Alice 2.0

Exploring Alice and Object-Oriented Programming

PROJECT 1

CASE PERSPECTIVE

Dr. Carole Dodgson needs your help. She would like to use a sample virtual world to demonstrate some fundamental concepts of computer programming to students in her Information Technology course.

A virtual world is one that exists only in the memory of a computer, not in our real, physical world. Some virtual worlds have no user interaction and are similar to films or animated cartoons, while others are more like interactive video games with many mouse and keyboard controls.

Your task is to use Alice 2.0, a new easy-to-use programming system, to develop a sample virtual world for demonstrations in Dr. Dodgson's course. The new virtual world will contain an ice skater on a frozen lake. The overall project has been broken down into four smaller projects in which you will build the sample virtual world, then progressively add more sophisticated features to the world. When you have finished the fourth project, you will have a complete virtual world with programs to animate the ice skater and to let a user control what the skater does. Dr. Dodgson can then demonstrate concepts of computer programming in her course using your ice skater world.

Alice 2.0

Exploring Alice and Object-Oriented Programming

PROJECT 1

Objectives

You will have mastered the material in this project when you can:

- Describe the concept of object-oriented programming (OOP)
- Start Alice
- Locate and describe the components of the Alice interface
- Load and play an existing Alice world
- Create and save a new Alice world
- Print the code for Alice methods and events

What Is Alice?

Alice is a programming environment that allows users with little or no computer experience to program characters and objects in a virtual world, much like a modern animated film or video game. Like the real world, the virtual world of Alice has three-dimensional space (and time), and each object has properties just like physical objects, such as color, size, location, and so on. Alice has a camera that allows you to see its virtual world on a computer screen, so you can see what is happening in your virtual world as your programs run. Figure 1-1 shows an Alice virtual world.

A **computer program** is a step-by-step set of instructions telling a computer how to perform a specific task. Learning to write computer programs can be difficult because of two major problems faced by novice programmers: language and visualization. The language problem occurs when people trying to learn about programming concepts must also learn a new programming language at the same time. The visualization problem occurs when people have difficulty trying to visualize what will happen inside a computer when a program runs. Computer scientists and educators at Carnegie Mellon University and the University of Virginia have developed Alice to make programming easier to learn by minimizing the problems of language and visualization.

The graphic nature of Alice solves the visualization problem described above. For example, if you tried to program the skater in Figure 1-1 to skate around in a circle, and instead she simply stayed in one spot and spun around, you could see that happening on the screen, and correct your program accordingly.

The instructions in the language of Alice are contained on tiles, which you can drag and drop into place to write new programs. You can see the Alice instruction tiles in Figure 1-2, which shows a program under construction. Creating a program in this manner allows you to focus on the concepts of computer programming, especially the logic of your programs, instead of having to worry about the particular details of a new language, such as spelling and punctuation. This helps to solve the language problem faced by students who have traditionally had to create programs like that shown in Figure 1-3, which is from an introductory Java programming course.

FIGURE 1-1

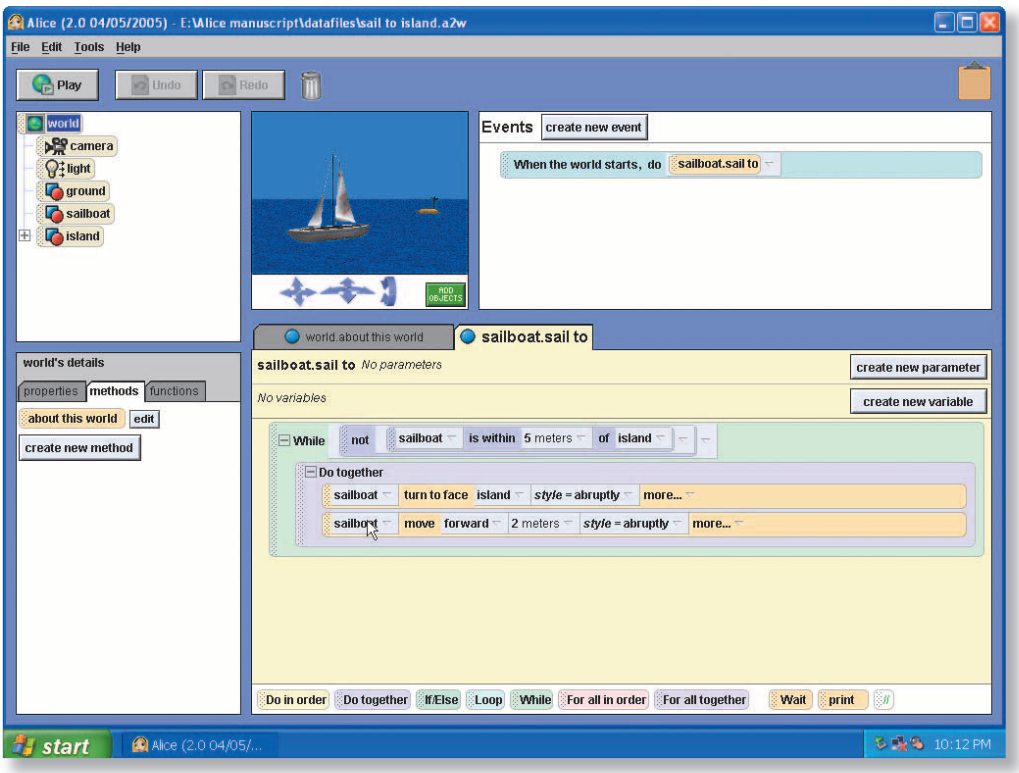

FIGURE 1-2

```
/*      This program demonstrates simple math.
        It takes user input and displays it.  */

import javax.swing.JOptionPane;
public class SimpleMath
{
        public static void main (String arg[] )
        {

                // declare variables
                double integer_one, integer_two;
                String number_one, number_two, sum;

                //get first of two integers
                number_one = JOptionPane.showInputDialog (null, "Please enter first number: ");
                integer_one = Integer.parseInt (number_one);

                number_two = JOptionPane.showInputDialog(null, "Please enter the second number");
                integer_two = Integer.parseInt (number_two);

                JOptionPane.showMessageDialog (null, number_one, "First Number", JOptionPane.OK_CANCEL_OPTION );
                JOptionPane.showMessageDialog (null, number_two, "Second Number", JOptionPane.OK_CANCEL_OPTION );

                JOptionPane.showMessageDialog (null, "Sum equals " + (integer_one + integer_two), "Sum of the numbers",
JOptionPane.PLAIN_MESSAGE);
                JOptionPane.showMessageDialog (null, "Difference equals " + (integer_one - integer_two), "Difference of two
Numbers", JOptionPane.PLAIN_MESSAGE);
                JOptionPane.showMessageDialog (null, "Product equals " + (integer_one * integer_two), "Integer Product",
JOptionPane.PLAIN_MESSAGE);
                JOptionPane.showMessageDialog(null, "Quotient equals " + (integer_one / integer_two), "Integer Division",
JOptionPane.OK_CANCEL_OPTION);

                System.exit(0);
        }

}
```

FIGURE 1-3

Although you will be working with Alice throughout this book, your ultimate goal is not to learn about Alice itself, but rather to use Alice to learn some of the fundamental concepts of modern computer programming. In many ways, Alice is very similar to most programming languages in use today, such as C#, C++, Java, and Visual Basic. It requires you to plan your programs ahead of time and work with the logical structures of programming, just as you must do with other languages. It is both object-oriented and event-driven, so that you can learn about these two important aspects of modern programming as well.

You may be surprised to discover how easy it is to learn computer programming once you get past the two big hurdles of language and visualization. In some ways, programming is like telling a story, except that you are describing what *will* happen instead of what has already happened. In other ways, it is like creating your own video game, as you build controls for the user and organize programs so that one action depends on another. You will also discover that working with Alice can be fun and interesting, which never hurts when you are trying to learn something new.

More About

Virtual Worlds

Alice is one of the programs used in the Building Virtual Worlds course for third-year students offered by the Entertainment Technology Center at Carnegie Mellon University. For more information on the course and for a look at some of the worlds created by students, visit www.etc.cmu.edu/curriculum/bvw/.

Project One — Creating an Alice World

This project will introduce you to the Alice interface and some basic ideas about object-oriented programming. You will explore the Alice interface, load and play an existing Alice world, and then create a short Alice world of your own, in which an ice skater will perform a few short moves and then say "Hello, World!" as shown in Figure 1-4. Discussions of object-oriented programming are included within the project.

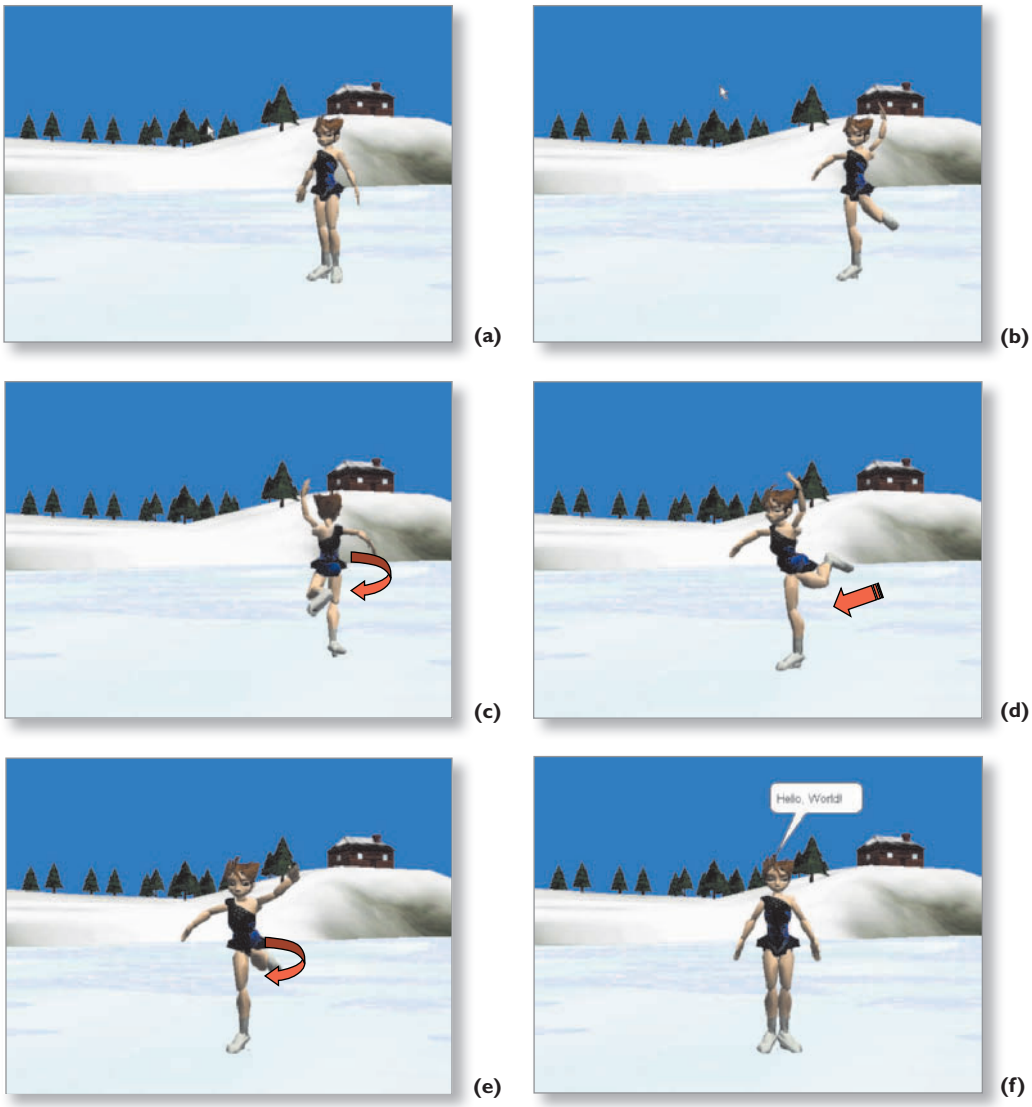

FIGURE 1-4

Objects and Object-Oriented Programming

An object is anything that can be represented by data in the memory of a computer and manipulated by a computer program. An object can be a tangible item in the physical world or just an abstract idea. For example, most large airplanes have a computer, called an autopilot, that can fly the plane. The autopilot is a computer that manipulates an object in the physical world.

Most objects that computers manipulate, however, are not physical objects. Payroll information, student records, bank accounts, and the characters in a video game are all examples of objects that are not physical objects, but simply concepts or ideas represented in the memory of the computer. Whether an object actually exists in the physical world does not matter much in terms of what happens inside a computer. To a computer, an object is simply something that can be represented by data in the computer's memory and manipulated by computer programs.

The data that represents an object is organized into a set of **properties**. Each property describes the object in some way. For example, the weight of an airplane, its location, and the direction in which the plane is facing are all properties of the airplane. A computer manipulates an object by changing some of its properties or some

of the properties of the object's subparts. Sometimes the hardware in a computer can translate these changes in properties into actions that affect real objects in the physical world. For example, the autopilot might change the angle of a wing flap, which is a subpart of the airplane, and that, in turn, affects the entire airplane. Both the representation of the airplane in the computer's memory and the real airplane in the physical world are affected by what the computer does. In other cases, the changes to an object's properties only affect information in the computer's memory and have no other effect on the physical world. When a student's grade is recorded on a computer, the final grade property of the student's computerized academic record is changed, but there is no other direct effect on the physical world.

The programs that manipulate the properties of an object are called the object's **methods**. You can think of an object as a collection of properties and the methods that are used to manipulate those properties. The values stored in the properties of the object at any one time are called the **state** of the object.

A **class of objects** is a set of similar objects that each have the same properties and methods. They usually do not have the same values stored in their properties, but they have the same properties. For example, two objects from the tree class of object will have all of the same properties—such as height, age, and trunk diameter—but one may have a different value stored in its height property, which means that it will be taller or shorter than another tree from the same class. Each individual object is called an **instance** of a class, and the process of adding an individual object to a world is called **instantiation**.

Most modern computer programming languages are object-oriented languages, in which programs are organized into a set of methods that manipulate the properties of objects stored in a computer. This modern approach to computer programming is known as **object-oriented programming**, or **OOP** for short.

Alice is an object-oriented system of programming. The objects in Alice exist in a three-dimensional virtual world, much like a modern video game. In fact, the virtual world itself is an object in Alice; it has properties, and methods that can be used to manipulate those properties. As you continue with this exploration of the Alice interface, you will see the properties and methods of some of the objects in the lakeSkater world.

More About

Clicking and Dragging

Many Alice tiles contain white parameter boxes, such as the **1 second** box in the **Wait 1 second** instruction tile. When clicking and dragging such tiles you should be careful to click an open background area on the tile, not a parameter box.

More About

Slow and Steady Alice

You may notice that there is a Slow and Steady Alice icon in addition to the standard Alice icon. The standard version of Alice uses video hardware for screen graphics. On computers with older video cards, especially older notebook computers, this may not work well. The Slow and Steady version of Alice uses software video acceleration to avoid this problem, but runs a bit slower than the standard version of Alice. More information about different versions of Alice for various computers, including hardware requirements, is available on the Web at: www.alice.org/downloads/authoringtool/.

Starting Alice

Before starting, you should have a computer system with Alice properly installed. The software is available free of charge from Carnegie Mellon University via the Alice Web site at www.alice.org. Alice has been designed so that it is easy to install, and so that it will not interfere with other software on your computer system. It does use a lot of memory, though, so generally it is not a good idea to run Alice and other programs at the same time. The following steps show how to start Alice.

To Start Alice

1

- **Navigate to the location where the Alice.exe program has been installed on your computer, or where a shortcut to the Alice.exe program has been placed. Often this will be on the desktop as in Figure 1-5, but your instructor or systems administrator may direct you to a different location.**

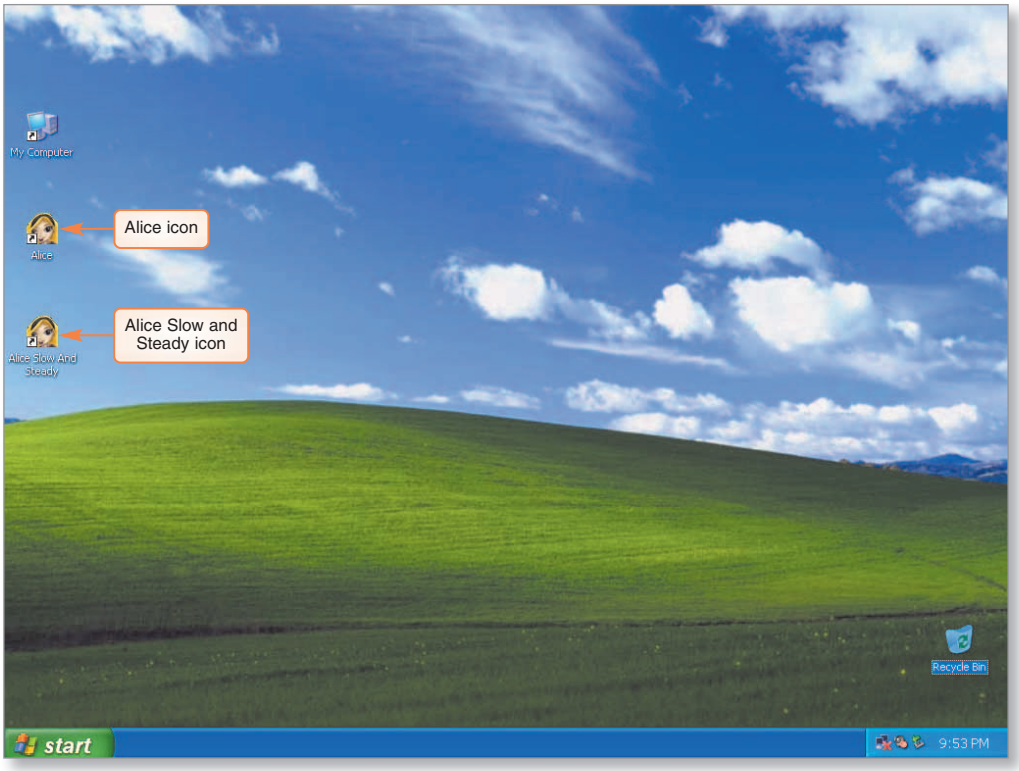

FIGURE 1-5

2

- **Double-click the** Alice **icon. If file extensions are visible on your system it will be named Alice.exe.**

The Alice title box appears while Alice is loading, as shown in Figure 1-6. Note that Alice is a large program, so it may take up to a minute to load.

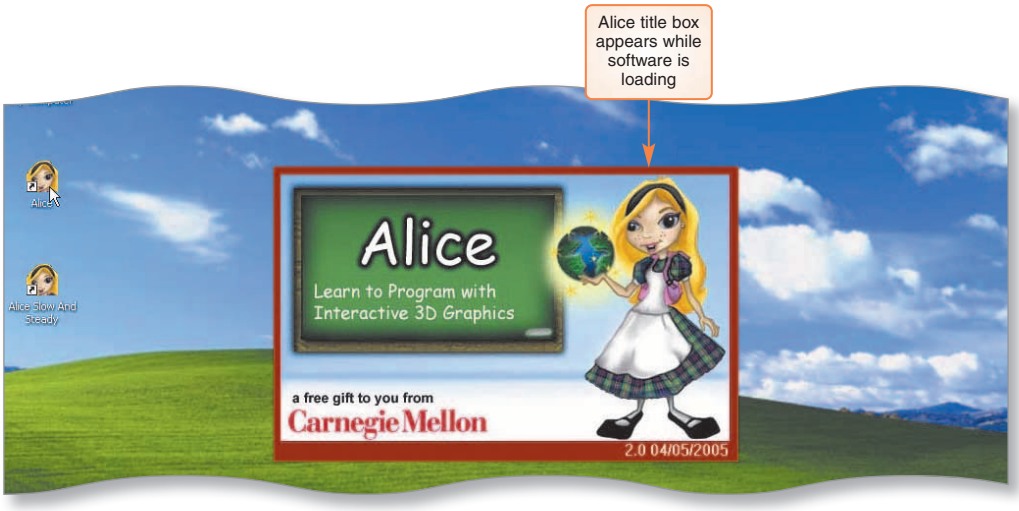

FIGURE 1-6

The Welcome to Alice! dialog box appears (Figure 1-7). You may return to this dialog box at any time while using Alice by clicking File on the menu bar, and then clicking New World or Open World.

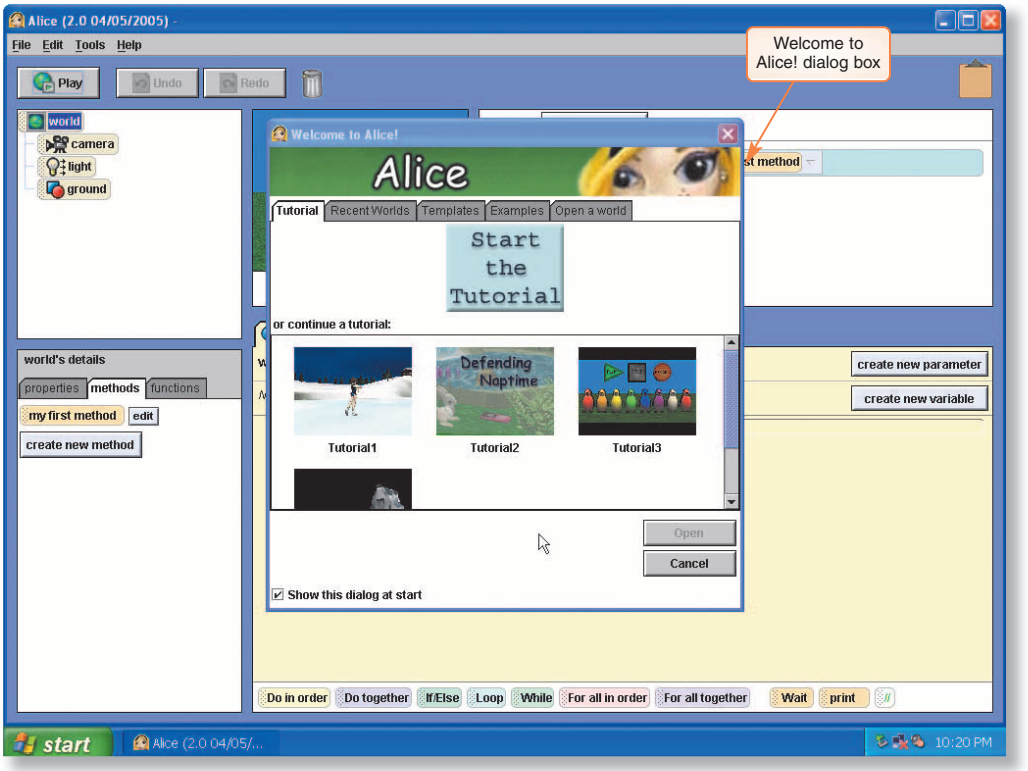

FIGURE 1-7

The Alice Interface

Users interact with Alice through the Alice Integrated Development Environment (IDE). An **IDE** is a computer program that is used to facilitate the writing of other computer programs. Specialized IDEs exist for most modern programming languages and often contain everything programmers need to create and run computer programs. A typical IDE may contain a text editor for entering code, a debugger for finding and fixing errors, and an object library to store modules of prewritten code. Alice contains similar features, but its IDE is simpler than most. The Alice IDE is often called the **Alice interface**.

The Welcome to Alice! Dialog Box

The first item the user sees when the Alice interface opens is the Welcome to Alice! dialog box (Figure 1-8), which contains several tabs: Tutorial, Recent Worlds, Templates, Examples, and Open a world. The following section describes the items in this dialog box:

- Tutorial: You will not use the tutorials now, but you may want to come back to them later as an exercise on your own. When you are ready to use the tutorials, either click the tutorial you would like to run, or click the large Start the Tutorial button to follow them in order. They were created by the developers of Alice to help people learn the system and are quite easy to follow.
- Recent Worlds: The Recent Worlds tab contains thumbnail sketches of the most recently saved Alice worlds.

- Templates: Alice has six blank templates for starting a new virtual world: dirt, grass, sand, snow, space, and water. Each template includes a texture for the ground and a background color for the sky.
- Examples: Several example worlds are included with Alice. This tab is used later in the project to run one of the example worlds.
- Open a world: This tab is used to open other Alice worlds saved on your computer and is similar to the Open dialog boxes seen in other programs, such as Microsoft Word. Navigation icons appear across the top, a list of folders and Alice worlds from the current folder appear in the middle, and some controls to view and open files appear at the bottom of the dialog box.

In the following steps, you explore the Welcome to Alice! dialog box.

Q&A

Q: How can I prevent the Welcome to Alice! dialog box from being displayed?

A: Click the **Show this dialog at start** check box in the Welcome to Alice! dialog box to deselect it. After this, Alice will open to a new world with a blank grass template instead of the Welcome to Alice! window.

To Explore the Welcome to Alice! Dialog Box

1

• **Click the** Tutorial **tab (if it is not already selected).**

Four Alice tutorials appear, as shown in Figure 1-8.

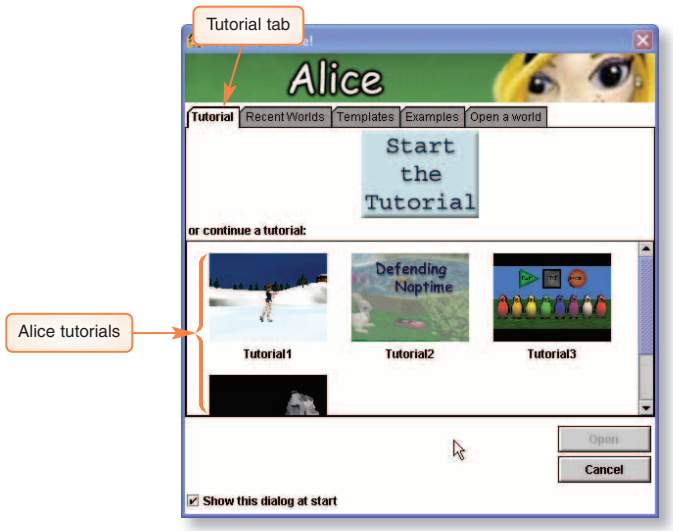

FIGURE 1-8

2

• **Click the** Recent Worlds **tab.**

A typical Recent Worlds tab is shown in Figure 1-9, but if no worlds have been saved since Alice was installed on your system, the message "No recent worlds." appears.

