable. In the dialog box you enter the variable's name and select the variable's type and initial value. Figure A-27 shows the *Create New Local Variable* dialog box, which appears when you click the *create new variable* button. When you click the *create new parameter* button, a dialog box that is virtually identical to the one in Figure A-27 is displayed.

After you provide a name for the variable, select its type, specify its initial value, and click the *OK* button, a tile for the variable is created in the method.

**Figure A-27** The *Create New Local Variable* dialog box

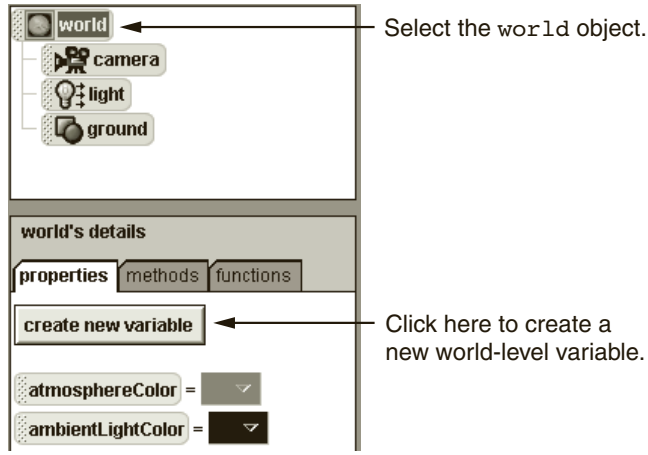



To create a world-level variable you perform the following steps:

1. Select the `world` object in the Object Tree.
2. In the Details Panel select the *properties* tab.

3. Click the *create new variable* button, which appears at the top of the *properties* tab, as shown in Figure A-28.
4. Enter the variable's name, type, and initial value in the *create new variable* dialog box, which is similar to the one shown in Figure A-27. When you click the dialog box's *OK* button, a tile for the variable will be created in the Details Panel, under the *properties* tab.

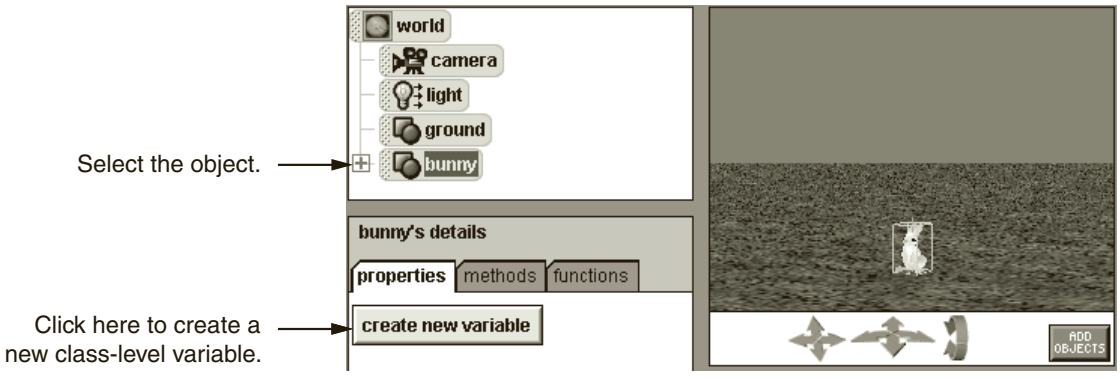
**Figure A-28** Creating a world-level variable




To create a class-level variable in an object you perform the following steps:

1. Select the object in the Object Tree.
2. In the Details Panel select the *properties* tab.
3. Click the *create new variable* button, which appears at the top of the *properties* tab, as shown in Figure A-29.
4. Enter the variable's name, type, and initial value in the *create new variable* dialog box, which is similar to the one shown in Figure A-27. When you click the dialog box's *OK* button, a tile for the variable will be created in the Details Panel, under the *properties* tab.

**Figure A-29** Creating a class-level variable



## Variable Assignment

When you create a variable, you give it an initial value. The initial value will remain in the variable until you store a different value in the variable. In an Alice method you can create *set instructions* that store different values in the variable. A set instruction simply "sets" a variable to a new value.

To create a set instruction for a variable, you drag the variable tile and drop it into the Method Editor at the point where you want the set instruction to occur. A menu appears, and you select *set value*. Another menu appears that allows you to specify the value you wish to store in the variable. As a result, a set instruction is created.