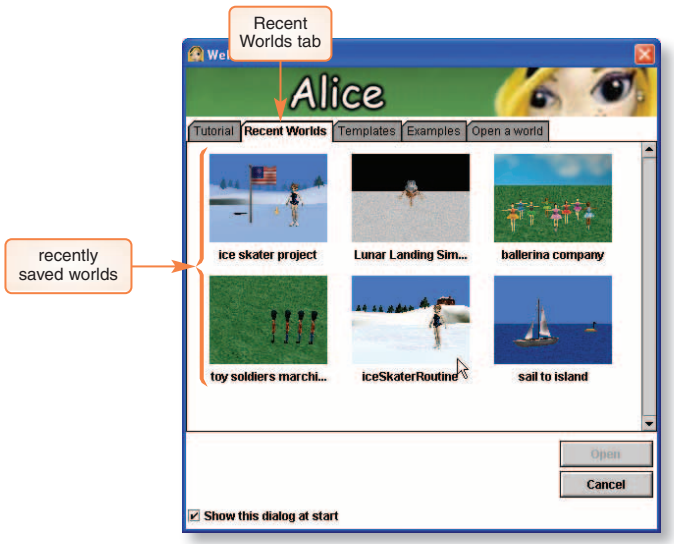

FIGURE 1-9

3

- **Click the** Templates **tab.**

The six default templates are displayed, as shown in Figure 1-10.

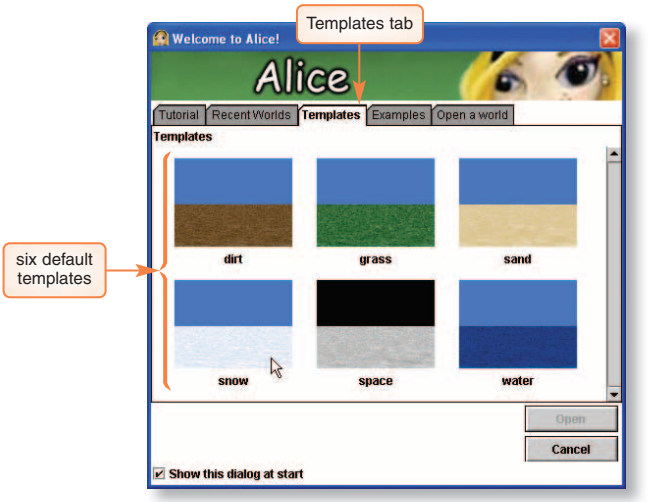

FIGURE 1-10

4

- **Click the** Examples **tab.**

Thumbnail sketches of example worlds are visible on the Examples tab, as shown in Figure 1-11.

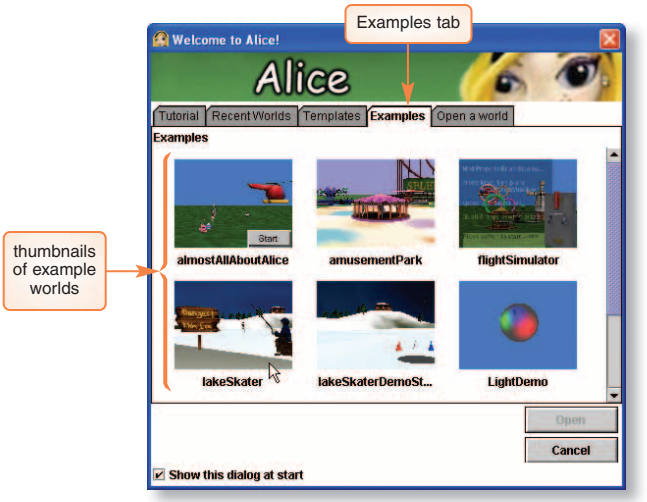

FIGURE 1-11

5

- **Click the** Open a world **tab.**

The Open a world tab appears (Figure 1-12). Notice that Alice world files end with the extension .a2w. These files were created with version 2.0 of Alice, the most recent version.

FIGURE 1-12

While you are viewing or editing an Alice world, a save warning window will appear every 15 minutes, warning you that you have not saved your Alice world (Figure 1-13). If this happens while you are simply viewing an Alice world, such as in this section of the project, then it is probably safe to ignore the warning. If it happens while you are creating or editing your own Alice world, then it is a good idea to save your world.

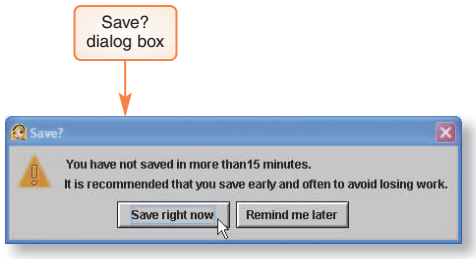

FIGURE 1-13

Q&A

Q: How can I change the size of the areas of the Alice interface?

A: Click the blue background in the space between the areas of the interface and then drag the mouse pointer left, right, up, or down as appropriate to resize the areas of the Alice interface.

The Main Work Areas of the Alice Interface

The Alice interface has five main work areas: the World window, the Object tree, the Details area, the Editor area, and the Events area. In the following set of steps, you will open and manipulate a sample world in order to explore and become familiar with the Alice interface.

To Open the lakeSkater World and Explore the Alice Interface

1

- **Click the** Examples **tab in the Welcome to Alice! dialog box.**

The Examples tab is displayed (Figure 1-14).

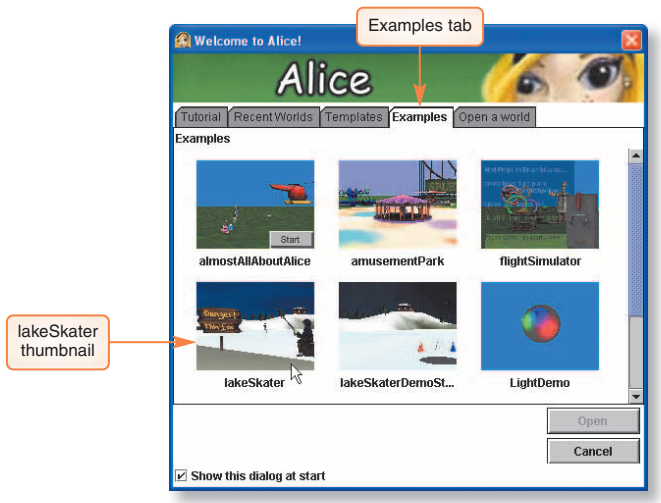

FIGURE 1-14

2

• **Click the** lakeSkater **thumbnail, and then click the** Open **button to open the** lakeSkater **Alice world.**

It takes a few seconds for Alice to load all of the elements of the world. The names of the elements flash past in a small window in the center of the screen while this happens. When Alice is finished loading, the standard interface will appear as shown in Figure 1-15.

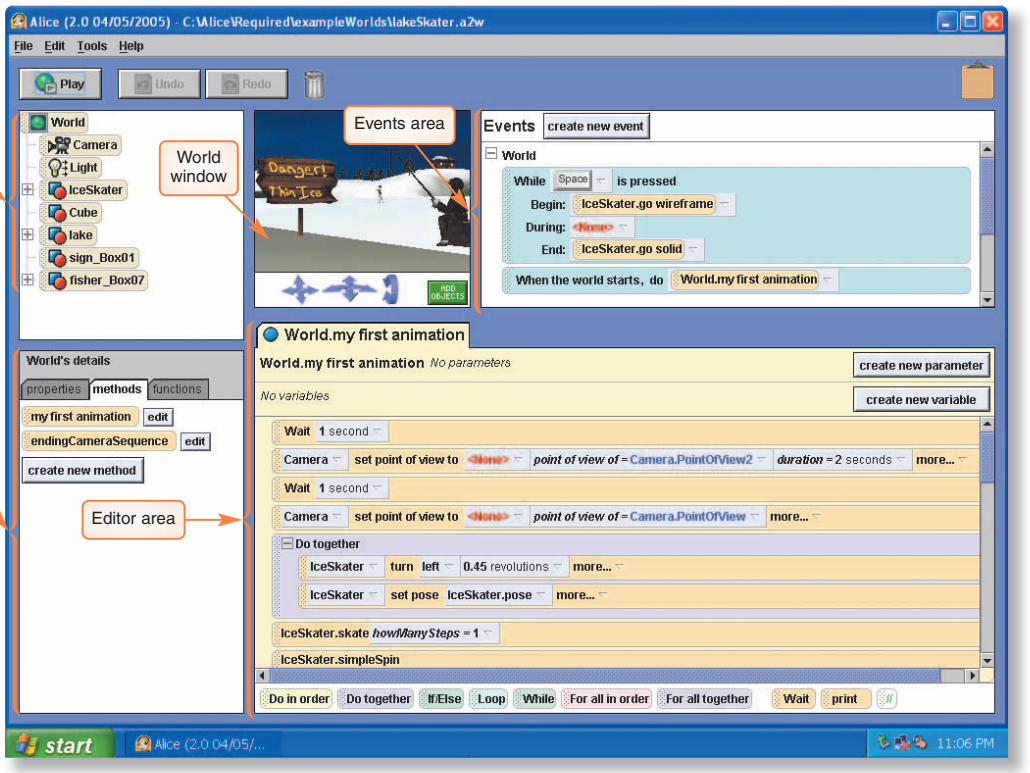

FIGURE 1-15

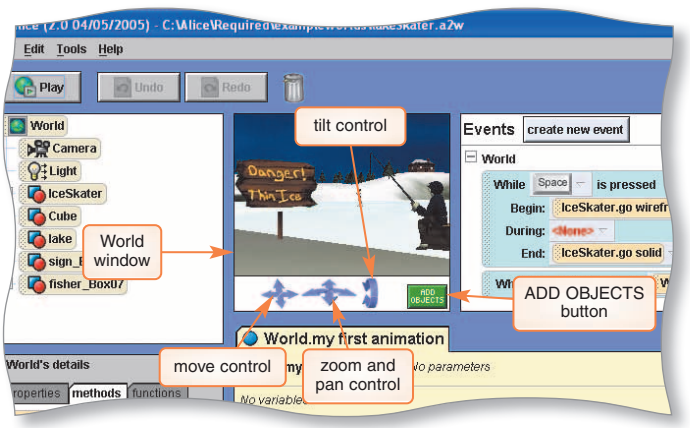

FIGURE 1-16

The World Window

The **World window** contains a view of the current Alice world, as shown in Figure 1-16.

You can pan, tilt, zoom, and move the Alice camera using the blue arrows below the World window. The curved arrow on the right is the camera's **tilt control**. It is used to tilt the camera up or down, similar to the way that you might tilt your head up or down. The center control is a mixed control, to **zoom** and **pan** the camera. A camera can zoom in and zoom out, and pan left and pan right. Zooming in means the camera is moved in closer to get a tighter shot of an item, so that it fills more of the screen. Zooming out means the camera is moved out further to get a longer shot of an item, so that it becomes smaller on the screen. Panning means to turn the camera left or right without moving the position of the camera, although it is possible that you could pan and move at the same time. The left set of arrows at the bottom of the world window is the **move control**, which provides controls to move the camera left and right, and up and down.

The large green ADD OBJECTS button, located to the right of the camera control arrows, switches the interface from the standard mode to Scene Editor mode, which is used to add and position objects in an Alice world before playing the world. You will work with Scene Editor mode later in this project.

More About

Zoom Controls

Most cameras have a lens that will allow the photographer to zoom in and zoom out without moving the camera. In Alice, you zoom in and zoom out by actually moving the camera forward and backward.

The Object Tree

The **Object tree** is to the left of the World window. It shows the objects in the current Alice world organized as a tree of **tiles**, with a tile for each object (Figure 1-17). Four object tiles — World, Camera, Light, and ground — appear in every new Alice world, although the creator of this lakeSkater world has deleted the ground object from the world. The plus sign next to an object shows that it has subparts, which may be seen by clicking the plus sign.

In the steps on the next page, you view the subparts of the IceSkater object.

FIGURE 1-17

To View the Subparts of the IceSkater Object

1

• **Click the plus sign next to the IceSkater tile in the Object tree.**

Tiles for each subpart of the IceSkater, such as left and right thighs (ThighL and ThighR), are displayed (Figure 1-18). These parts are actually objects themselves, which together compose the IceSkater object. Notice that some of these parts have plus signs, indicating that they also have subparts.

2

• **Click the minus sign next to the IceSkater tile.**

The IceSkater tile's subparts are hidden.

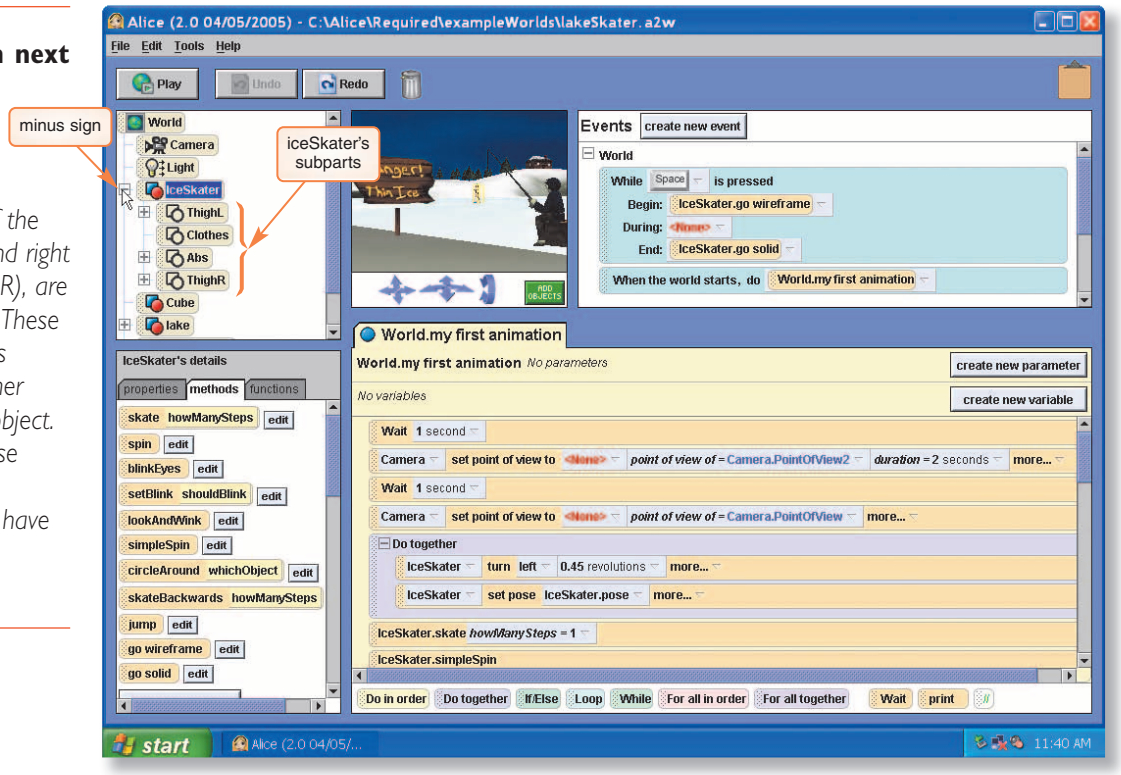

FIGURE 1-18

The Details Area

The **Details area** of the Alice interface is located below the Object tree. It has tabs to show properties, methods, and functions for an Alice object. Figure 1-19 shows the methods tab in the Details area for the IceSkater object. As described above, properties contain information about an object, such as its color and position in the world, and methods are programs that manipulate an object, such as the set of instructions used to make an **IceSkater** turn. A **function** is simply a method that computes and returns a value, such as the distance between two objects. The values that functions return can

sometimes be used like properties, but while a property is a stored value that describes an object, the value returned by a function must be calculated by the computer.

Information about the currently selected object is displayed in the Details area. You may select an object by clicking that object in the World window or by clicking its tile in the Object tree. When you do so, that object's properties, methods, and functions will appear on the tabs in the Details area. In the following steps, you explore the Details area to view the details of an object.

To View the Details of an Object

1

• **Click the** IceSkater **tile in the Object tree. If necessary, click the** methods **tab to display the** IceSkater **object's methods.**

The IceSkater's methods are displayed in the Details area (Figure 1-19).

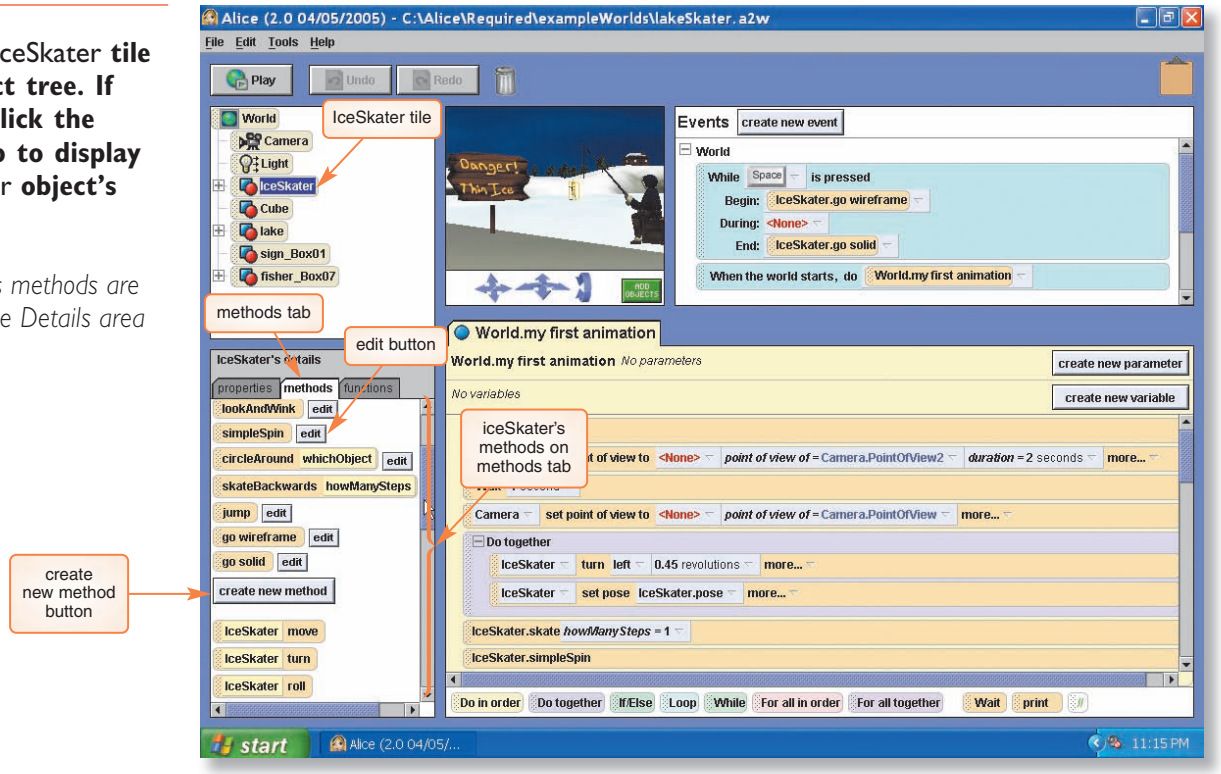

FIGURE 1-19

2

• **Click the** properties **tab.**

The IceSkater's properties tab is displayed in the Details area (Figure 1-20).

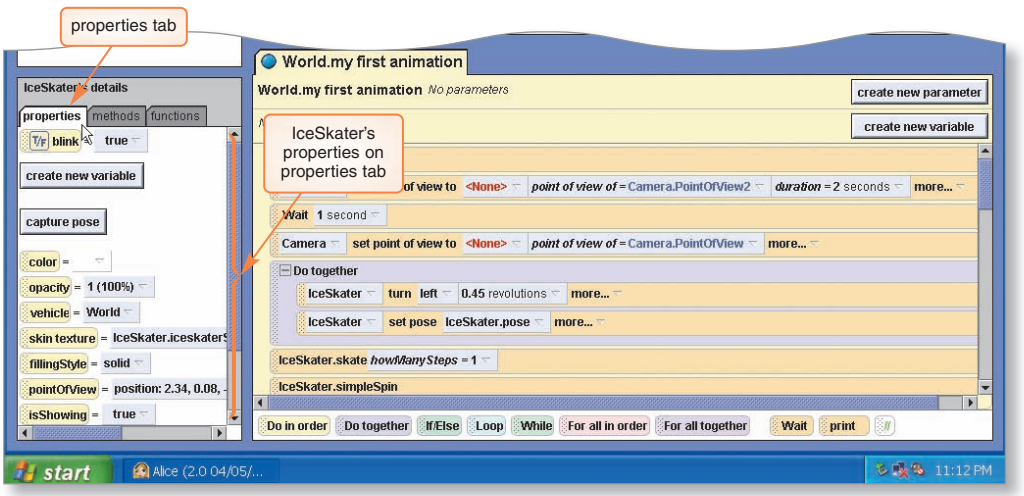

FIGURE 1-20

3

- **Click the functions tab.**

The IceSkater's functions are displayed in the Details area (Figure 1-21).

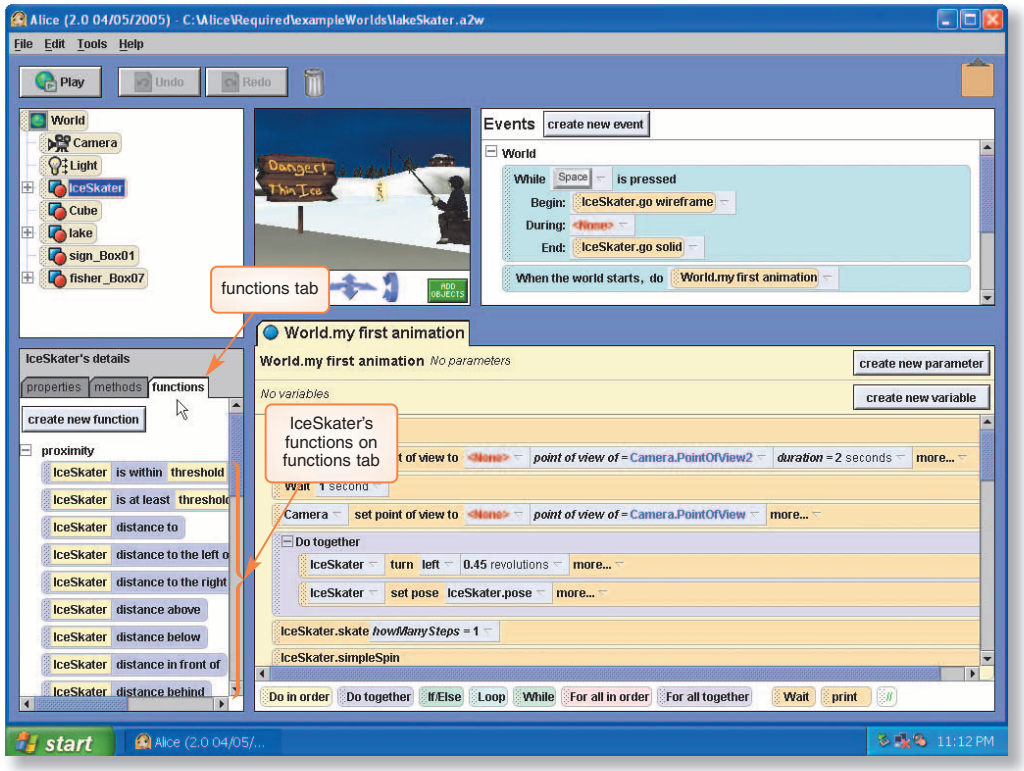

FIGURE 1-21

Even though a function is a method, functions are placed on a separate tab in the Details area because of the special nature of functions. None of the objects in this lakeSkater sample world has any user-created functions; they are all primitive functions, similar to the primitive methods described in the next section.

Primitive and User-Defined Methods

In Figure 1-19 on page AL 14, you can see a create new method button. Note that an edit button appears next to the method tiles above this button, and there is no edit button next to the method tiles below the button. The tiles that have an edit button are for user-created methods; the tiles that do not have an edit button are for primitive methods. **Primitive methods** provide basic behaviors for objects, such as turning and moving, and are examples of encapsulated methods. The details of an encapsulated method are hidden from the user. **Encapsulated methods** can be used within other programs, but they cannot be changed or edited because their details are hidden. Encapsulation is an important part of object-oriented programming.

Later in this project, you will begin working with methods. For now, it is enough to see where methods are located on the Alice interface. In the following steps, you view the details of a user-created method.

To View the Details of a Method

1

• **Click the** edit **button next to the** spin **tile on the** methods **tab.**

The details of the spin method appear in the Editor area, as shown in Figure 1-22. This is an example of a user-created method that is not encapsulated.

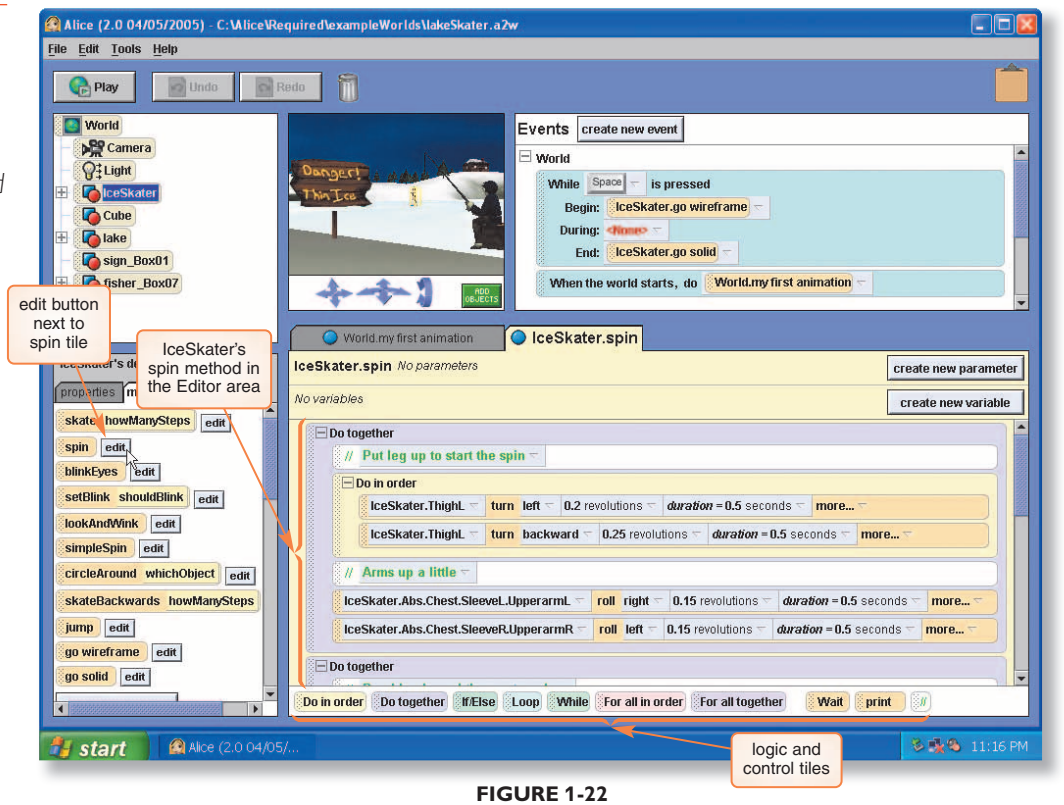

FIGURE 1-22

The Editor Area

The largest area of the Alice interface is the **Editor area**, which is located below the World window, as shown in Figure 1-22 above. Methods are assembled here by clicking and dragging tiles from other parts of the interface. The bottom of the Editor area has a row of logic and control tiles that can be used to put branching, looping, and other logical structures into Alice methods. Later in this project, you will build your first method. Project 3 contains more information about logic in Alice methods and using the logic and control tiles.

The Events Area

The **Events area** in Alice is located above the Editor area, as shown in Figure 1-23. This part of the interface shows existing events and is used to create new events. An **event** consists of a condition, called an event trigger, and the name of a method, called an event handler. The event handler runs whenever the event trigger occurs. For example, you could create an event to make the ice skater spin whenever the user presses the s key. Pressing the s key would be the event trigger, and the method that makes the ice skater spin would be the event handler.

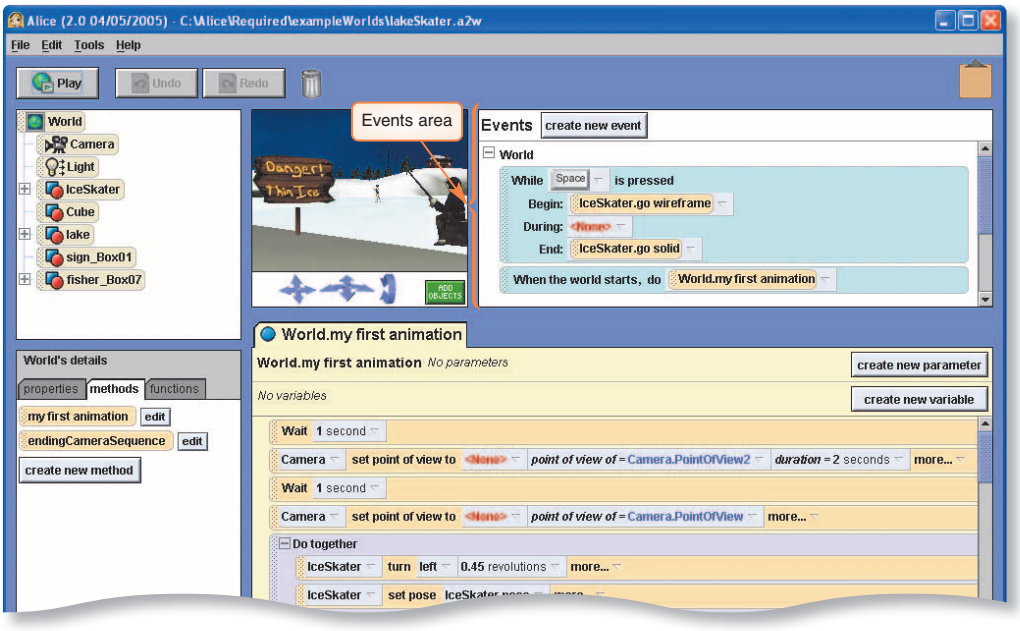

FIGURE 1-23

Software with events is called **event-driven software**. Operating systems that allow the user to manipulate the computer by using the mouse and keyboard shortcuts, such as Microsoft Windows, contain event-driven software. Events are covered in detail in Project 4.

Other Elements of the Alice Interface

In addition to the main work areas that you have just explored, the Alice interface has two icons, three buttons, and a menu bar near the top of the screen, as shown in Figure 1-24. These are discussed in the following sections.

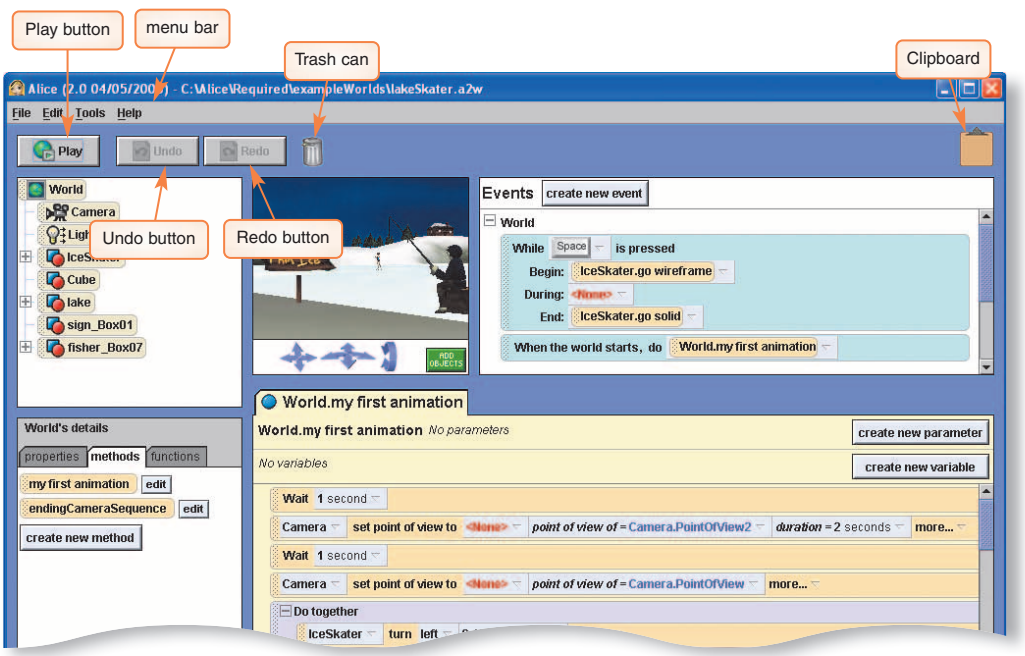

FIGURE 1-24

Q&A

Q: How can I tell where an item that is being dragged may be dropped?

A: The color of the border around an Alice tile, and around other elements of the interface, will indicate where an object that is being dragged may be dropped. When you begin to drag an object, yellow borders will appear around places where the object may be dropped. When an object is correctly in place over a location where it could be dropped, then the border of the target location and of the item being dragged will both turn green. A red border indicates that the item being dragged cannot be dropped in its current location.

The Clipboard

The **Clipboard** icon is used for copying instruction tiles in Alice methods. You can copy a tile by dragging and dropping it on the Clipboard icon in the upper-right corner of the interface, and then dragging it from the Clipboard icon and dropping it in its new location. The following steps illustrate copying an instruction tile.

To Copy an Instruction Tile

1

• **Click the** World.my first animation **tab in the Editor area.**

The instructions in World.my first animation *appear in the Editor area (Figure 1-25).*

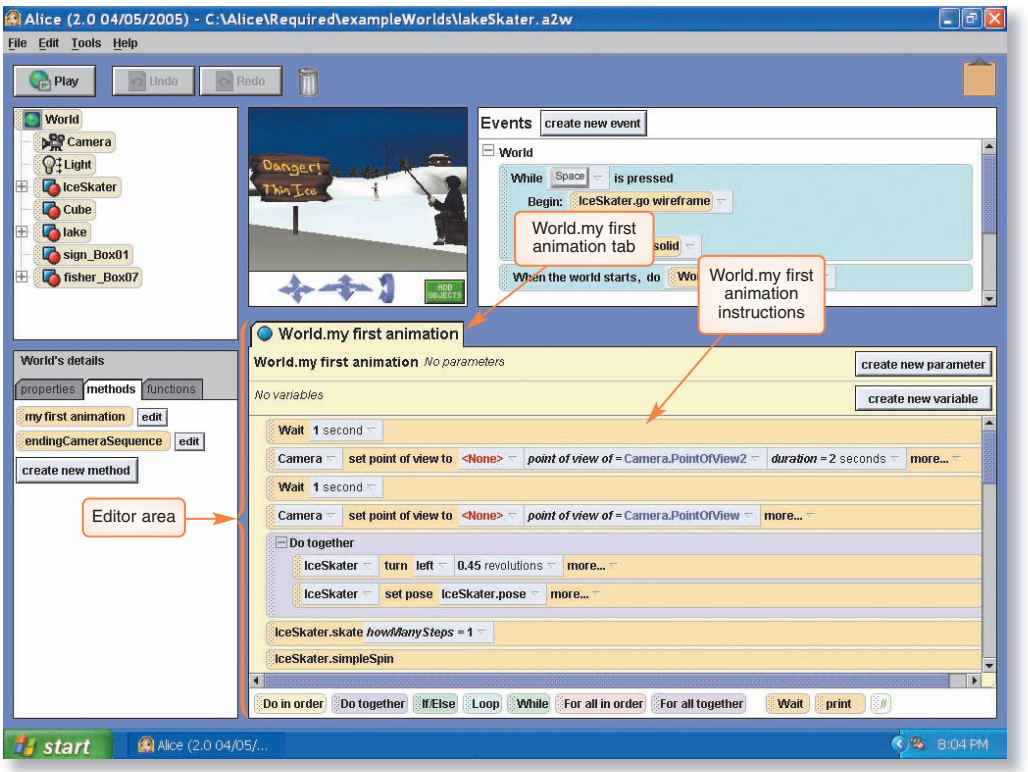

FIGURE 1-25

2

• **Drag the** Wait 1 second **instruction tile from the** World.my first animation **tab in the Editor area and drop it on the Clipboard.**

The instruction is stored in the Clipboard's memory. A white paper appears on the Clipboard whenever there is something in the Clipboard's memory, as shown in Figure 1-26.

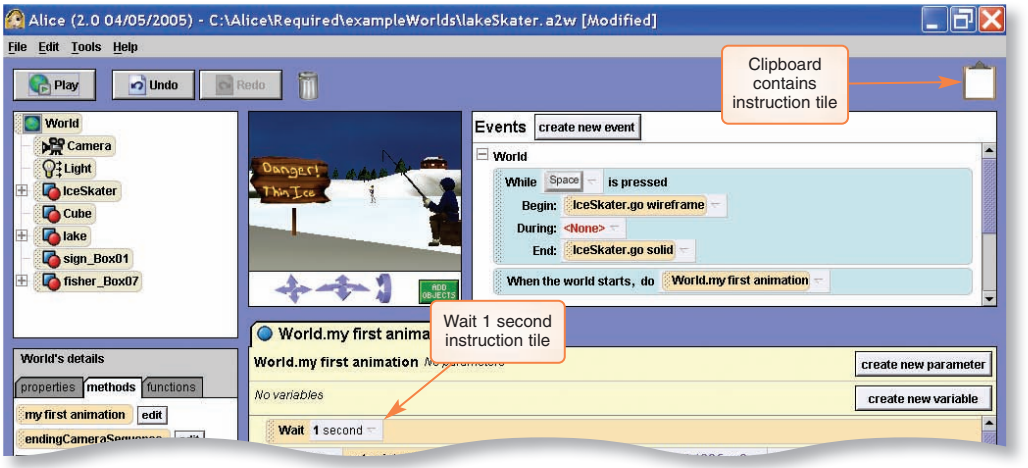

FIGURE 1-26

3

• **Drag the Clipboard and drop it on the** World.my first animation **tab, just below the first** Wait 1 second **instruction tile.**

A new copy of the Wait 1 second instruction tile appears just below the first one, as shown in Figure 1-27.

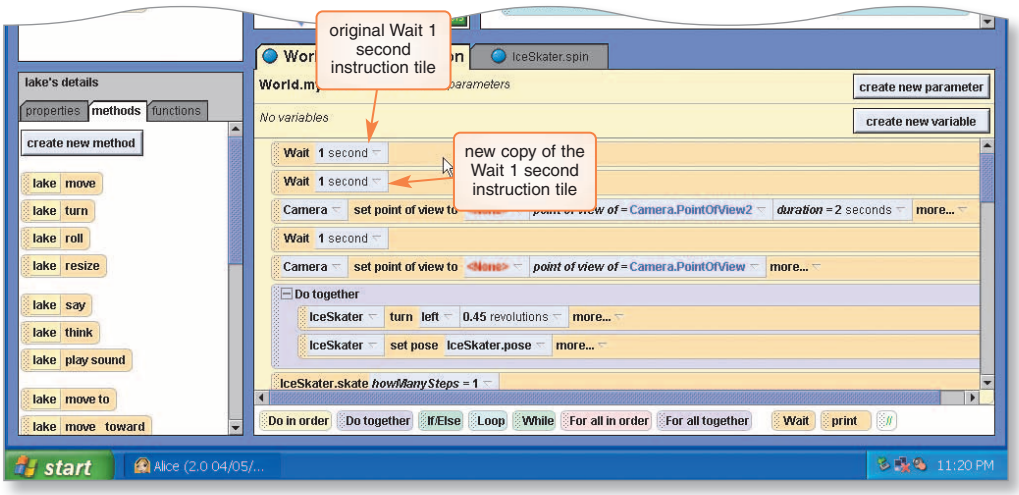

FIGURE 1-27

You can also duplicate a method tile by right-clicking it and selecting make copy on the menu that appears. This works only with methods and not with Alice objects.

The Trash Can

The **Trash can** icon just below the menu bar is used for deleting both Alice objects and instruction tiles in Alice methods. You can delete an item by dragging and dropping it on the Trash can icon, as shown in the following step.

To Delete an Instruction Tile from a Method

1

• **Drag the second instance of the** Wait 1 second **instruction tile from** World.my first animation **in the Editor area and drop it on the Trash can.**

The second copy of the Wait 1 second instruction tile is removed from the Editor area (Figure 1-28).

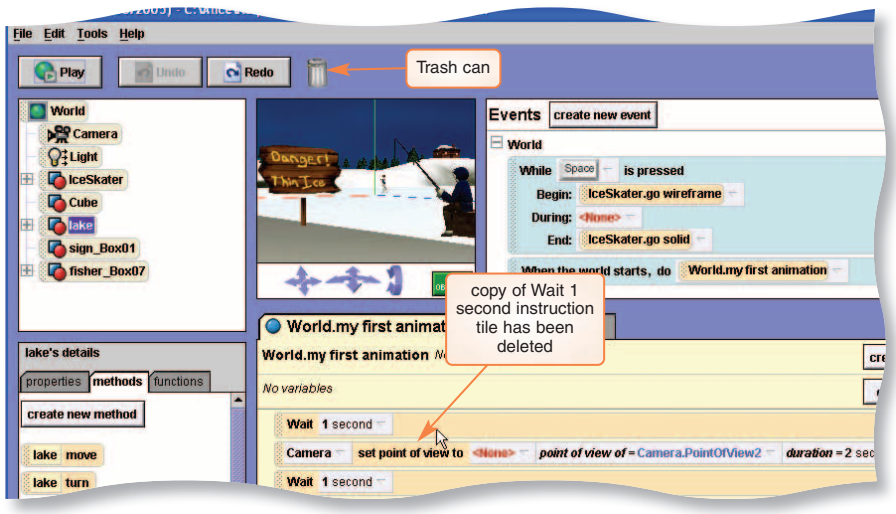

FIGURE 1-28

You can also use the Trash can to delete an object from the Alice World, as shown in the following steps.

To Delete an Object from an Alice World

1

• **Drag the** fisher_Box07 **object tile from the Object tree and drop it onto the Trash can.**

The fisherman sitting on the box disappears from the Alice World window (Figure 1-29).

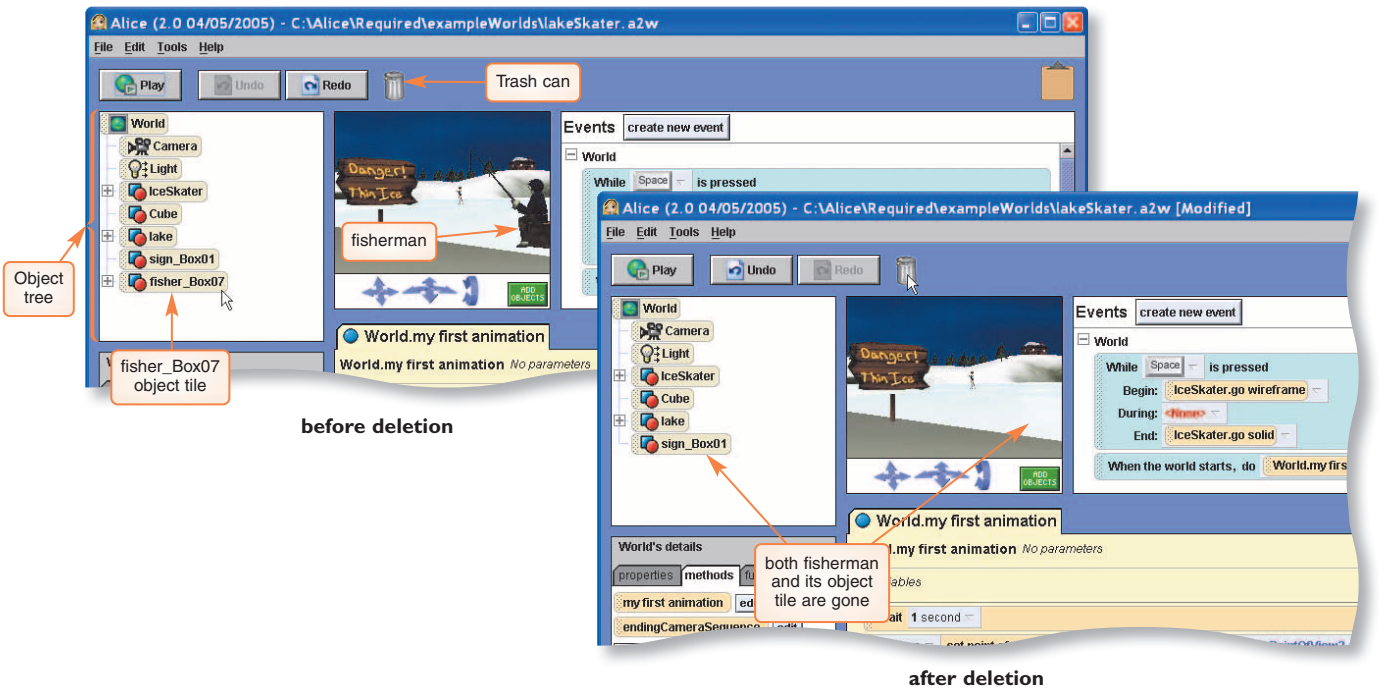

FIGURE 1-29

Other Ways

1. Right-click an object or tile, click delete

The Play Button

The Play button is used to start an Alice world. In the terminology of virtual reality software, this is called "playing" a world or "running" a world. People with a background in film and video production tend to use the term "playing," while computer programmers tend to use the term "running," but in the world of virtual reality software the terms are often used interchangeably. You shortly will use the Play button to view a world in Alice.

The Undo and Redo Buttons

The Undo and Redo buttons, located below the menu bar (see Figure 1-30), are also useful when editing an Alice world. You can undo the last change you made by clicking the Undo button. The effects of the Undo button can be reversed by clicking the Redo button. Alice can remember the last several dozen changes that you made. The following steps undo the deletion of the fisherman object.

To Undo and Redo Changes to an Alice World

1

- **Click the** Undo **button.**

The fisher_Box07 object tile reappears in the Object tree and the fisherman sitting on the box reappears in the Alice World window (Figure 1-30).

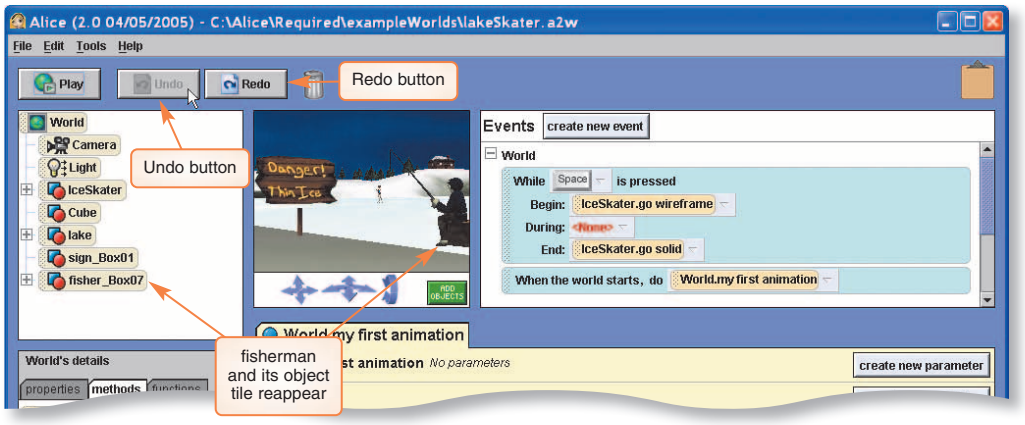

FIGURE 1-30

2

- **Click the** Redo **button.**

The fisherman sitting on the box and its fisher_Box07 object tile are deleted again, as you have just redone the instruction to delete it from the Object tree and the Alice World window (Figure 1-31).

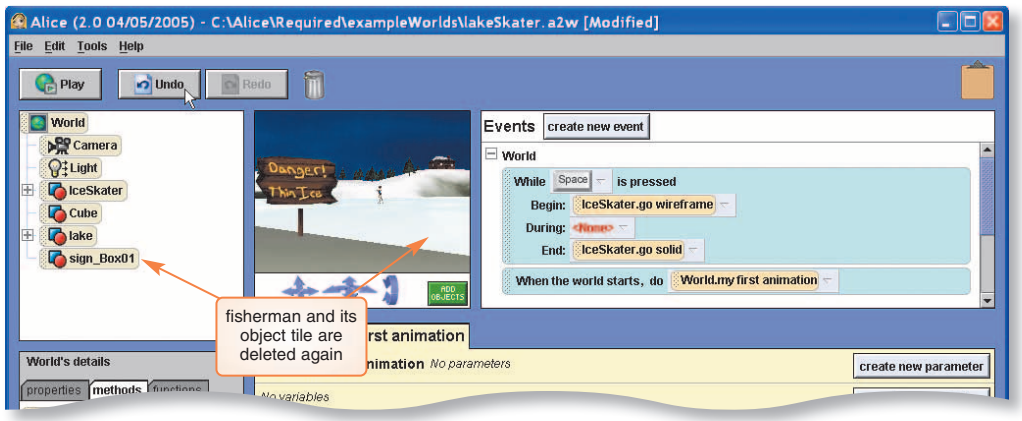

FIGURE 1-31

3

- **Click the** Undo **button again.**

The fisher_Box07 object tile reappears in the Object tree and the fisherman sitting on the box reappears in the Alice World window.

Other Ways

1. Press CTRL+Z to undo a change and CTRL+Y to redo a change.

The Menu Bar

The Alice interface has a menu bar at the top of the screen with four menus: File, Edit, Tools, and Help. The menus are used much less frequently in Alice than in most other computer programs. The following section describes the menus and their functions.

FILE MENU The Alice File menu, similar to the File menu found in other programs, has commands for opening, closing, and saving Alice worlds, as well as options to export an Alice world as a movie file or as an HTML Web page, as shown in Figure 1-32. It also has options to let you import saved objects into an Alice world, add 3-D text to your world, and to make a billboard (which is a graphical image, such as a photograph, that has been converted into an Alice object).

You will use these options throughout the projects in this book, except for the Export movie option, which has been disabled in the current version of Alice.

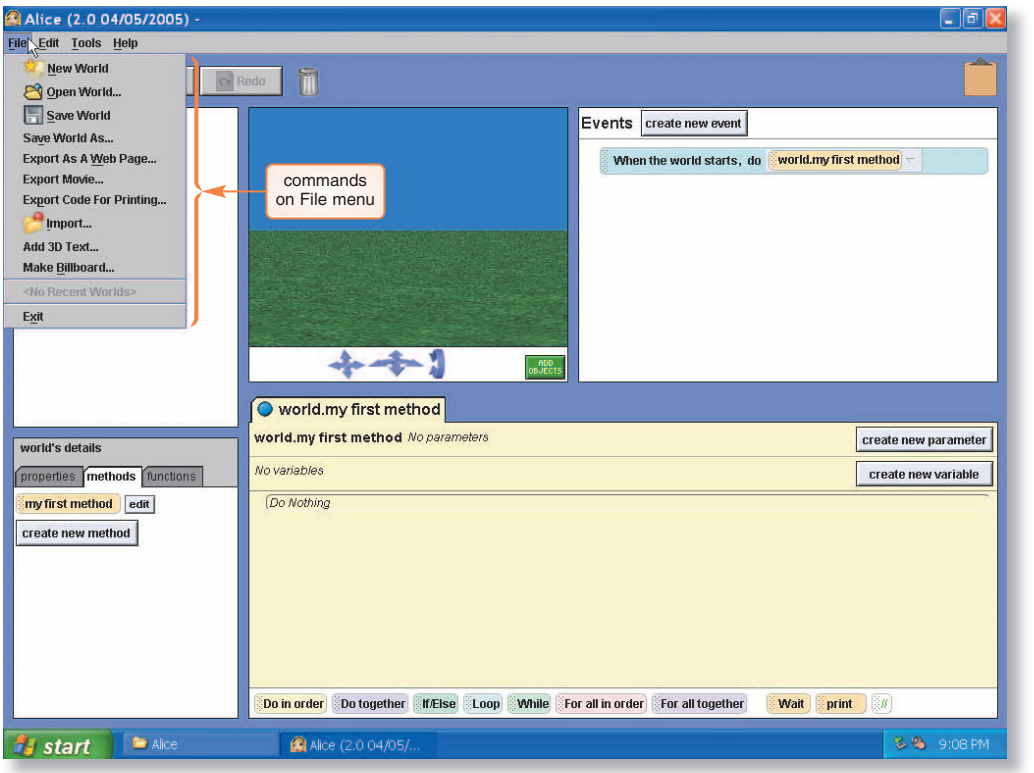

FIGURE 1-32

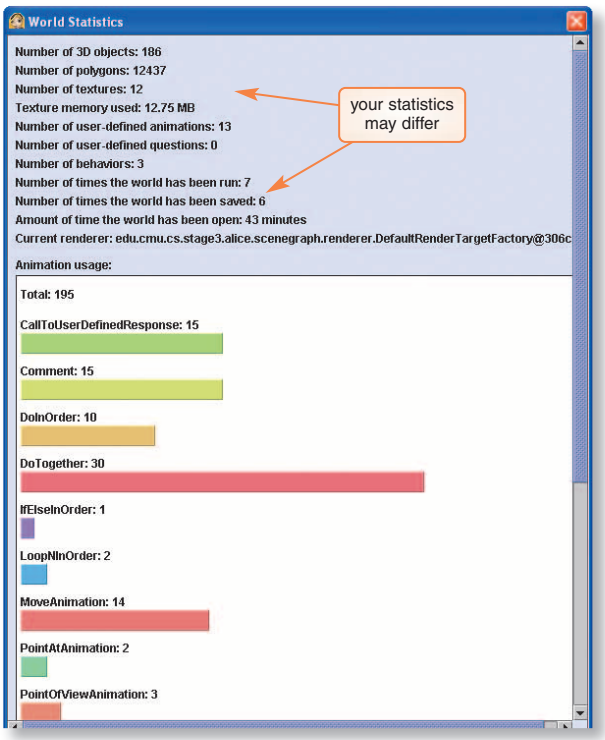

FIGURE 1-33

EDIT MENU Currently, the only option on the Alice Edit menu is Preferences, which is used to change settings for Alice. Appendix B describes these settings. The most important thing to know for now is that the Alice Edit menu is not used to edit Alice methods in the same way that, for example, the Edit menu in Microsoft Word can be used to edit elements in a document. Instead, Alice emphasizes the use of a drag-and-drop interface, which uses the editing icons and buttons described earlier in this project.