## Events

An event is an action that takes place while a program is running. When Alice worlds are running, they are capable of detecting several different types of events. For example, an event occurs when the user clicks an object with the mouse. An event also occurs when the user types a key on the keyboard. Table A-3 describes all of the events that an Alice world can detect while it is running.

**Table A-3** Events that Alice can detect

| Event | Description |
|---|---|
| `When the world starts` | This event occurs immediately when the world is started. It happens only once, each time the world is played. |
| `When a key is typed` | When the user types a key on the keyboard, this event occurs when the key is released. |
| `When the mouse is clicked on something` | This event occurs when the user clicks an object in the world with the mouse. |
| `While something is true` | When a condition that you have specified becomes true, this event occurs as long as the condition remains true. |
| `When a variable changes` | This event occurs when a variable's value changes. |
| `Let the mouse move <objects>` | This event allows the user to move an object in the world by clicking and dragging it with the mouse. |
| `Let the arrow keys move <subject>` | This event allows the user to move an object in the world by typing the arrow keys on the keyboard. |
| `Let the mouse move the camera` | This event allows the user to move the camera through the world by clicking and dragging the mouse. |
| `Let the mouse orient the camera` | This event allows the user to change the camera's orientation (the direction in which it is pointing) by clicking and dragging the mouse. |

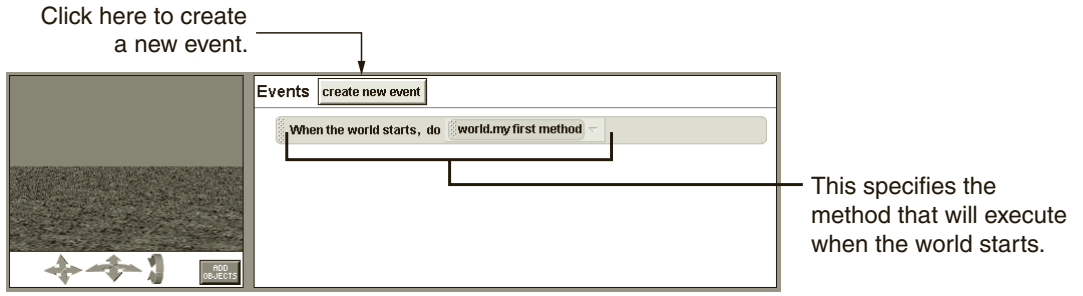
When any of the events listed in Table A-3 occur, your Alice world can perform an action in response to that event, such as calling a method.

At the top right of the screen in the Alice environment, you see an area labeled *Events*, as shown in Figure A-30. This area is called the *Events Editor*. When you create an Alice world, a tile appears in the Events Editor that reads as follows:

```
When the world starts, do world.my first method
```

This tile specifies that when the world starts, the method `world.my first method` will be executed. The left portion of the tile shows the name of an event, `When the world starts`, and the right portion of the tile is a drop-down box that shows the name of the method that will be executed when the event occurs. You can click the down arrow on the drop-down box to select a different method. Any method that is selected in this tile will be automatically executed when the world starts.

**Figure A-30** The Events Editor




The process of responding to an event is commonly called *handling the event*. In order for an Alice world to handle an event, a tile for that event must appear in the Events Editor. When a world is first created, the only tile that appears in the Events Editor is for the `When the world starts` event. If you want the world to handle any other events, you must create a new tile for the event in the Events Editor. To create a new event tile, you click the *create new event* button, as shown in Figure A-30. A menu of available events will appear next. You select the event that you want to handle from this menu. A tile for the event will then be created in the Events Editor.

Most event tiles require that you specify additional arguments, such as the method that you want to execute in response to the event. A method that is executed in response to an event is commonly referred to as an *event handler*. For example, the event tile that is shown in Figure A-30 specifies that when the world starts, `world.my first method` is called. The method `world.my first method` is the event handler.

Figure A-31 shows another example of an event tile. Assume that this tile appears in a world that has an object named `fridge` (a refrigerator object). The event tile specifies that when the mouse is clicked on the `fridge` object's `fridgeDoor` subpart, the `fridgeDoor` will turn left 0.25 revolutions.

**Figure A-31** Example of an event tile