TOOLS MENU The Tools menu contains three options: World Statistics, Text Output, and Error Console. The World Statistics option allows you to see statistics, such as the number of objects in a world, the time the world has been open, and many other useful items.

The World Statistics window is shown in Figure 1-33. Only some of the information here will be meaningful to you until you learn more about Alice and computer graphics. You can use this feature, however, to see the amount of time the current world has been open, how many times it has been run, and how many times it has been saved. You can also see how much memory is used for the world's texture maps, which form the surfaces of objects, such as the **IceSkater**'s clothes. Generally, texture maps and sound files are the two items in an Alice world that consume the most memory.

The Text Output option allows you to see system messages generated as you play Alice worlds, and the Error Console can be used to look at detailed Alice error messages. Both of these tools are rather sophisticated, and are not very useful for novice programmers.

HELP MENU The Help menu has three options: Tutorial, Example Worlds, and About Alice. The Help menu does not contain an option to look up the features of Alice, as you might expect. By not providing a way to look up features, the developers of Alice were hoping to encourage people to learn about Alice through experimentation.

Example Worlds and Tutorial both take you back to the Welcome to Alice! dialog box that you saw earlier in this project. The About Alice option provides general information about the development of Alice and displays the URL for the Alice Web site, www.alice.org, where you can find out more about Alice.

The About Alice version 2.0 04/05/2005 dialog box is shown in Figure 1-34. Take a moment to read the text in the dialog box before continuing.

Q&A

Q: What is a texture map?

A: A texture map is a graphic image that is used to paint the surface of an object. It can be a simple pattern that is repeated on a flat surface such as a grass or snow ground object in Alice, or a more complicated image that is wrapped around a 3-D object, such as the image of the IceSkater's costume, face, and hair.

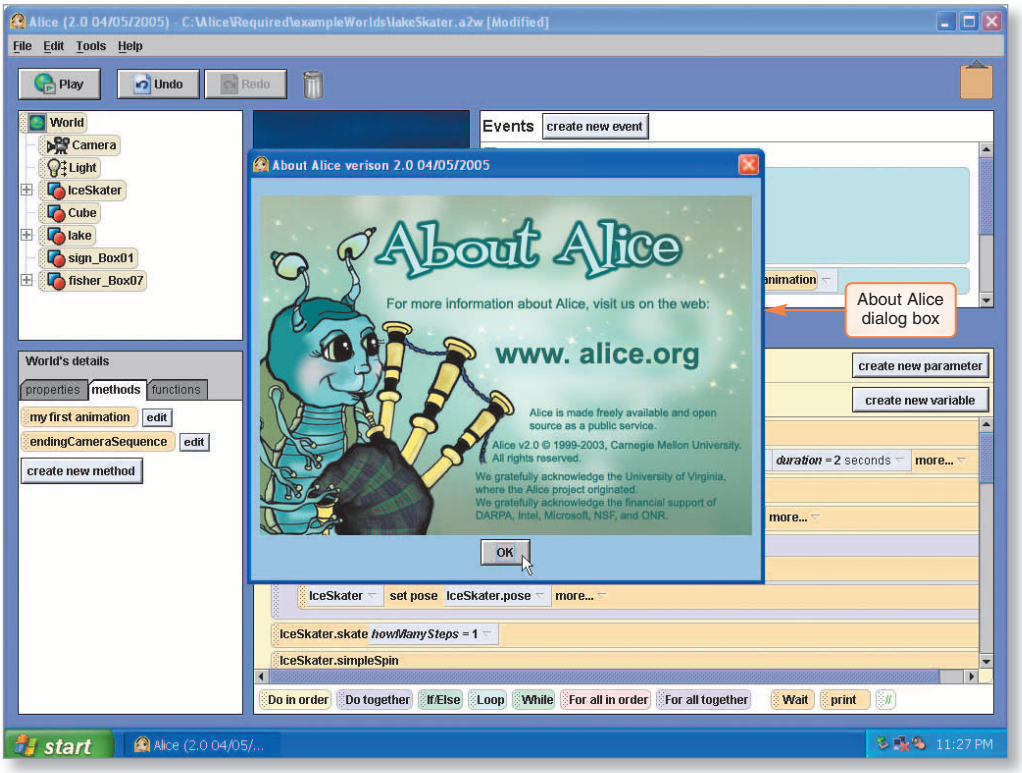

FIGURE 1-34

Playing an Existing Alice World

You can play an open Alice world at any time by clicking the **Play** button on the standard Alice interface. In this part of the project, you will play the **lakeSkater** world to see how this works. Some Alice worlds are interactive in the way a video game is, with mouse and keyboard controls that work while the world plays; others are simply played and viewed like an animated film. The **lakeSkater** world is an example of a film-like world. When the world runs, you will see the skater perform a figure skating routine. As you will see, virtual worlds without user interaction still can be quite complex, especially when camera movements are included in the world.

More About

Error Messages

Occasionally you may see the Alice error message window appear. It contains a button to submit a bug report to the Alice development team at Carnegie Mellon University. Reporting any errors that occur will help to improve future versions of Alice, and in some cases a member of the team may contact you directly. You may also submit bugs and suggestions about Alice through the Alice web site at www.alice.org.

The window in which an Alice virtual world runs has a speed slider control and five buttons across the top of the window: Pause, Resume, Restart, Stop, and Take Picture. The speed slider control is used to change the speed of the world while it is playing. The Pause and Resume buttons work like the pause and play buttons on a VCR or DVD, pausing a world and then resuming it from the point at which it was paused. The Restart button is used to begin playing the current world again from the beginning. The Stop button stops the world that is currently playing and returns you to the standard Alice interface. Once the Stop button is clicked, you will need to click the standard interface's Play button to replay the world. Finally, the Take Picture button captures an image of the currently playing world and saves it in a data file. The stored image file can be viewed and used like any other computer image file. Appendix B has more information on changing the settings for Alice's screen capture function. (Note that the world continues to play while the Image captured and stored window is on the screen. In many cases, it is probably best to pause the playing world before taking a picture.)

The following steps show how to play, pause, stop, and restart an Alice world, and how to capture and save an image of the world.

To Play the lakeSkater World

1

- **Click the** Play **button.**

An Alice virtual world plays in a new window called the World Running window, shown in Figure 1-35. When the skater stops moving, her routine is finished.

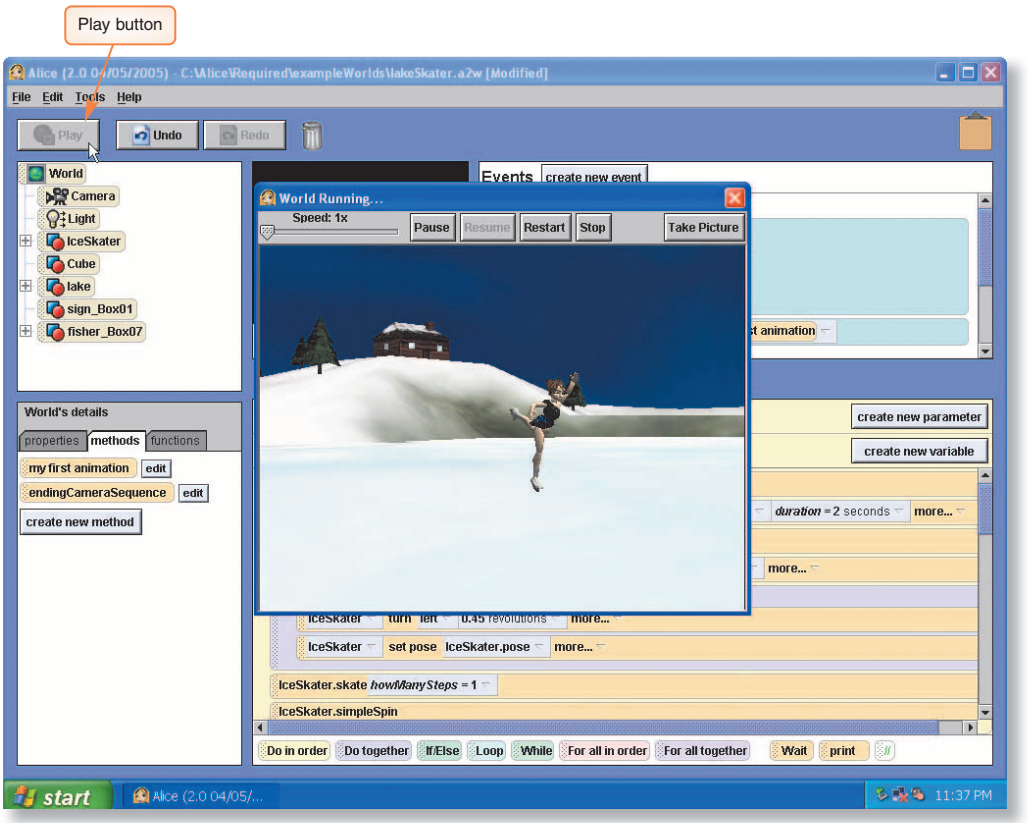

FIGURE 1-35

2

- **Click the** Restart **button in the World Running window.**

The world plays again from the beginning.

3

- **Click the** Restart **button in the World Running window again to restart the** lakeSkater **world, and then experiment with the** Pause **and** Resume **buttons.**
- **Restart the world, and experiment with the speed slider control.**
- **Restart the world, let it play for about 3 or 4 seconds, and then click the** Take Picture **button to take a picture of the world.**

An Image captured and stored dialog box is displayed, showing you the full path name of the file that was saved (Figure 1-36).

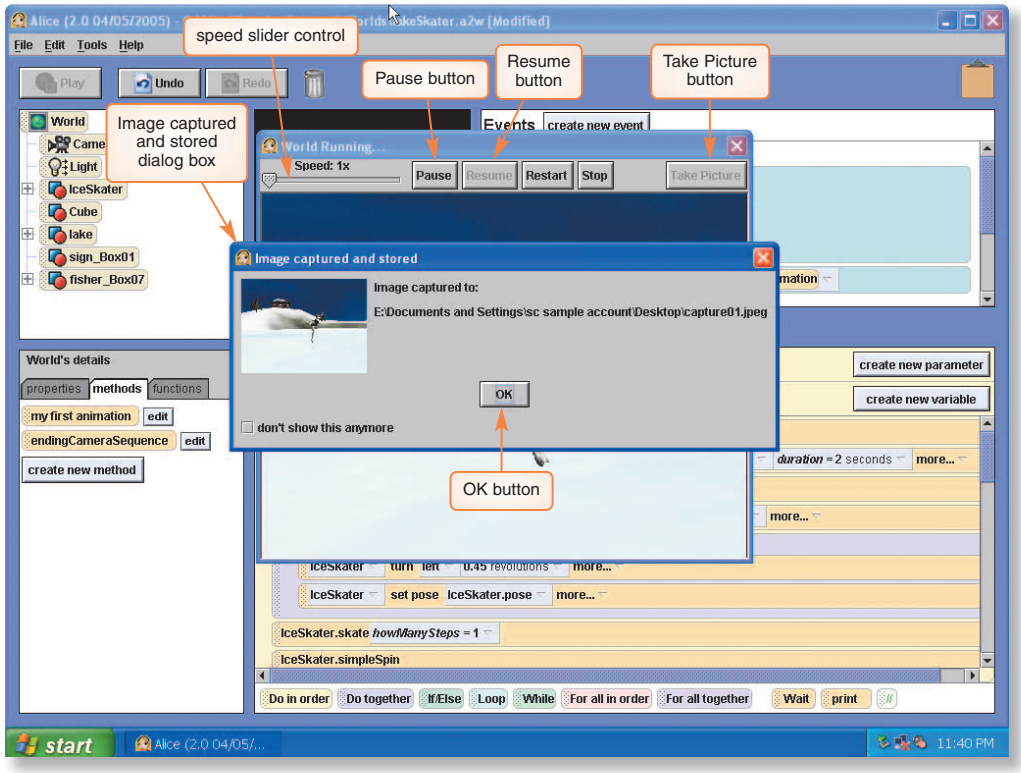

FIGURE 1-36

4

- **Click the** OK **button in the Image captured and stored dialog box.**

The Image captured and stored dialog box closes, and you are returned to the World Running window, where the Alice world is still playing (Figure 1-37).

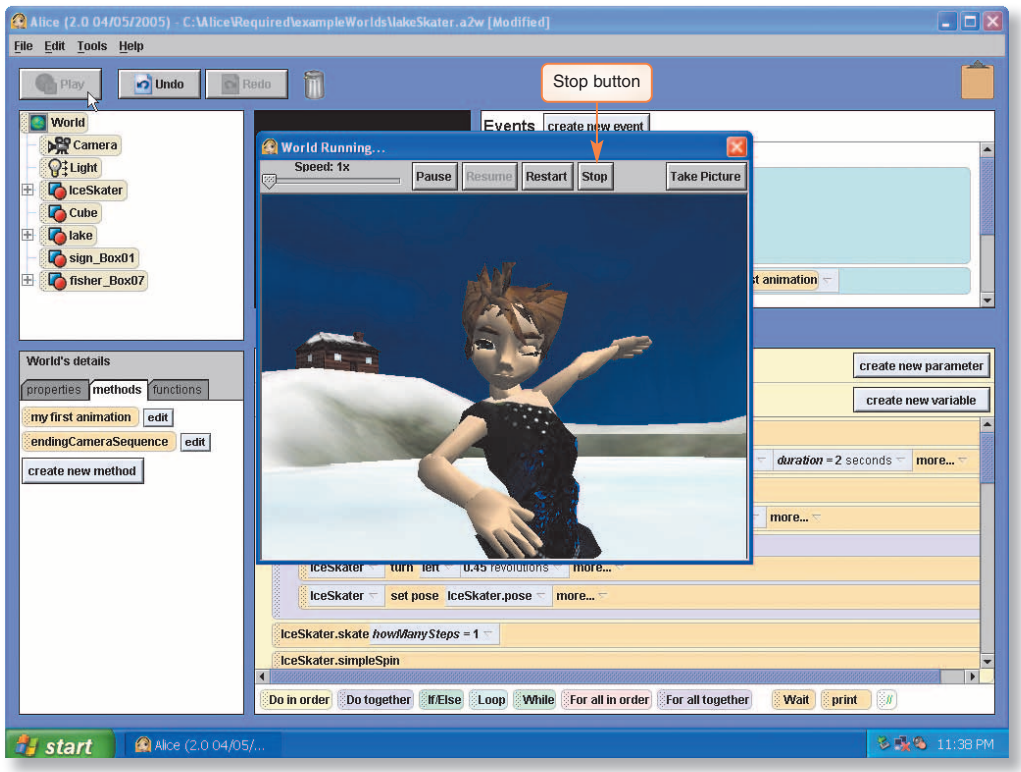

FIGURE 1-37

5

• **Click the** Stop **button in the World Running window.**

The world stops playing and you are returned to the standard Alice interface, as shown in Figure 1-38.

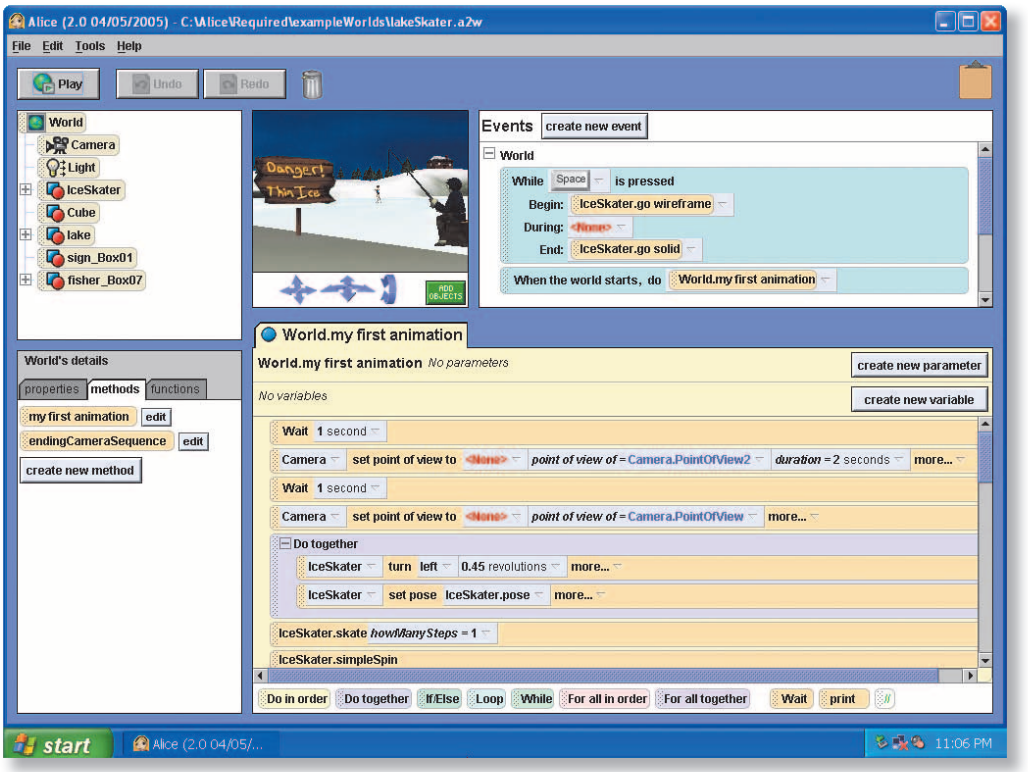

FIGURE 1-38

You have finished exploring the Alice interface and now know how to load and play an existing world. In the next section of this tutorial, you will create, save, and play a world of your own that is similar to the lakeSkater world.

Creating a New Alice World

In this section of Project 1, you create and save a simple Alice world that looks something like the lakeSkater example world you previously saw. You start with a blank Alice world, add the skater and the lake scenery, and write a simple method to make the skater perform a few simple actions. In the projects that follow, you add events and additional methods to the world to build an interactive ice skater world while learning about fundamental concepts of computer programming. As you continue, remember that your ultimate goal is not to build an Alice world, nor to learn about all of the features of Alice, but to use Alice to learn about some fundamental concepts of computer programming.

Before continuing, it is a good idea to exit Alice and restart it. Alice worlds are rather large and can fill a great deal of a computer's internal memory. Alice doesn't always empty a world completely from its memory when a new world is loaded, but it does clear the memory when you quit Alice. If too much of the computer's memory is filled, then the computer will run more slowly and could even stop running. The following steps exit and then restart Alice.

To Exit and Restart Alice

1. **Click** File **on the Menu bar to display the File menu, and then click** Exit **on the File menu to exit Alice. Click** No **if asked to save the Alice world.**
2. **Start the Alice program again by double-clicking the Alice icon on the desktop, or in the same way that you were directed to do so by your instructor at the beginning of this project.**

Once the program is loaded you will see the Welcome to Alice! dialog box over the front of the Alice interface, just as you saw when you first started the program earlier in this project (and as shown previously in Figure 1-7). The Examples tab should be visible in the window.

In the following steps, you select a template for a new Alice world, add some objects to the world, position them, and then begin to write methods to animate one of the objects.

To Create a New Ice Skater World

1

- **Click the** Templates **tab on the Welcome to Alice! dialog box, as shown in Figure 1-39.**

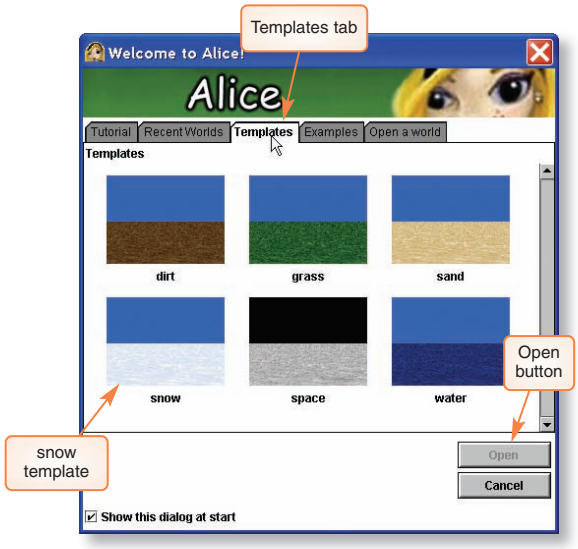

FIGURE 1-39

2

- **Click the** snow **template and then click the** Open **button.**

A new Alice world opens with a snow texture for the ground. Figure 1-40 shows the World window for this new world.

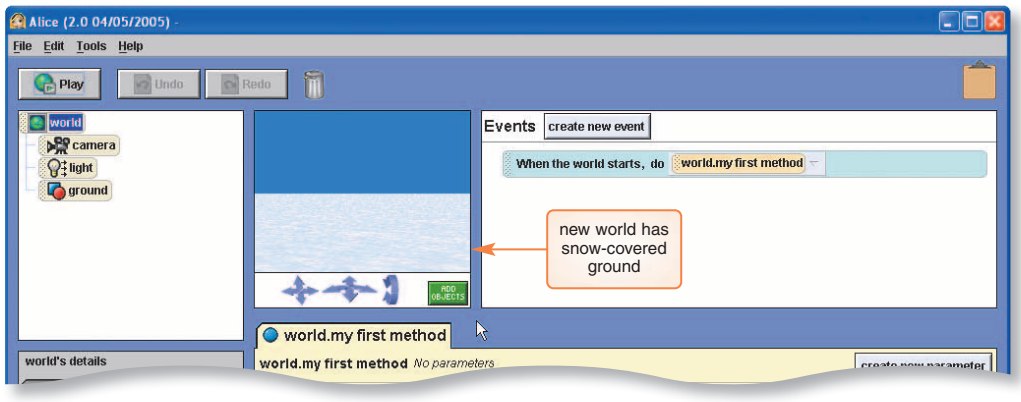

FIGURE 1-40

You now have a new Alice world as the basis for your work, but it only has the four default objects: camera, light, ground, and the world itself. You need to add a few more objects using Scene Editor mode and position them in their starting positions. The large green ADD OBJECTS button in the lower-right corner of the World window will take you into Scene Editor mode.

Once you are in Scene Editor mode you will be able to add objects from Alice's object galleries and move things around to set the scene for your new world. There are two object galleries: a Local Gallery that has been installed on your computer as part of Alice, and a more extensive Web Gallery at Carnegie Mellon University that can be accessed from Alice whenever your computer has a live Internet connection. The projects in this book do not require you to use the Web Gallery, but you may choose to do so as part of the end-of-project exercises if your Internet connection is fast enough.

You will use the same lake scene that you saw in the lakeSkater example world earlier in this project. This scene has many objects in it, including a cabin and more than 100 trees. Fortunately, the scene has already been assembled and saved in one of the object galleries so that you can load it from there instead of having to load and position all of the objects individually. Creating a scene by positioning objects in a virtual world can be a very tedious process that often takes longer than writing all of the methods and events for the new world. The following step allows you to enter Scene Editor mode.

To Enter Scene Editor Mode

1

- **Click the** ADD OBJECTS **button in the lower-right corner of the World window.**

Alice enters Scene Editor mode, as shown in Figure 1-41. Notice that the Events area and Editor area from the standard Alice interface are not visible. The World window is now larger, with object manipulation tools appearing to the right of the window, and the Local Gallery visible at the bottom of the screen. The Local Gallery has tiles for folders that contain Alice objects, such as the Amusement Park folder, the Beach folder, and so on.

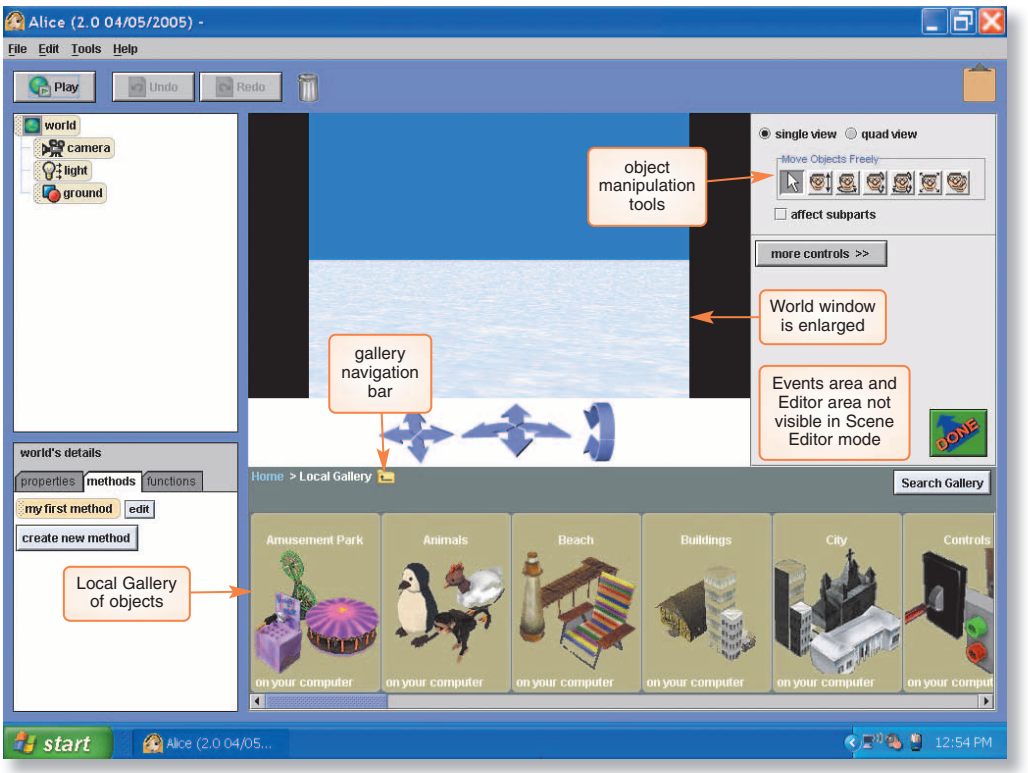

FIGURE 1-41

The galleries are organized as a tree of folders containing related objects. You can navigate the tree of galleries by clicking a gallery folder to enter that gallery, or by using the gallery navigation bar, which is just above the galleries. You may wish to come back to the Local Gallery later on your own to explore some of the folders, but for now you will simply use the gallery to add objects to the ice skater world you are building. You need to add the **lake** object and the **iceSkater** object to the world. The lake is in the Environments folder, and the ice skater is in the People folder. Note that the object you need is not the one named Frozen Lake, but simply Lake.

The Environments folder has a subfolder for Skies and tiles for other Alice classes of environment objects, as shown below in Figure 1-42. You should recall that a class of objects is a set of similar objects that each have the same properties and methods, and that the process of adding an individual object to a world is called instantiation. You are now about to instantiate an object from the Lake class in your Alice world.

To Add the Lake Object to the World

1

- **Using the horizontal scroll bar at the bottom of the screen, scroll to the right until you can see the** Environments **tile. Click the tile to enter the** Environments **folder.**

The Environments folder opens, as shown in Figure 1-42.

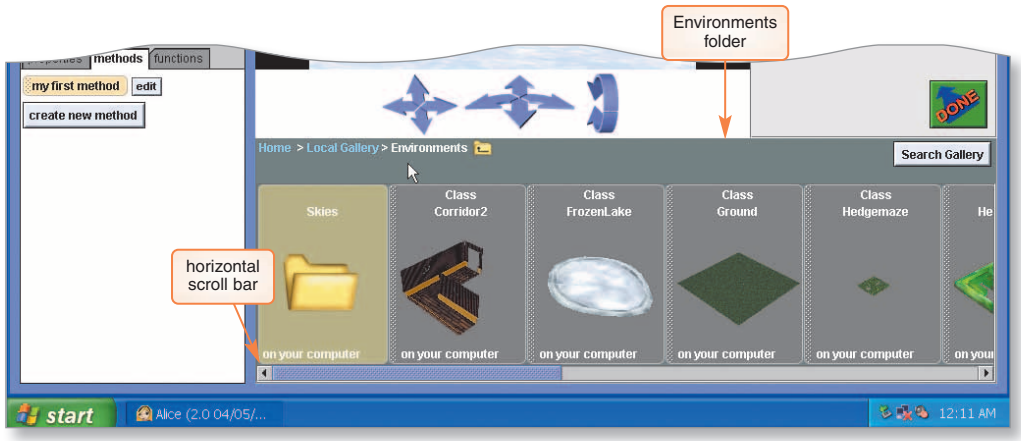

FIGURE 1-42

2

- **Scroll to the right until you can see the** Lake **tile, then click the tile once.**

The Lake class information dialog box appears in the Alice interface, as shown in Figure 1-43. The window shows the size of the object and the number of parts that it has. Notice that this is a rather large object, with 127 parts that consume more than 2 megabytes of memory.

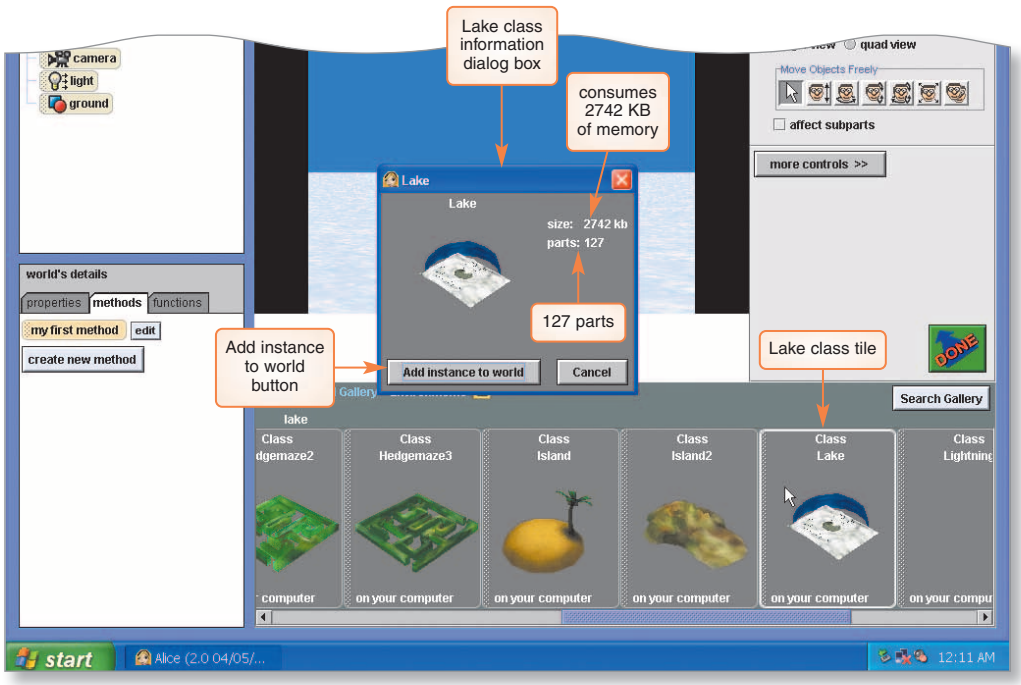

FIGURE 1-43

3

• **Click the** Add instance to world **button to add an instance of the Lake class to your world. It will take a few seconds for Alice to finish adding the object and all of its subparts to your world.**

Part of the lake object is now visible in the World window. Also notice that a lake tile has been added to the Object tree for your world. Both can be seen in Figure 1-44.

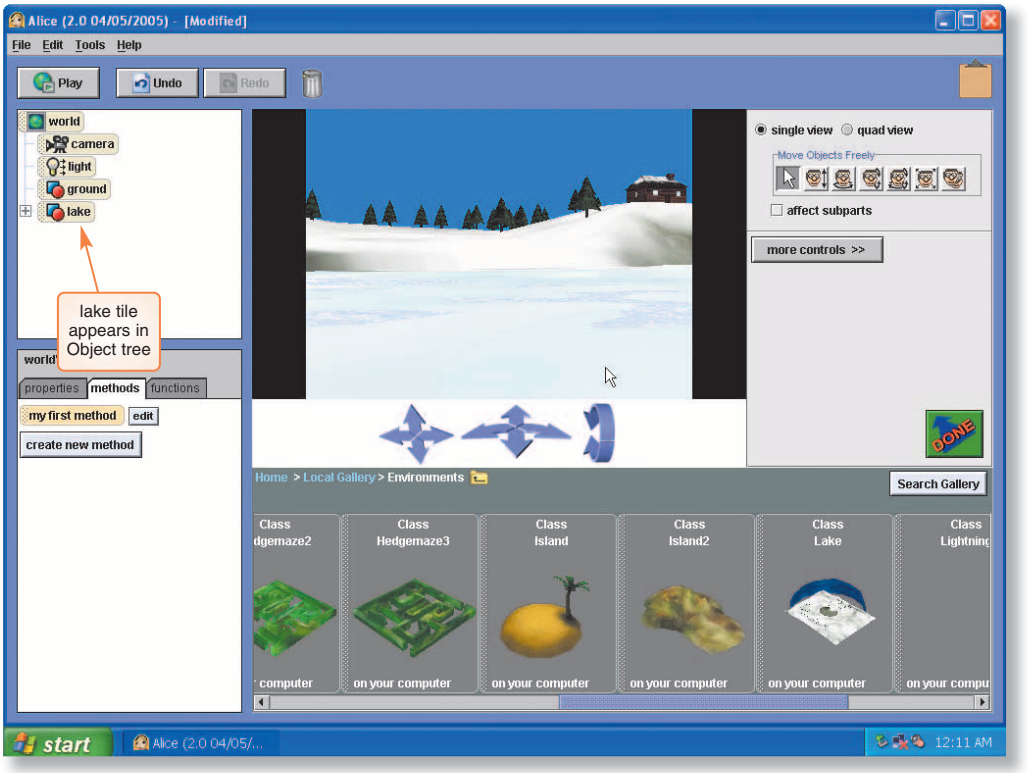

FIGURE 1-44

Next you need to add the ice skater to your world, as shown in the following steps. The ice skater object class is named IceSkater and is stored in the People folder.

To Add the Ice Skater to the World

1

• **Click the** Move Up a Level **button, shown in Figure 1-45, to go up one level in the** Local Gallery's **tree of folders.**

The main level of the Local Gallery is now shown.

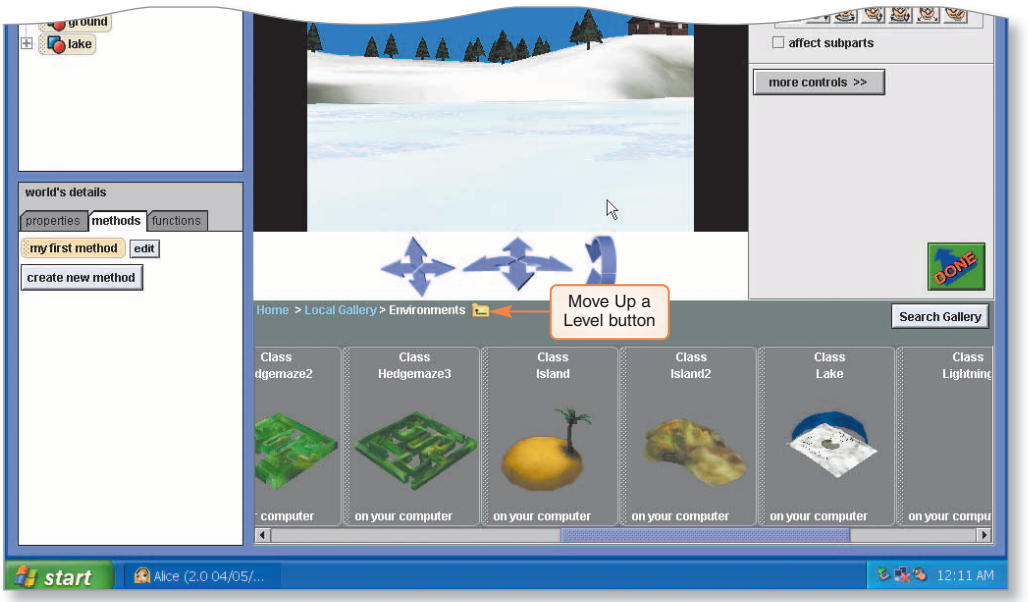

FIGURE 1-45

2

• **Scroll to the right until the** People **folder tile is visible, then click the** People **folder tile to enter the folder.**

The People folder is now visible on the screen (Figure 1-46).

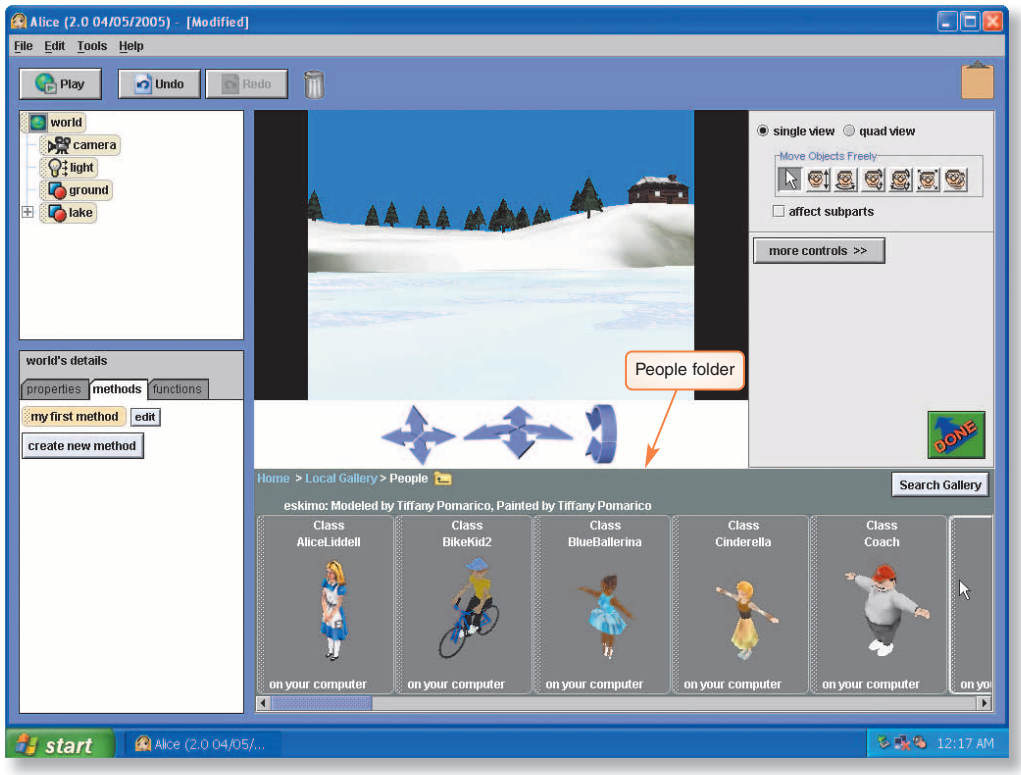

FIGURE 1-46

3

• **Take a moment to scroll back and forth through the folder and see some of the many characters that you can add to Alice worlds, then find and click the** IceSkater **class tile.**

The IceSkater information dialog box is displayed. Notice that an instance of this object has 32 parts and will consume 493 kilobytes of memory, as shown in Figure 1-47.

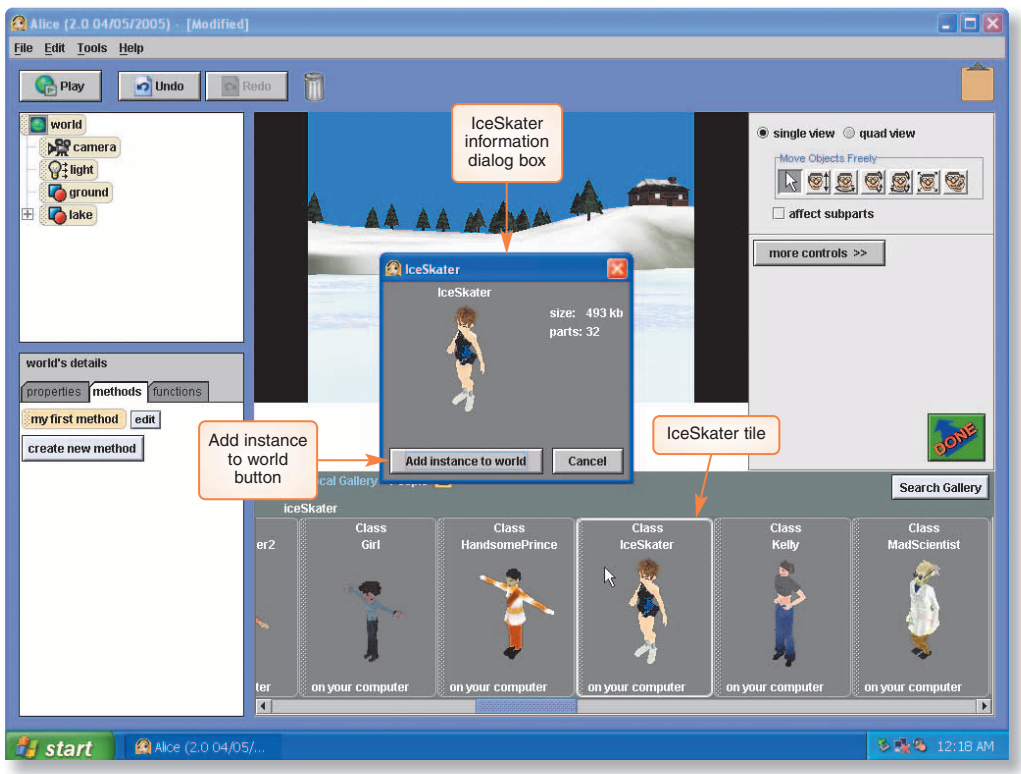

FIGURE 1-47

4

• **Click the** Add instance to world **button to add an instance of the** IceSkater **class to your world.**

Your screen should now look something like Figure 1-48. It may look slightly different, depending on your computer's screen size and screen resolution.

FIGURE 1-48

More About

Naming Objects and Classes

The name, lake, in the Object tree starts with a lowercase letter, whereas the name, Lake, on the Lake class tile in the object gallery starts with an uppercase letter. Normally, object class names begin with an uppercase letter and object instance names begin with a lowercase letter, although programmers do not always follow this convention.

You now have a blank world with the objects you need in place. Before doing anything else, you should save the world. After that, you will create your first method in Alice, which will animate the iceSkater.

You should expect to make mistakes as you create computer software, so it is a good idea to know how to recover from your mistakes. Alice's Undo button is one feature that will help when this occurs, but you can protect yourself even more by frequently saving your work. Each time you finish setting the scene for a new Alice world, you should save that scene so you can get it back if problems occur during editing. In the following steps, you save the new Alice world.

The Save World As dialog box has features similar to Save As dialog boxes for Microsoft Office programs. By default, Alice saves files on the Windows desktop, but you may save files wherever you wish. Ask your instructor where you should save your files, as this may depend on how your classroom computers are configured. If you are working on your own, you should decide where to save your Alice worlds, and remember (or write down) where you saved them. Choose a folder that will be easy to find.

The Save World As dialog box contains a File name text box for you to enter the name of the world. You will name this world skaterHello.a2w, as shown in the following steps.

To Save the Alice World

1

• **Click** File **on the menu bar to display the File menu, then click** Save World **on the File menu.**

Because this is the first time you have saved this world, a Save World As dialog box is displayed (Figure 1-49).

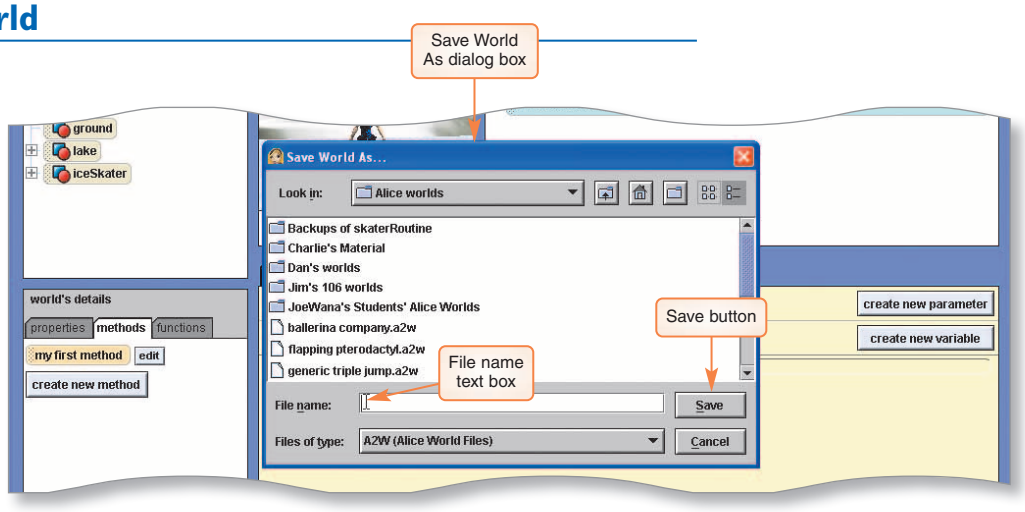

FIGURE 1-49

2

• **In the Save World As dialog box, navigate to the folder in which you wish to save your world, type** `skaterHello` **in the File name text box, and then click the** Save **button.**

A file named, skaterHello.a2w, has been saved in the folder you selected. This world uses more than 3 megabytes of storage space, so you may have to wait a few seconds while the world saves.

Positioning the iceSkater in the New World

Once the necessary objects have been added to a new world, they need to be properly positioned. You can do this by using the seven object manipulation tools to the right of the Scene Editor World window. The functions of these tools are summarized in Table 1-1.

Table 1-1 The Object Manipulation Tools

BUTTON	NAME	FUNCTION
	Pointer tool	Selects an object and moves the object parallel to the ground. (X-Y plane movement)
	Vertical tool	Moves an object up or down. (Z-axis movement)
	Turn tool	Turns an object parallel to the ground. (X-Y plane rotate)
	Rotate tool	Rotates an object forward or backward (Z-axis rotate).
	Tumble tool	Freely turns and rotates an object in any direction.
	Resize tool	Changes the size of an object.
	Duplicate tool	Creates a new instance of the same object.

More About

Storage Space

Virtual worlds, by their very nature, tend to consume a lot of storage space. The number of objects affects their size, but sound files and texture maps tend to consume the most space.

You only need to position the IceSkater in this world. She should be between the center and the right side of the screen, and she should be pushed back from her original location. She should also be in a three-quarter view, turned so that she is facing about halfway between looking directly at the camera and directly off to the left side.

The following steps position the iceSkater object.

To Position the iceSkater

1

- **If you are not already in Scene Editor mode, click the** ADD OBJECTS **button to enter Scene Editor mode.**
- **The** Pointer **tool is already selected. Click and drag the** iceSkater **back and to the right so that she is in the location shown in Figure 1-50.**

The iceSkater is now positioned.

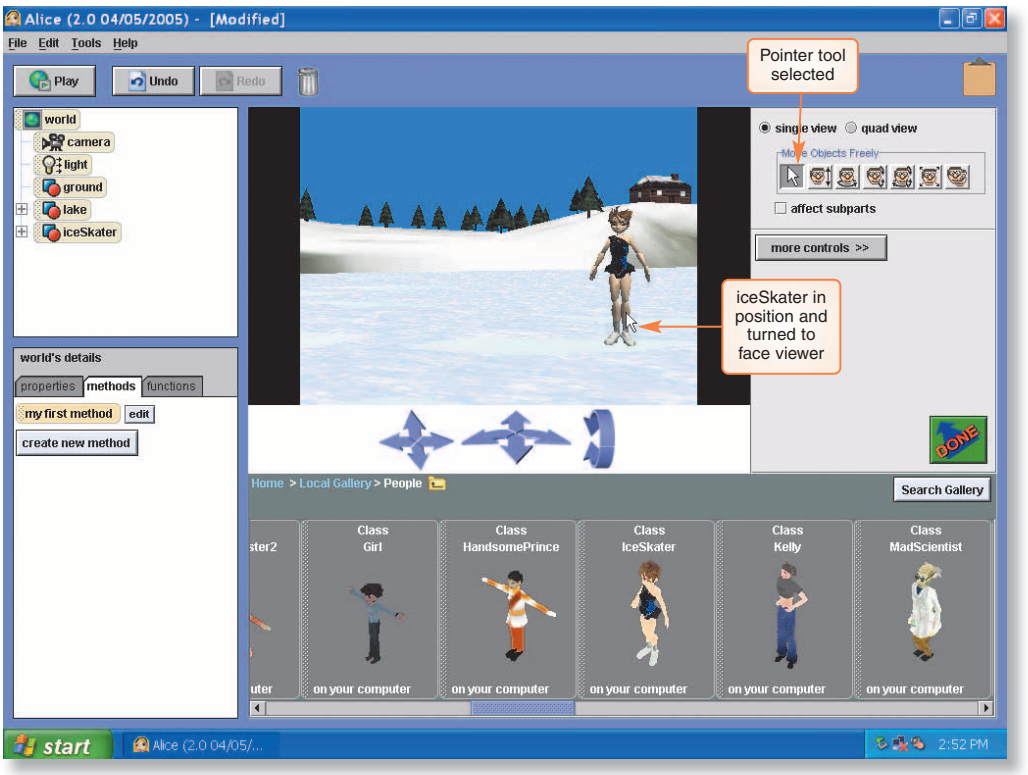

FIGURE 1-50

2

- **Select the** Turn **tool and turn the object so that she is facing as shown in Figure 1-51.**

The iceSkater is facing the correct direction.

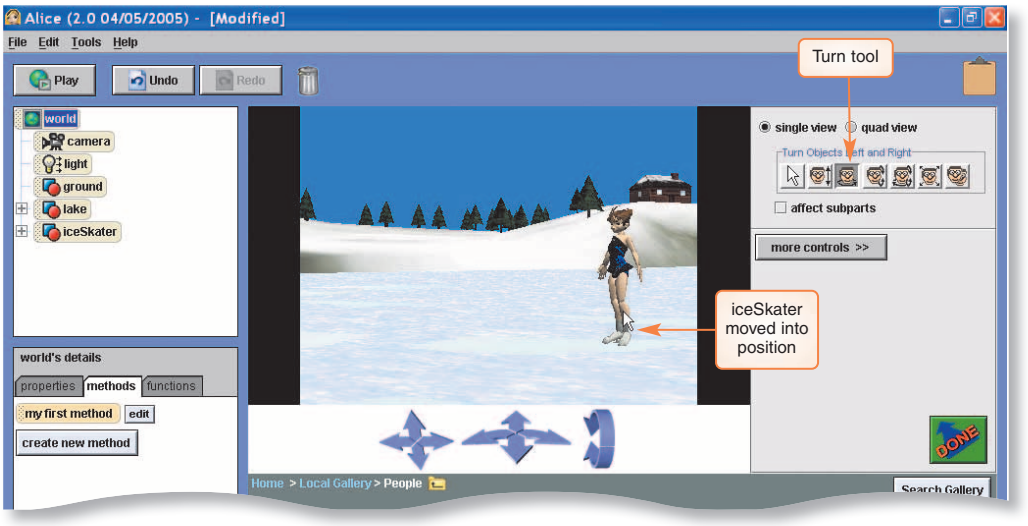

FIGURE 1-51

3

• **The** iceSkater **is now in place. Click the** DONE **button to close Scene Editor mode and return to the standard Alice interface.**

The standard Alice interface is restored (Figure 1-52).

4

• **Click** File **on the menu bar, then click** Save World **on the File menu to save the world again with the skater in position.**

Since you have already saved the world once, it will be saved this time without asking you for a file name.

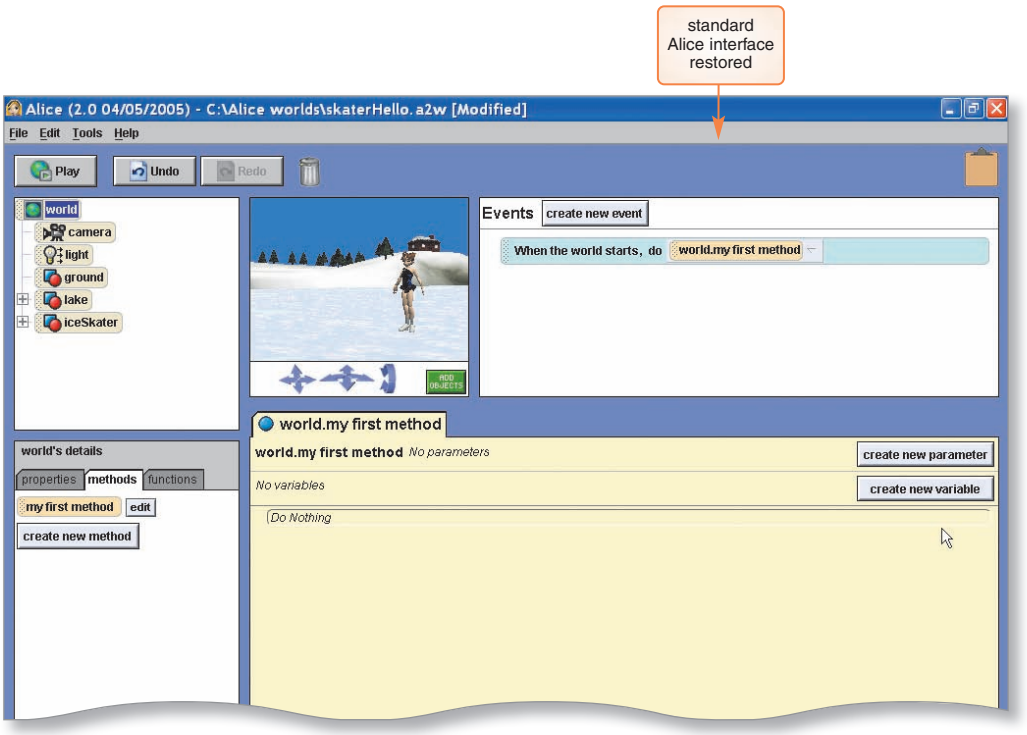

FIGURE 1-52

Creating a Method to Animate the iceSkater

Next, you will create a method to animate the iceSkater. To be precise, you will add instructions to the default method, world.my first method, that appears in each new Alice world. The full name of each Alice method has two parts separated by a period. The name of the object with which the method is associated comes before the period, and the specific name of the method comes after the period. In this case, world is the object and my first method is the specific name of the method. world.my first method should be visible in the Editor area and a my first method tile should be visible on the methods tab in the Details area.

Currently, world.my first method has no instructions. The finished method will look like Figure 1-53. It is a simple sequence of instructions with no looping or branching. The method contains ten tiles — eight instruction tiles and two Do together control tiles that function as follows:

- The first instruction sets a pose for the skater. A pose is a position that an object can assume. The person who created the iceSkater object also took the time to create several poses and store them as properties of the object, which you will see shortly.
- The second instruction causes the ice skater to spin around twice.
- The third and fourth instructions are in a Do together tile. The skater will assume a new pose while moving forward 2 meters.
- The fifth instruction causes the iceSkater to spin around twice again.
- The sixth and seventh instructions, in a Do Together tile, will cause the skater to go back to her original pose while also turning to face the camera.
- The last instruction (not visible in Figure 1-53) causes the iceSkater to say "Hello, World!"

Q&A

Q: What if **world.my first method** is not visible in the Editor area?

A: First, click the **world** tile in the Object tree. Next, click the **methods** tab in the Details area, then click the **edit** button next to **my first method** on the **methods** tab.

Q&A

Q: Why does the skater in Project 1 say "Hello, World?"

A: The "Hello World!" program first appeared in a book for the C programming language by Brian Kernighan and Dennis Ritchie. Dr. Kernighan actually wrote the first "Hello World!" program for a tutorial on an earlier programming language. Ever since then, it has been the custom for someone programming in a new language to write a version of the "Hello World!" program as his or her first program. The Association for Computing Machinery (ACM) "Hello World" project page is on Louisiana Tech's Web site at http://www2.latech.edu/~acm/HelloWorld.shtml.

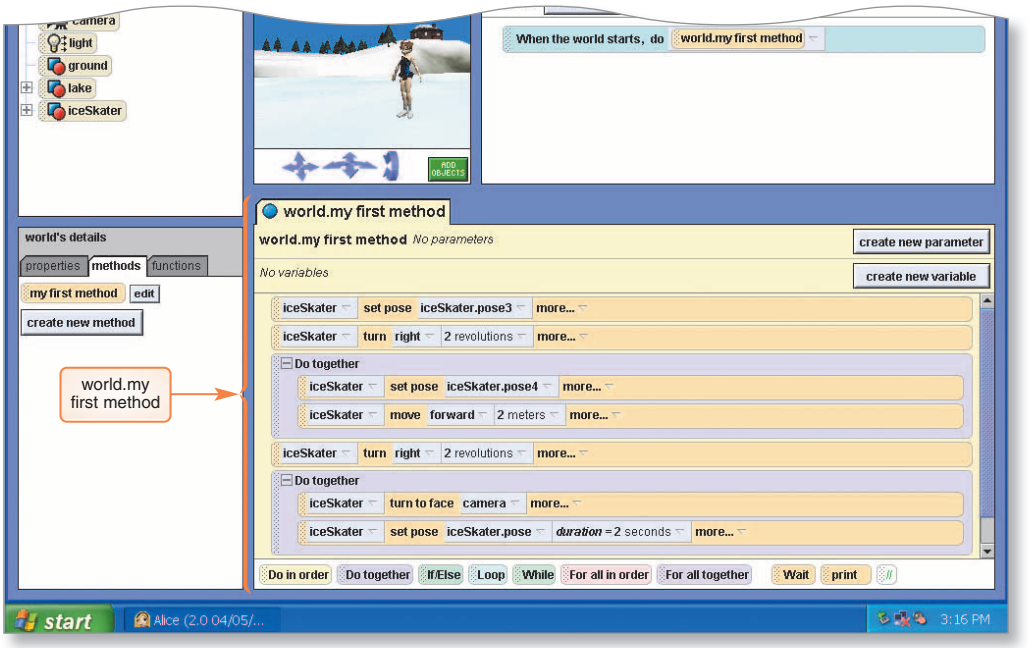

FIGURE 1-53

Notice that there is an Event in the Event area that says, When the world starts, do world.my first method. This is Alice's default event, which will cause world.my first method to play whenever the world starts playing. Later, you will create events of your own. For now, you will simply add the instructions described above to world.my first method to animate your first Alice world, as shown in the following series of steps.

To Start Coding world.my first method

1

- **Click the** iceSkater **tile in the Object tree and then click the** properties **tab in the Details area.**

The iceSkater's properties are visible in the Details area. Notice that four different saved poses for the iceSkater appear on the properties tab above the capture pose button: pose, pose2, pose3, and pose4 (Figure 1-54).

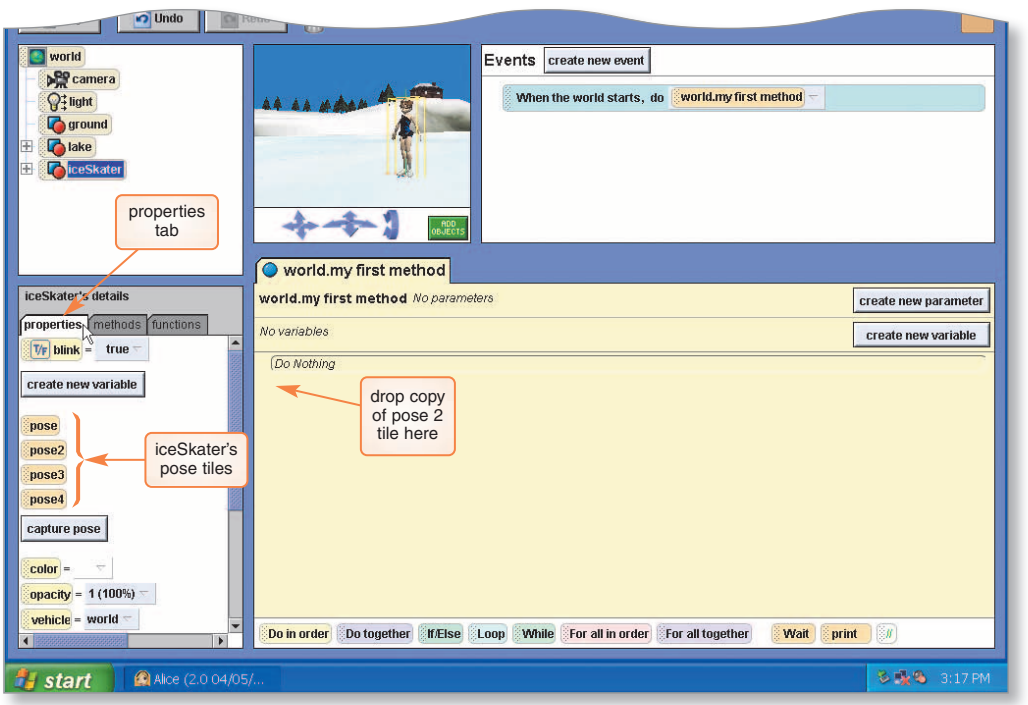

FIGURE 1-54

2

• **Drag a copy of the** pose2 **tile from the** properties **tab and drop it in the blank area below** Do Nothing **in the Editor area.**

An instruction tile to set the iceSkater's pose to pose2 appears (Figure 1-55).

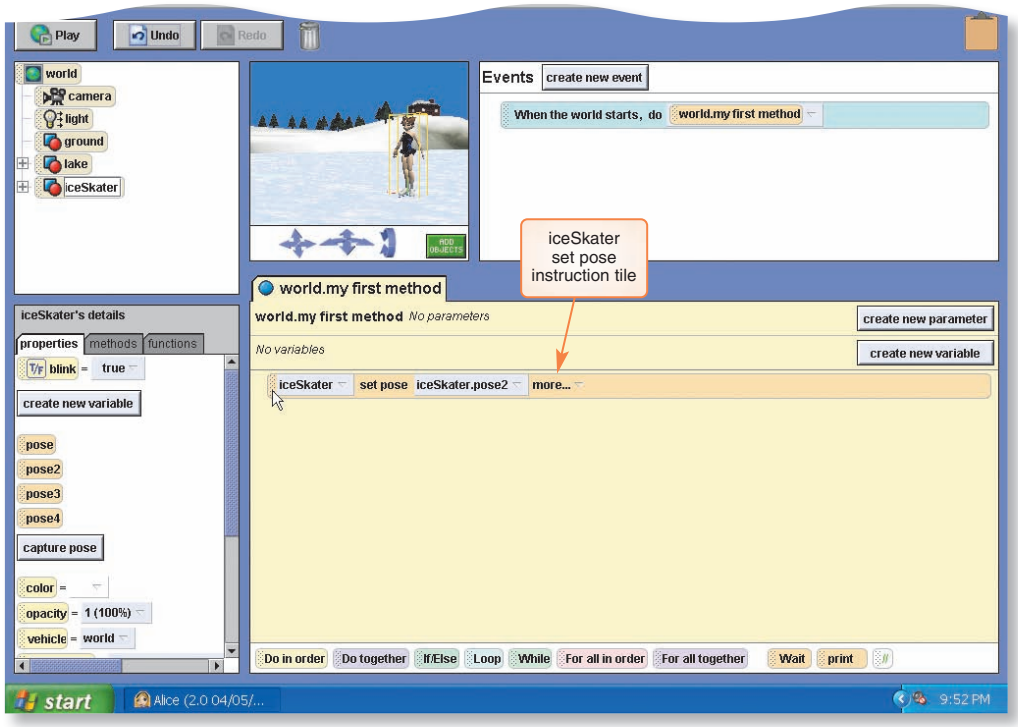

FIGURE 1-55

To Add an Instruction to Make the iceSkater Spin

1

• **Click the** methods **tab in the Details area, then drag a copy of the** iceSkater turn **tile from the** methods **tab and drop it in the Editor area below the** iceSkater set pose **instruction.**

A short menu appears, prompting you to specify the turn's direction, as shown in Figure 1-56.

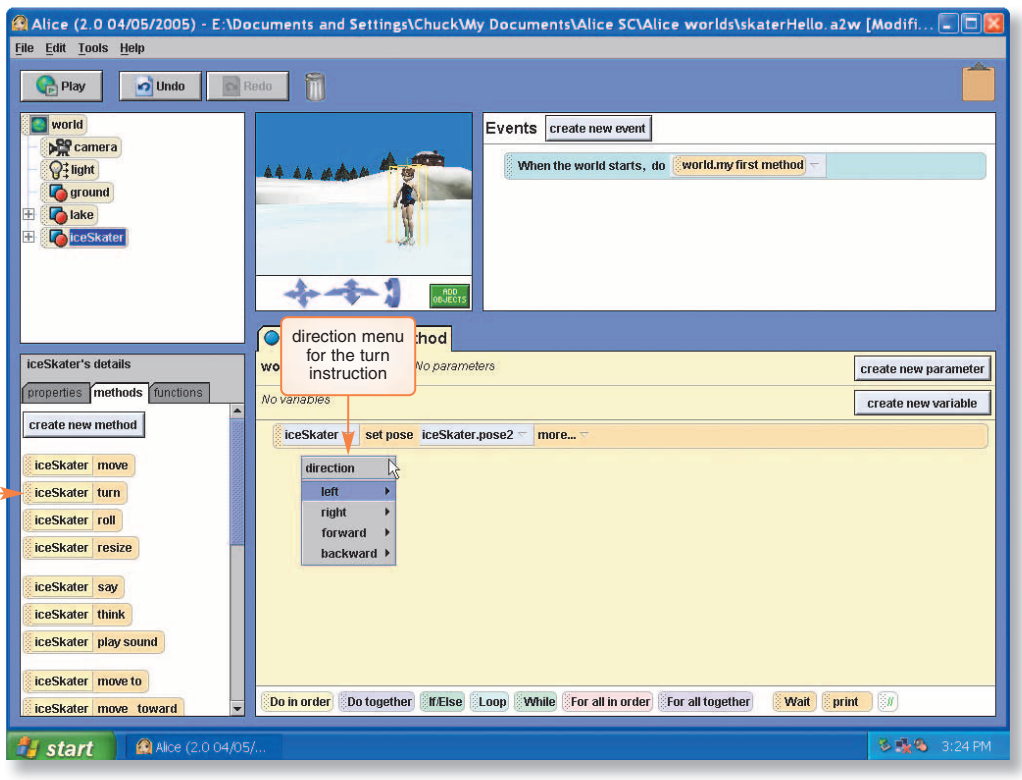

FIGURE 1-56

2

• **Point to** right **on the** direction **menu.**

Another short menu appears, as shown in Figure 1-57, allowing you to set the amount for the turn instruction.

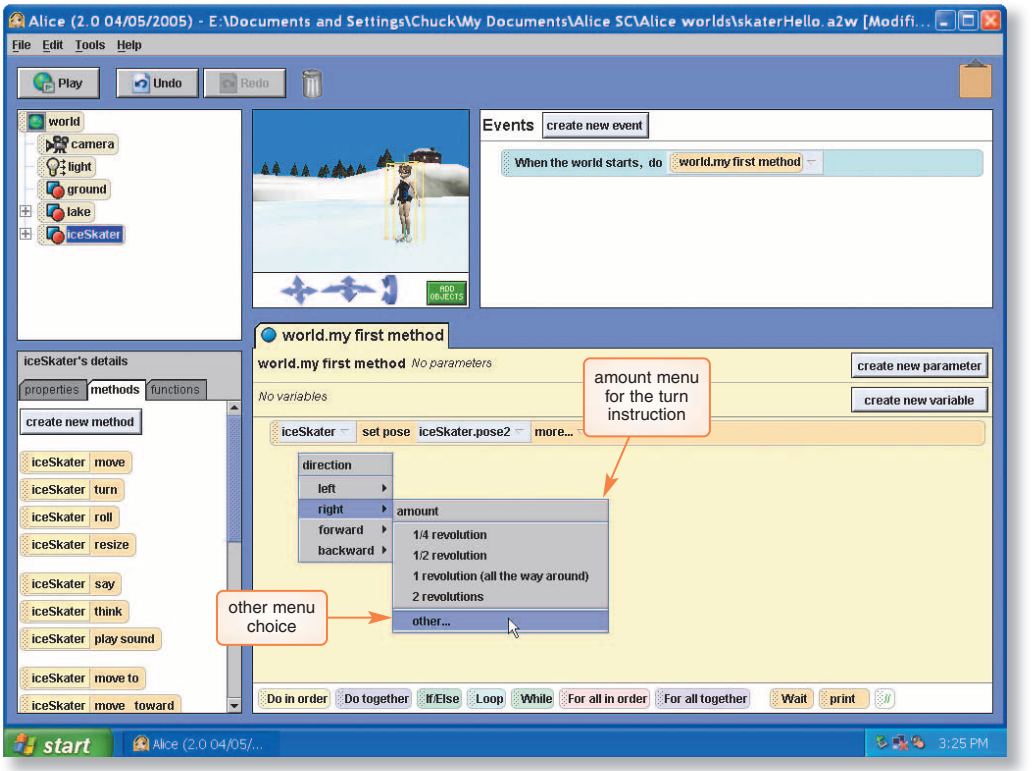

FIGURE 1-57

3

• **Click** other **on the** amount **menu.**

The Custom Number dialog box, which looks like a calculator, appears (Figure 1-58).

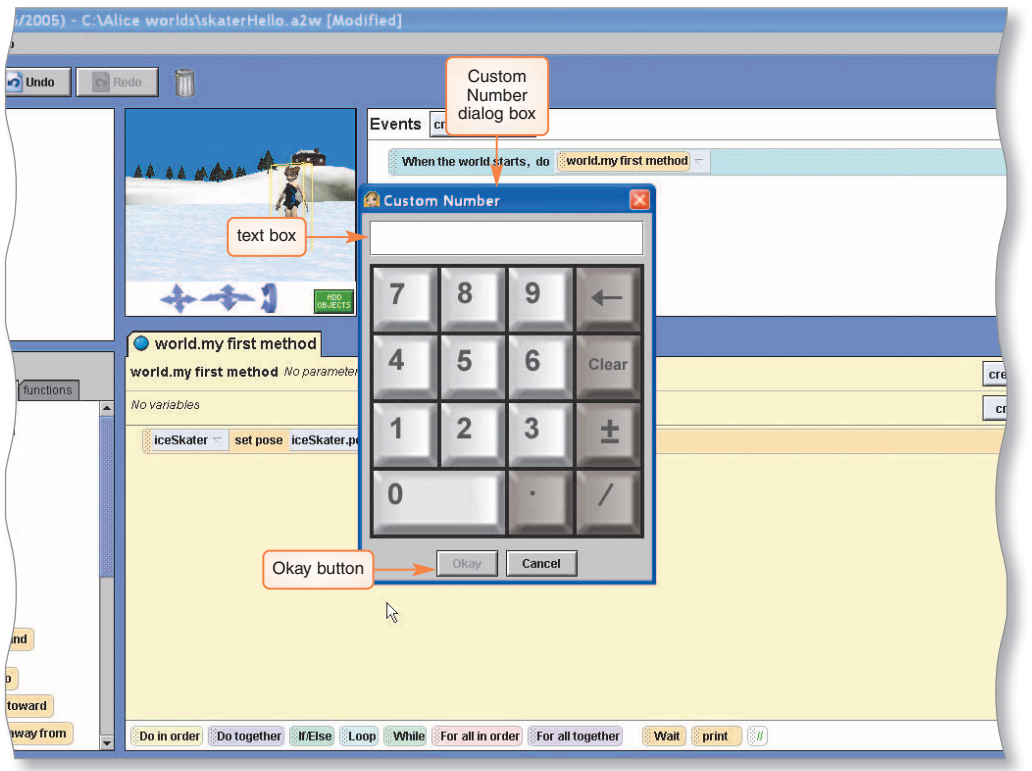

FIGURE 1-58

4

• **Type** 2 **in the text box and then click the** Okay **button.**

An iceSkater turn right 2 revolutions instruction tile appears in the Editor area (Figure 1-59).

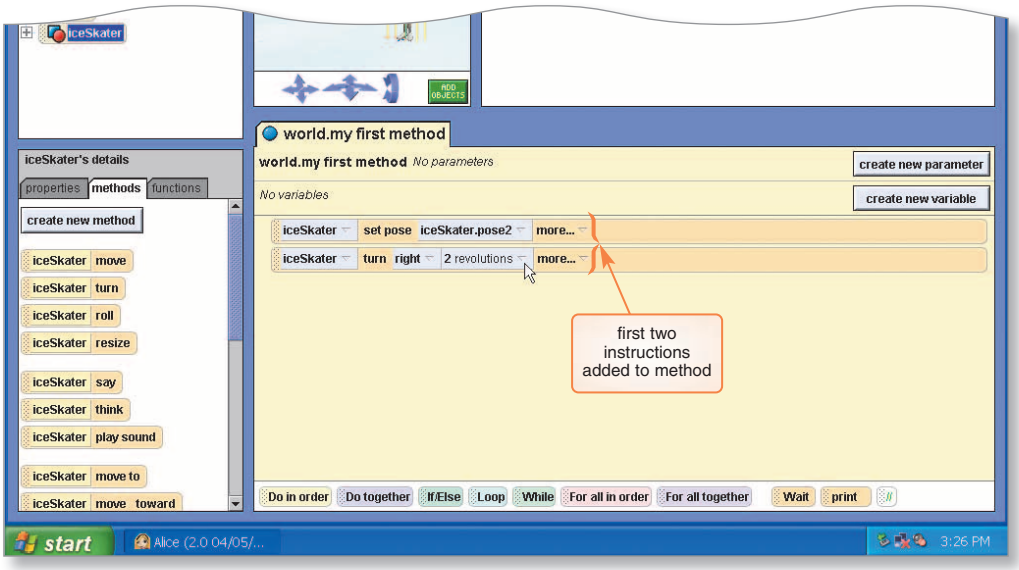

FIGURE 1-59

Other Ways

1. Click appropriate buttons on calculator-style keypad in Custom Number dialog box

The next two instructions, to again change the iceSkater's pose and move the iceSkater forward, should happen at the same time, so they should be in a Do together tile. The Do together tile is one of several logic and control tiles at the bottom of the Editor area, as shown in Figure 1-60.

To Add Instructions to Make the iceSkater Change Poses While Moving

1

• **Drag a copy of the** Do together **tile from the bottom of the Editor area and drop it in the Editor area below the first two instructions.**

A blank Do together tile appears in the method (Figure 1-60).

2

• **Click the** properties **tab in the Details area, then drag a copy of the** pose4 **tile and drop it in place of** Do Nothing **in the** Do together **tile.**

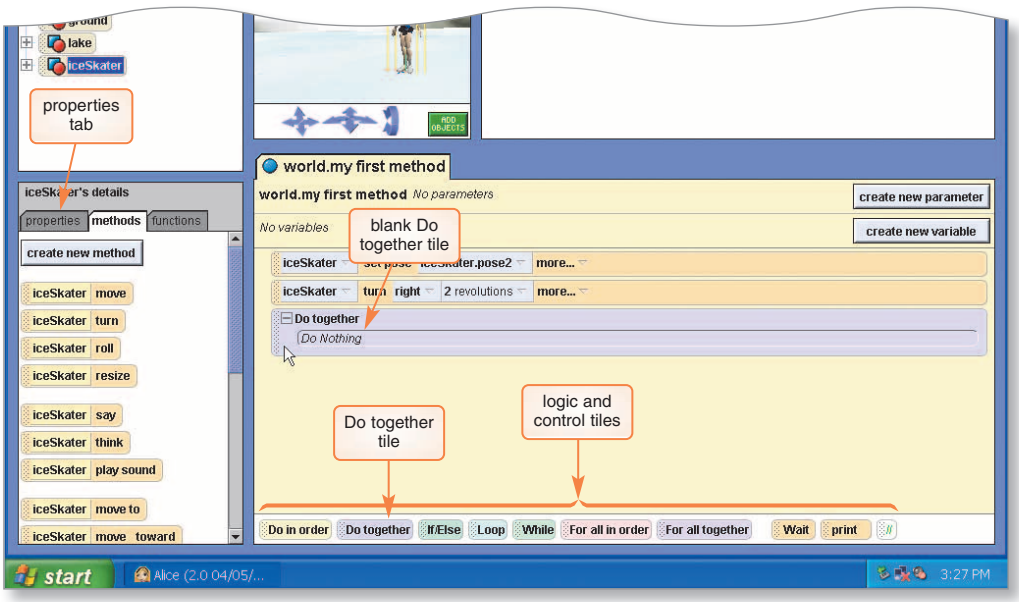

FIGURE 1-60

The Do together tile now contains an instruction to set the iceSkater's pose to pose4.

3

• **Click the** methods **tab in the Details area, then drag a copy of the** iceSkater move **tile and drop it just below the** iceSkater set pose **tile that you just put into place.**

A menu appears, allowing you to set the direction for the move instruction as up, down, left, right, forward, or backward. These are the six major directions in three-dimensional space.

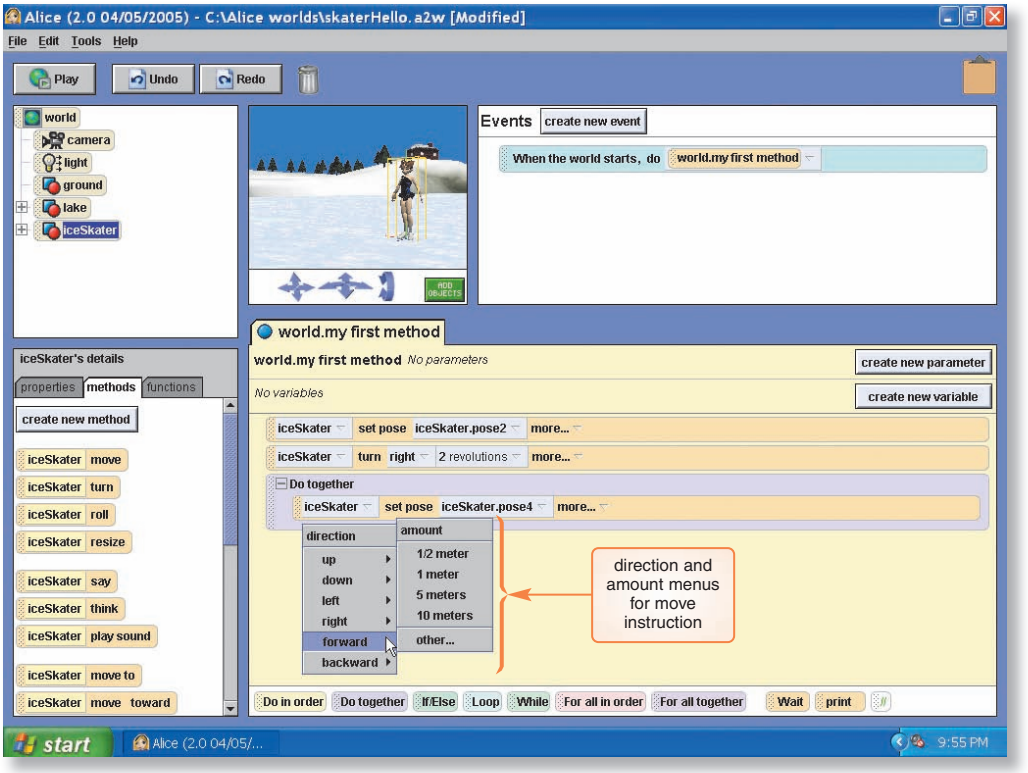

FIGURE 1-61

4

• **Point to** forward **on the** direction **menu, then click** 2 meters **on the** amount **menu, as shown in Figure 1-61. If** 2 meters **is not an option on the** amount **menu, click** other..., **type** 2 **in the Custom Number window, and then click the** Okay **button.**

An iceSkater move forward 2 meters instruction tile appears in the Do together tile below the iceSkater set pose tile (Figure 1-62).

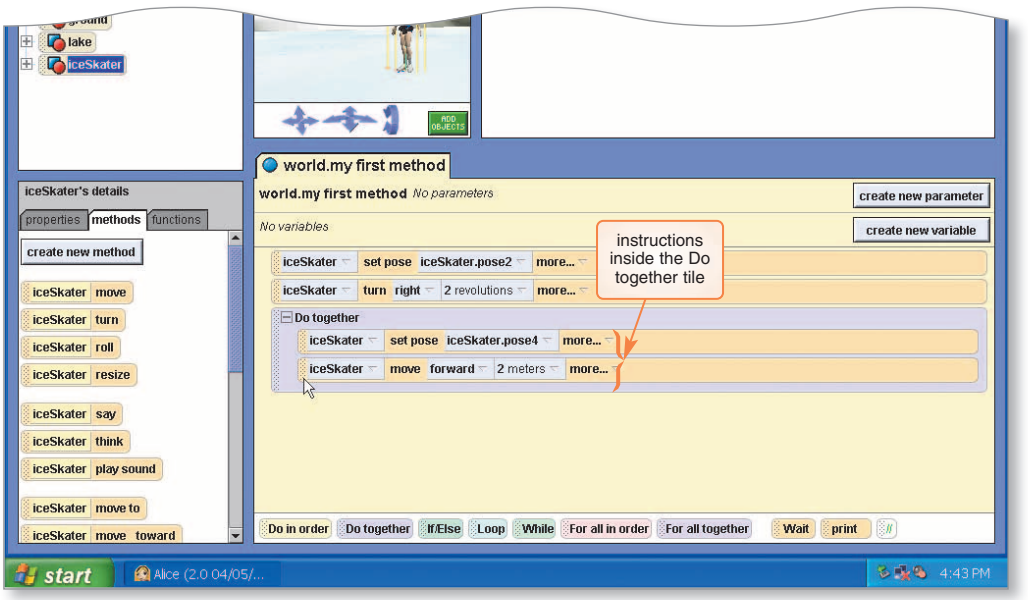

FIGURE 1-62

More About

Alice Parameters

Distance parameters in Alice are in meters because the development of Alice was funded in part by the National Science Foundation, which requires the use of the metric system in educational projects.

Next, you need an instruction to make the ice skater spin around. In this case, the iceSkater will turn right 2 revolutions. The direction and number of revolutions are parameters for the turn method, just as direction and amount were parameters for the move method above. The following steps add an instruction to make the skater spin again.

To Add an Instruction to Make the iceSkater Spin Again

1

• **Drag a copy of the** iceSkater turn **tile from the Details area and drop it below the** Do together **tile in the Editor area.**

A menu appears asking for the direction for the turn instruction.

2

• **Point to** right **on the** direction **menu, and then select** 2 revolutions **on the** amount **menu.**

An iceSkater turn right 2 revolutions instruction appears in the method (Figure 1-63).

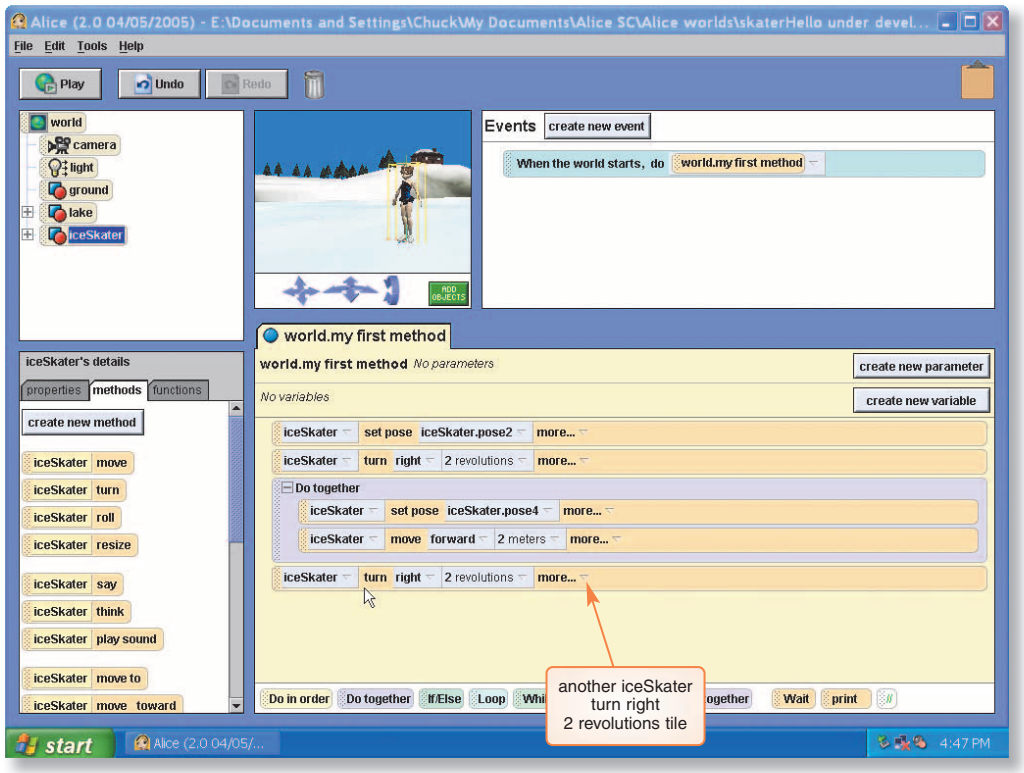

FIGURE 1-63

After the iceSkater spins around twice, the sixth and seventh instructions in the method should cause her to go back to her original position while turning to face the camera. These two instructions will happen at the same time, so they should be placed in a Do together tile, as shown in the following steps.

To Make the iceSkater Pose and Face the Camera at the Same Time

1

• **Drag a copy of the** Do together **tile from the bottom of the Editor area and drop it in** world.my first method, **below all of the existing instruction tiles.**

A blank Do together tile appears at the bottom of the method (Figure 1-64).

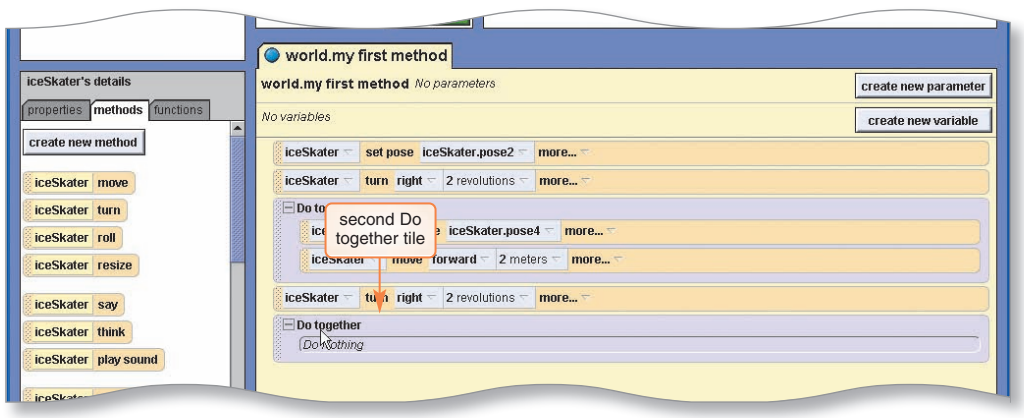

FIGURE 1-64

2

• **Click the** methods **tab in the Details area, scroll down if necessary, and then drag a copy of the** iceSkater turn to face **tile and drop it in the lower** Do together **tile.**

The target menu appears (Figure 1-65).

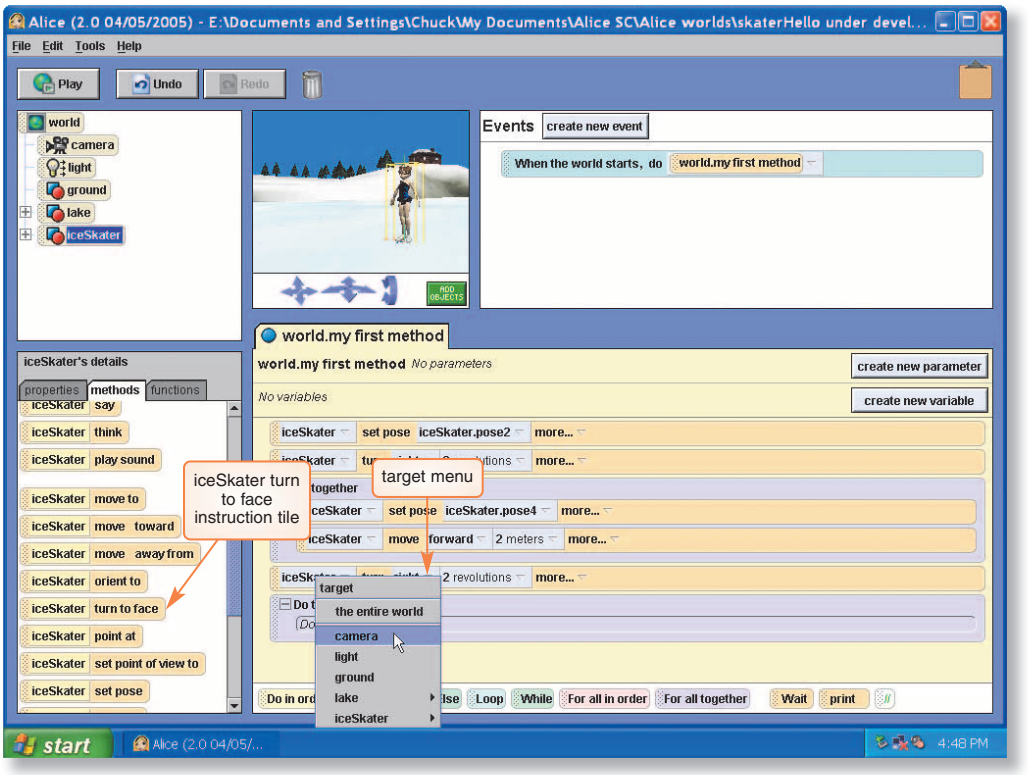

FIGURE 1-65

3

• **Click** camera **to select it as the target that the ice skater should turn to face.**

An iceSkater turn to face camera instruction appears in the Do Together tile (Figure 1-66).

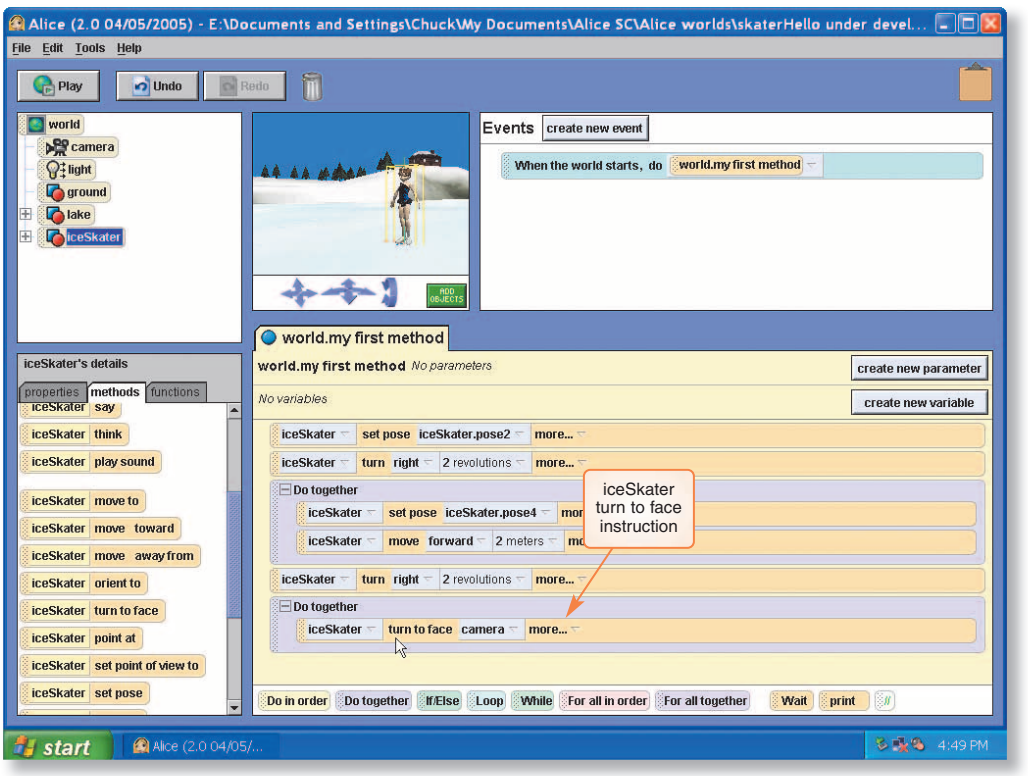

FIGURE 1-66

4

• **Click the** properties **tab in the Details area, then drag a copy of the** pose **tile and drop it in the** Do together **tile below the** iceSkater turn to face camera **instruction.**

The Do together tile now contains an instruction to set the iceSkater's pose to pose (Figure 1-67).

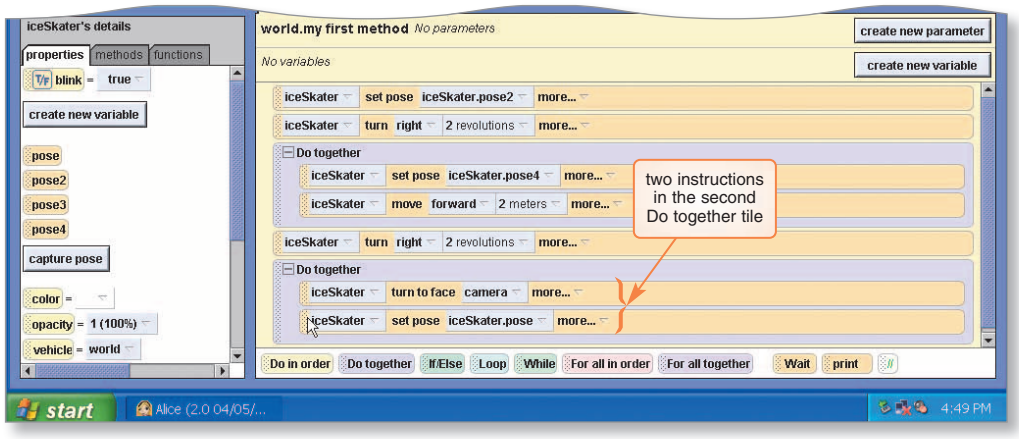

FIGURE 1-67

You only need one more instruction to finish the method. The iceSkater should say "Hello, World!" when she has finished her routine. By default, messages from the say command only stay on the screen for one second, so it will be necessary to adjust the instruction's duration parameter to make it stay on the screen longer. Parameters like distance and amount for the move and turn commands are almost always set by the programmer. Other parameters that exist but which are not often used are hidden and can be accessed using the more button, following the instruction's other parameters. Figure 1-68 shows the additional parameters for the say method that can be accessed with the more button. For now, you will change the duration to three seconds and ignore the other parameters. After you put the say instruction in place, you will check the world and save it. The following steps illustrate adding an instruction to make the ice skater say "Hello, World!"

FIGURE 1-68

To Add an Instruction to Make the iceSkater Say "Hello, World!

1

• **Click the** methods **tab in the Details area, then drag a copy of the** iceSkater say **tile and drop it onto** world.my first method **in the Editor area, below all of the existing instruction tiles.**

The what menu appears, asking you what the iceSkater should say (Figure 1-69).

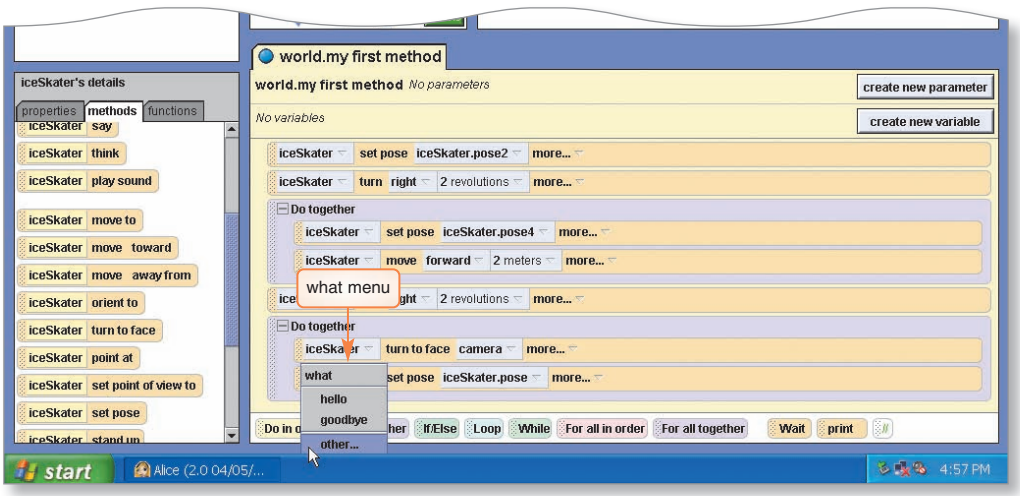

FIGURE 1-69

2

• **Click** other **to display the Enter a string dialog box.**

The Enter a string dialog box appears, as shown in Figure 1-70.

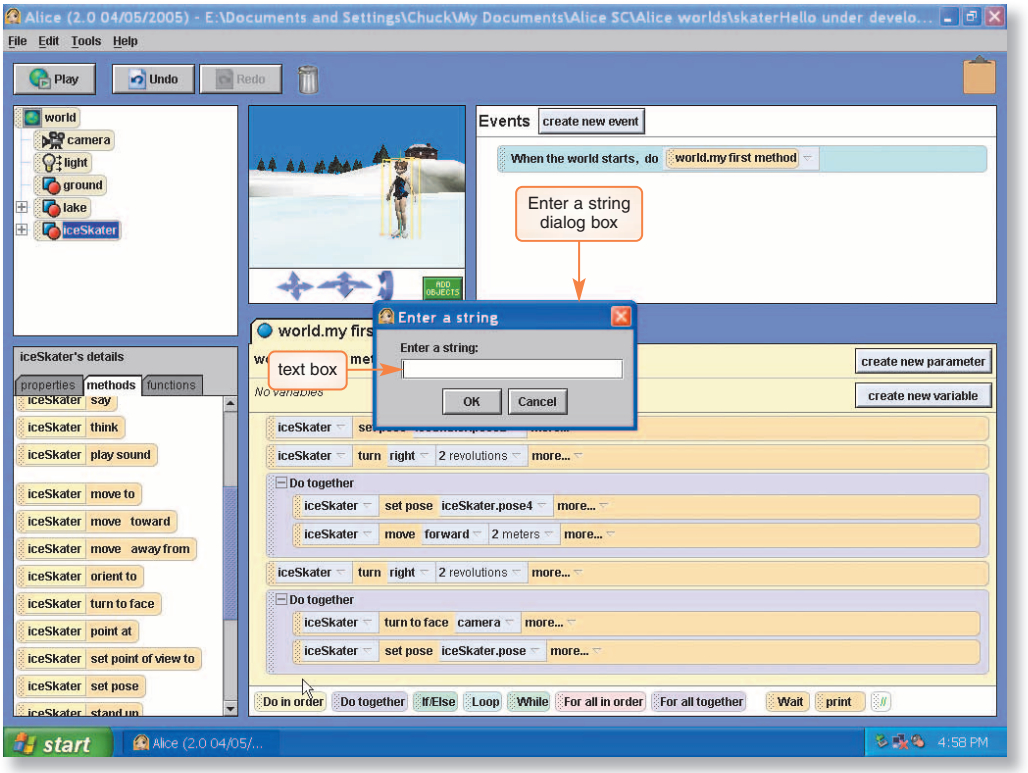

FIGURE 1-70

3

• **Type** `Hello, World!` **in the text box and then click the** OK **button to continue.**

An instruction to make the iceSkater say "Hello, World!" appears in the method. (Figure 1-71).

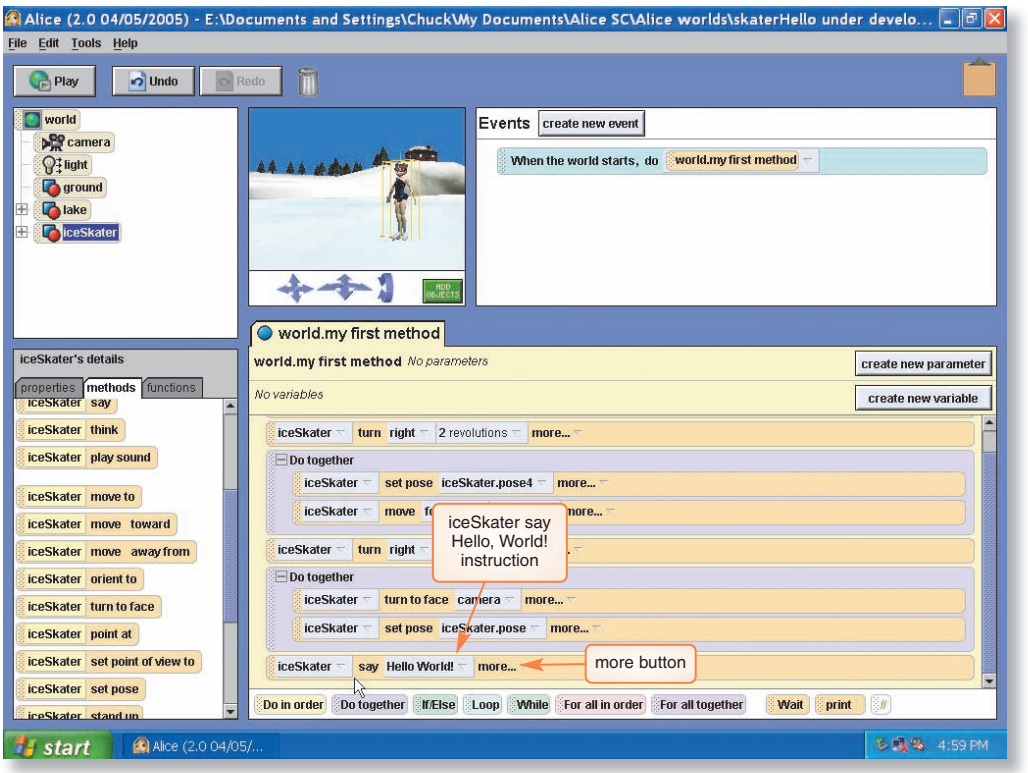

FIGURE 1-71

4

• **Click** more **on the** IceSkater say Hello, World! **instruction tile.**

A menu of additional parameters is displayed, as shown in Figure 1-72.

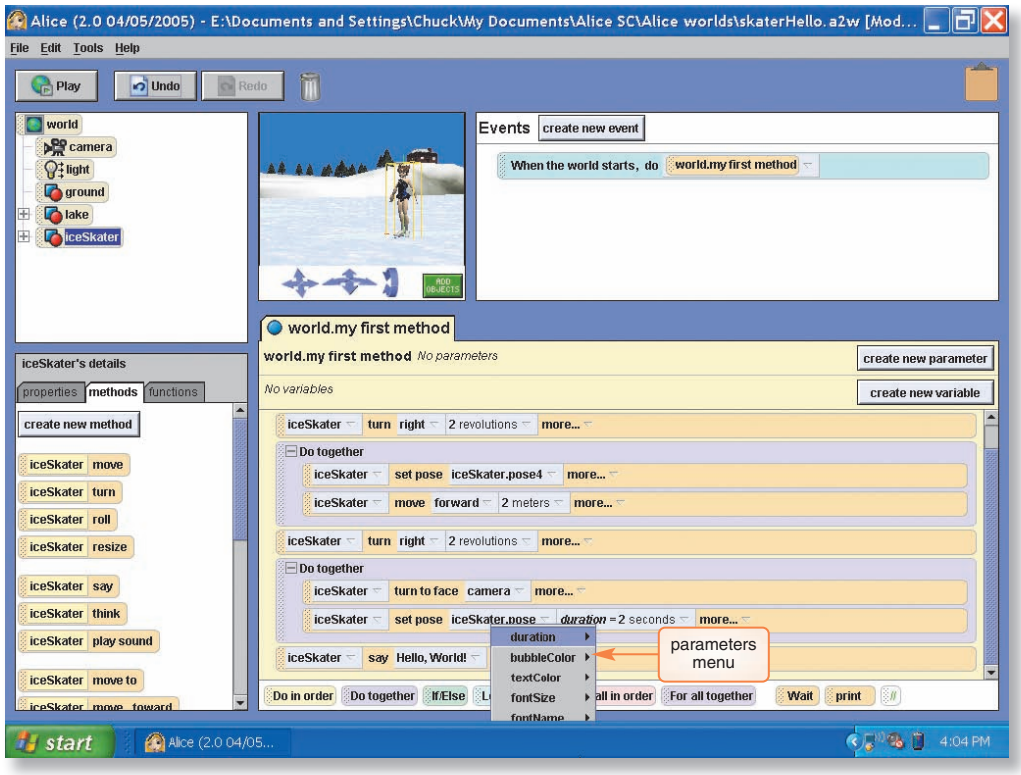

FIGURE 1-72

5

• **Point to** duration **to open the** duration **menu. If 3 seconds is not an option on the menu, then click** other **and use the calculator-style keypad to enter** 3 **as the amount. Click the** Okay **button.**

world.my first method is now complete, as shown in Figure 1-73. Before performing any other tasks, you should save your world.

6

• **Click** File **on the menu bar, and then click** Save World **on the File menu to save your world with its existing file name, skaterHello. As before, it will take a few seconds for Alice to save the world.**

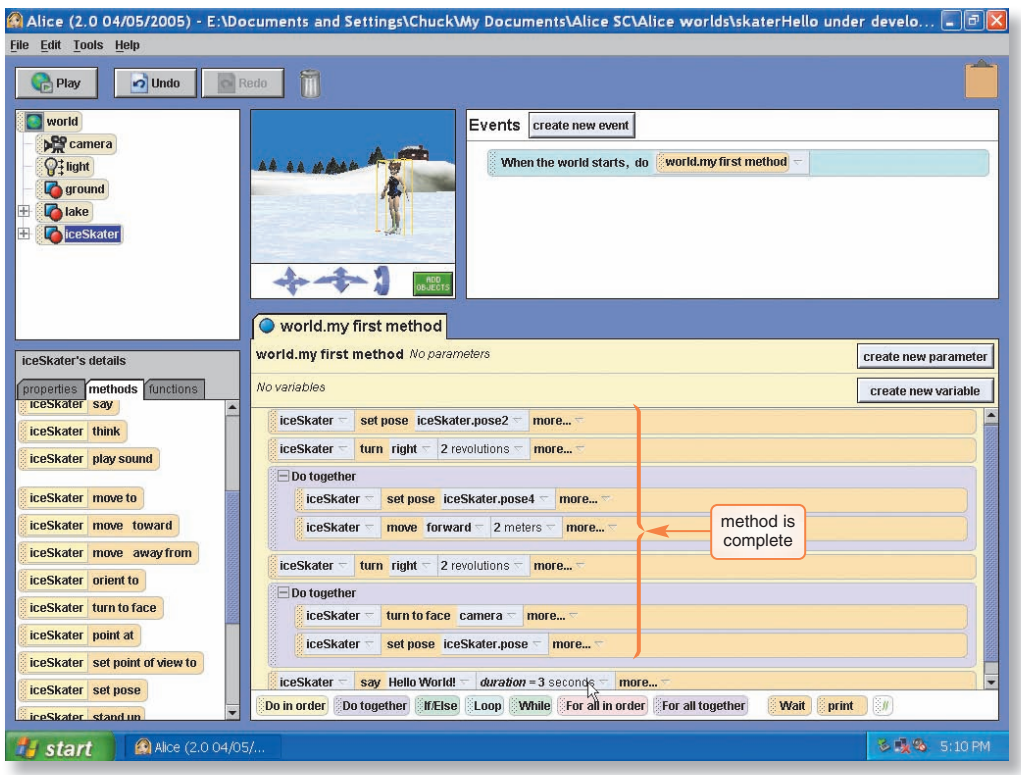

FIGURE 1-73

Testing new methods will be discussed in the next project, so for now, simply check to see that your new method matches the one shown in Figure 1-73 and save the world again before continuing if you make any changes.

In the following steps, you try the new method to see how it works.

To Play the skaterHello World

1

• **Click the** Play **button in the top left corner of the Alice interface to play the world, and see how it works.**

When the world is played, the skater's actions should follow the sequence of those shown in Figure 1-74.

2

• **Click the** Stop **button when finished playing the world.**

The world stops playing and the World Running window closes.

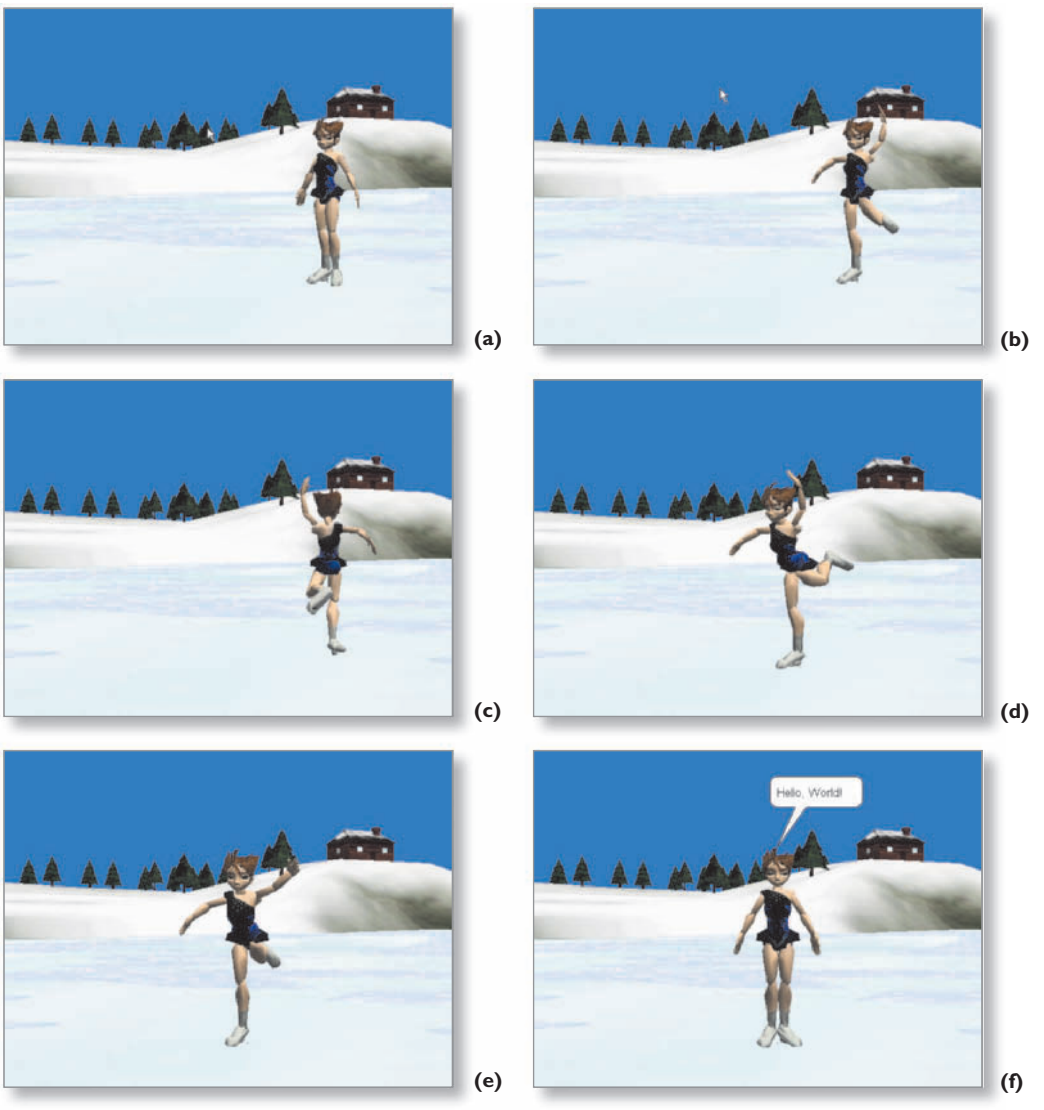

FIGURE 1-74

You may wish to play the world again and experiment with the Restart, Pause, and Resume buttons. See if you can match the actions on the screen with the instructions in world.my first method.

Printing the Code for an Alice World

The code for methods and events in Alice worlds can be exported to a Web page, using the Export Code for Printing feature on Alice's File menu. The resulting Web page can then be viewed and printed using any standard HTML Web browser, such as Internet Explorer, Mozilla Firefox, or Apple Safari. In addition, the HTML file with code from an Alice world can easily be sent to someone as an e-mail attachment, or used in almost any other way that an HTML page can be used. Figure 1-75 shows a Web page with the code from world.my first method in the world created during the previous section of this project.

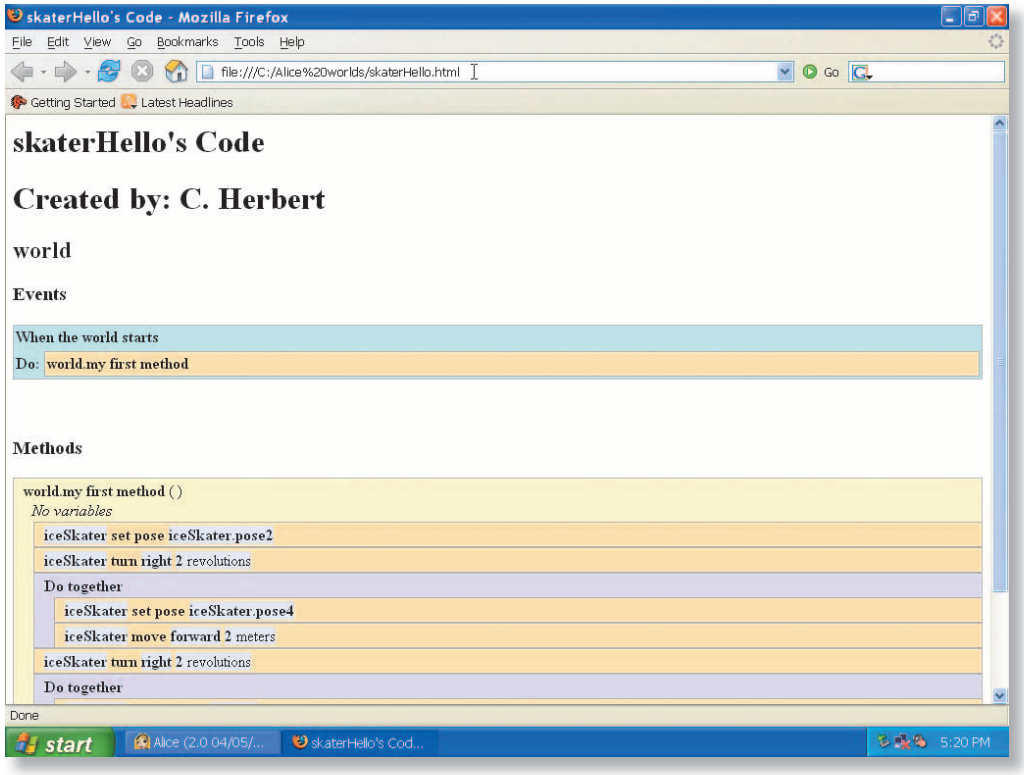

FIGURE 1-75

The Export to HTML dialog box will appear as you start to export your code to an HTML page, as shown in Figure 1-76. Most of the items in the Export to HTML dialog box are straightforward and easy to understand. You need to tell Alice what to export, where to export it, and the name of the author of the Alice world.

In the **What to export** section of the Export to HTML dialog box, you select what elements to export for printing. You can click the check boxes individually, or use the Select All or Deselect All buttons when selecting what methods and events to include on the Web page. By default, all methods and events will be selected. In skaterHello, the only method is world.my first method, and the only event is the default event. You will leave both selected in the steps below.

The Export to text box contains the full path name of the HTML file to be saved. By default, this will save your new Web page to the desktop, and the Web page will have the same name as the world, but with .html instead of .a2w as the file extension. Unless you have a good reason to do so, you should probably leave this as is. The Browse button can be used to select a different folder when you need to do so, or the new folder and file name can be entered in the Export to text box. You should leave the folder and file name as is in the following steps, unless your instructor tells you differently.

The Author's name text box is used to enter a name that will be displayed on the resulting Web page, as can be seen in Figure 1-75. In this case, you will type your own name, export the code to a Web page, then view and print the resulting page.

To Export Code from skaterHello.a2w to a Web Page

1

- **Code can be exported only from the currently open Alice world. If you have not closed the world since the steps above, then it should be ready to export. If it is not open, use the** Open World **command on the File menu to locate and open the world named** skaterHello.
- **Click** File **on the menu bar, and then click** Export Code For Printing.

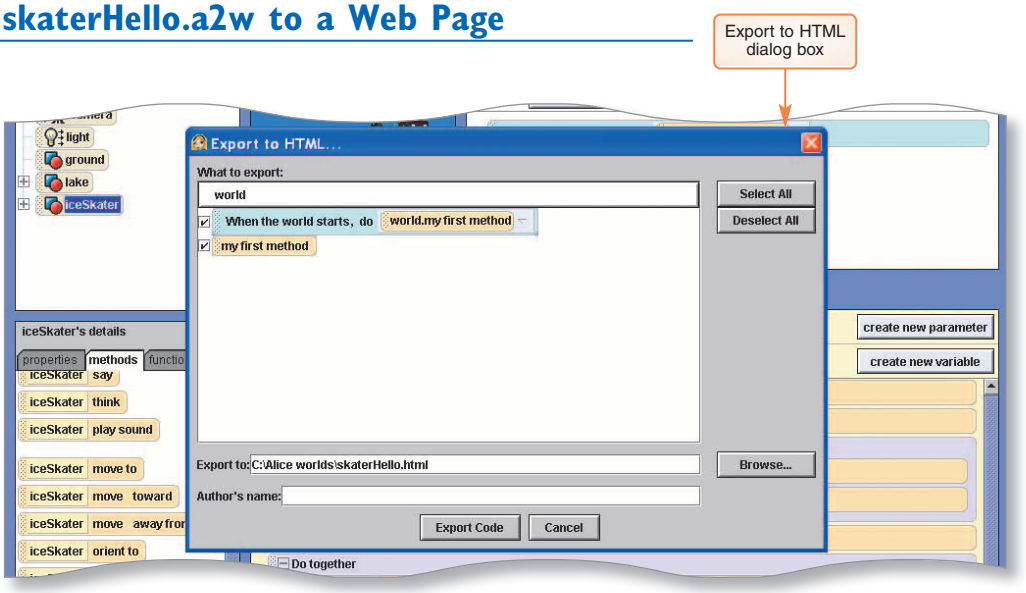

FIGURE 1-76

The Export to HTML dialog box opens, as shown in Figure 1-76.

2

- **In the** Author's name **text box, type your name as you would like it to appear on the Web page that will be created.**

Your name should be in the Author's name text box, as shown in Figure 1-77.

- **Leave the** Export to **text box as it is, unless your instructor has directed you to change the location for the resulting Web page. In either case, write down or remember the full path name that appears in the** Export to **text box.**

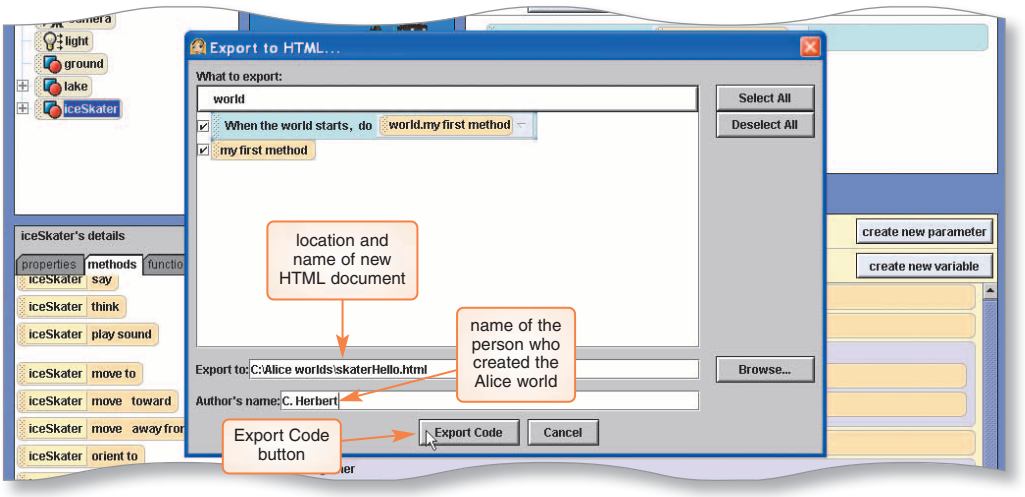

FIGURE 1-77

The full path name of the folder in which the new Web page will be stored is in the Export to text box (Figure 1-77).

3

• **Click the** Export Code **button to create a new Web page.**

A new Web page, containing the code shown in Figure 1-78, is saved in the folder you specified.

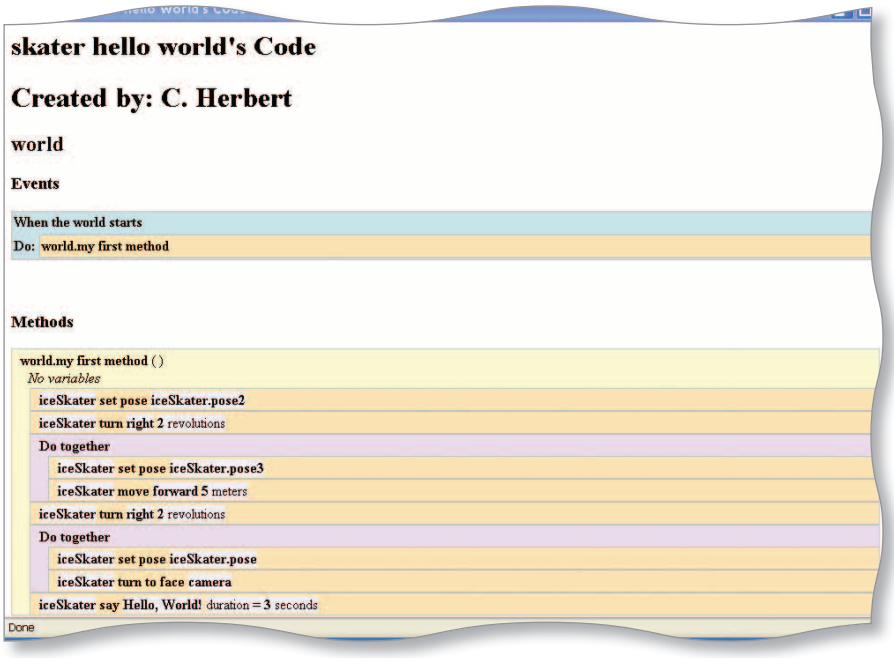

FIGURE 1-78

The following steps illustrate opening the HTML file for viewing and printing.

To Print the Code from the skaterHello Web Page

1

• **Close Alice, and then, using Windows Explorer, open the folder that contains the HTML file that was just created.**

The folder containing the HTML file, similar to the one in Figure 1-79, should be visible on your screen.

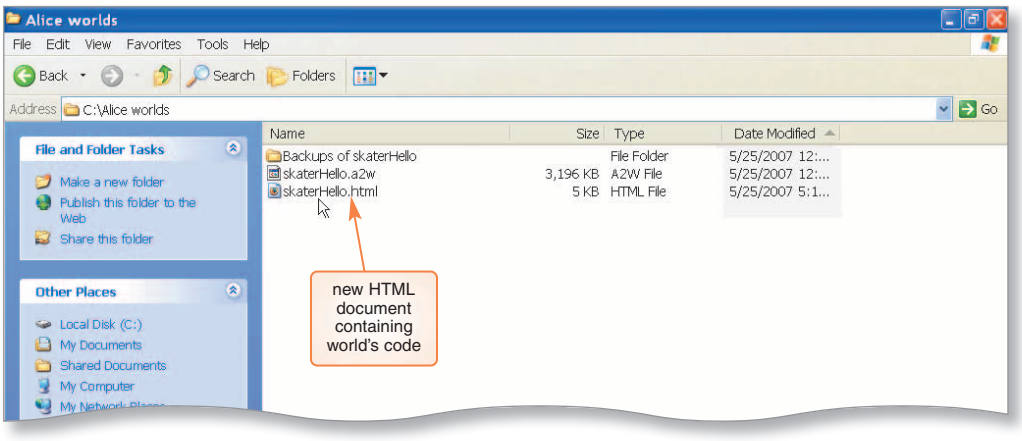

FIGURE 1-79

2

• **Double-click the** skaterHello.html **file to open it in your computer's default browser.**

The Web page with the code for skaterHello is visible on your screen.

3

• Use the Print command on your browser's File menu to print the page that shows the code for your new Alice world.

The Web page with the code for skaterHello prints at the printer.

Project Summary

In this project, you explored Alice and object-oriented programming. You started by exploring the Alice interface, including the five main work areas of the interface: the World window; the Object tree; the Details area, which contains the properties, methods, and functions tabs; the Editor area; and the Events area. You learned that the Alice Interface also has a Clipboard, a Trash can, Undo and Redo buttons, and a button to play an Alice world. You visited the four Alice menus: File, Edit, Tools, and Help, but learned that not all of them are used the way menus are in programs like Microsoft Word.

In the first part of the project, you learned that an object is a collection of properties and methods. You learned that properties contain information that describes an object, and that methods are computer programs to manipulate an object by changing its properties. The values stored in the properties of the object at any one time are called the state of the object. This modern approach to programming is called object-oriented programming.

You learned how to open and play an existing world, and how to start a new world using Alice's Scene Editor mode to add and position objects in the world. You also learned how to create code for methods in the new world. In the last part of the project, you learned to export the code from an Alice world to a Web page for printing.

What You Should Know

Having completed this project you should know how to:

1. Start Alice (AL 7)
2. Explore the Welcome to Alice! Dialog Box (AL 9)
3. Open the lakeSkater World and Explore the Alice Interface (AL 11)
4. View the Subparts of the iceSkater Object (AL 13)
5. View the Details of an Object (AL 14)
6. View the Details of a Method (AL 16)
7. Copy an Instruction Tile (AL 18)
8. Delete an Instruction Tile from a Method (AL 19)
9. Delete an Object (AL 20)
10. Undo and Redo Changes to an Alice World (AL 21)
11. Play the lakeSkater World (AL 24)
12. Exit and Restart Alice (AL 27)
13. Create a New Ice Skater World (AL 27)
14. Enter Scene Editor Mode (AL 28)
15. Add the Lake Object to the World (AL 29)
16. Add the iceSkater to the World (AL 30)
17. Save the Alice World (AL 33)
18. Position the iceSkater (AL 34)
19. Start Coding world.my first method (AL 36)
20. Add an Instruction to Make the iceSkater Spin (AL 37)
21. Add Instructions to Make the iceSkater Change Poses while Moving (AL 39)
22. Add an Instruction to Make the iceSkater Spin Again (AL 41)
23. Make the iceSkater Pose and Face the Camera at the Same Time (AL 41)
24. Add an Instruction to Make the iceSkater say "Hello, World!" (AL 43)
25. Play the skaterHello World (AL 46)
26. Export Code from an Alice World to a Web Page for Printing (AL 48)
27. Print the Code from the skaterHello Web Page (AL 49)

Apply Your Knowledge

1 An Animated Hello World Program

In this exercise you will apply the skills that you have learned to modify an existing world. The world already contains the word "Hello" as a 3-D text object in Alice. It spins faster and faster while rising off the screen. Your task is to add a bunny who will move onto the screen from the side and make a few comments.

Instructions: Start Alice. Open the file, Apply 1 Hello.a2w, from the Data Files for Students. See the inside back cover of this book for instructions for downloading the Data Files for Students or see your instructor for information about accessing the files required in this book. The word "Hello" is visible as 3-D text in the World window (Figure 1-80).

FIGURE 1-80

1. Click the Play button to see what happens in the existing world before continuing with the exercise. After the word "Hello" has left the screen, click the Stop button to return to the standard Alice interface.
2. Click the ADD OBJECTS button to enter Scene Editor mode.
3. Click the tile for the Animals folder, which will then open in the Alice gallery.
4. Click the Class Bunny tile to open the Bunny information dialog box, shown in Figure 1-81, then click the Add instance to world button to add a bunny object to the world.
5. You will now run three methods, one after the other, to position the bunny. First, right-click the bunny tile in the Object tree, point to methods on the menu that appears, point to bunny turn to face, and then click camera.

FIGURE 1-81

(continued)

Apply Your Knowledge

An Animated Hello World Program *(continued)*

6. Right-click the bunny tile in the Object tree, point to methods on the menu that appears, point to bunny turn, point to right, and then click 1/4 revolution.
7. Right-click the bunny tile in the Object tree once more, point to methods on the menu that appears, point to bunny move, point to backward, and then click 10 meters. The bunny is now off screen.
8. Using the Save World As command on the File menu, save the world with the file name, Apply 1 animatedHello.a2w.
9. Click the DONE button to return to the standard Alice interface. World.my first method is visible in the Editor area with the single instruction, Hello.whirled, in the method.
10. Click the bunny tile in the Object tree. Drag a copy of the bunny move instruction tile from the Details area and drop it into the method below the Hello.whirled tile. Select forward on the menu that appears, then click 10 meters.
11. Click the more parameter in the bunny move forward tile, point to duration, and then select 2 seconds as the value for duration.
12. Drag a copy of the bunny turn instruction tile from the Details area and drop it into the method below the bunny move forward tile. Select left on the menu that appears, and then click 1/4 revolution.
13. Drag a copy of the bunny say instruction tile from the Details area and drop it into the method below the bunny turn tile. Select other on the menu that appears, type `That was the famous "Hello Whirled" program.` in the Enter a string dialog box that appears, and then click the OK button.
14. Click the more parameter in the bunny say instruction tile, point to duration, and then click other on the menu that appears. Type the value 4 in the Custom Number dialog box, then click Okay.
15. Click the Hello tile in the Object tree, then drag a copy of the Hello move to instruction tile from the Details area and drop it into the method below the bunny say instruction tile. Make sure to drag the move to instruction and not the move instruction. Select bunny, and then select the entire bunny from the menus that appear.
16. Click the more parameter in the Hello move to bunny tile, point to duration, and then click 0.25 seconds on the menu that appears.
17. Click the bunny tile in the Object tree, then drag a copy of the bunny turn instruction tile from the Details area and drop it into the method below the Hello move to bunny instruction tile. Select backward on the menu that appears, then click 1/4 revolution.
18. Click the more parameter in the bunny turn backward tile, point to duration, then click the value 0.25 seconds on the menu that appears.
19. Drag a copy of the bunny say instruction tile from the Details area and drop it into the method below the bunny turn backward tile. Select other on the menu that appears, type `I don't write this stuff, I just work here.` in the Enter a string dialog box that appears, and then click the OK button.
20. Click the more parameter in the bunny say tile, point to duration on the menu that appears, and then select 4 seconds as the value for duration.
21. Your new world is now complete. Save it again with the Save World command on the File menu before playing the world.

In the Lab

1 Greetings from a Japanese Fan Dancer

Your task is to create a world in which a Japanese fan dancer will perform a short routine and then say "Welcome to the world of Alice."

Instructions:

1. Open an Alice world with a grass template.
2. Click the ADD OBJECTS button to enter Scene Editor mode.
3. Find and click the Japan tile in the Alice object gallery.
4. Find and click the Class Dojo tile, and then click the Add instance to world button to add a Dojo object to your world as a background for the dancer.
5. Find and click the Class FanDancer tile, and then click the Add instance to world button to add a fanDancer to your world.
6. Use the pointer tool to move the fanDancer back and to the right so that she is in the location shown in Figure 1-82, about halfway between the center and right edge of the World window.

FIGURE 1-82

7. Use the Turn tool to turn the fanDancer so that she is facing diagonally to her right, as shown in Figure 1-82.
8. Click the DONE button to exit Scene Editor mode.
9. Use Save World As on the File menu to save the world with the filename, Lab 1-1 fanDancer greeting.
10. Make sure that the fanDancer is selected in the Object tree and that the methods tab is selected in the Details area.
11. Drag a copy of the fanDancer turn tile from the methods tab and drop it in world.my first method in the Editor area.
12. Point to left on the direction menu that appears, then click 2 revolutions on the amount menu.
13. Drag a Do together tile from the bottom of the Editor area and drop it in world.my first method below the fanDancer turn instruction.

(continued)

In the Lab

Greetings from a Japanese Fan Dancer *(continued)*

14. Drag a copy of the fanDancer move tile from the methods tab and drop it in the Do together tile in place of Do Nothing.
15. Point to right on the direction menu that appears, then click other on the amount menu.
16. Use the Custom Number dialog box to enter the number 3 as the amount, and then click the Okay button.
17. Drag a copy of the fanDancer turn to face tile from the methods tab and drop it in the Do together tile below the move instruction.
18. Select camera on the target menu that appears.
19. Drag a copy of the fanDancer turn tile from the methods tab and drop it in world.my first method below the Do together tile.
20. Point to left on the direction menu that appears, then click 2 revolutions on the amount menu.
21. Click the plus sign next to the fanDancer tile in the Object tree, and then click the upperBody tile that appears.
22. Drag a copy of the upperBody turn tile from the methods tab and drop it in world.my first method below the Do together tile.
23. Select forward and ¼ revolution on the menus that appear.
24. Right-click the fanDancer.upperBody turn instruction in world.my first method and select make copy on the menu that appears. Be sure to right-click the tile and not one of the white parameter boxes in the tile.
25. There are now two copies of the fanDancer.upperBody turn instruction. Click the forward parameter in the bottom copy and select backward on the menu that appears.
26. Click the fanDancer tile in the Object tree to re-select the fanDancer.
27. Drag a copy of the fanDancer say tile from the methods tab and drop it into world.my first method below all of the existing instruction tiles.
28. Select other on the menu that appears.
29. Type "`Welcome to the world of Alice.`" in the Enter a string dialog box, then click the OK button.
30. Click more on the fanDancer say instruction tile and then select duration and 2 seconds on the menus that appear.
31. Your new fanDancer world is now complete. Save the world again before continuing. You may now play the world to see the fanDancer perform her routine.

2 A Penguin Demonstration

Problem: The penguin class of objects in the Local Gallery has built-in methods to make a penguin walk, jump up and down, flap its wings, and so on. Your task is to create a world that can be used to show people what some of the more useful built-in methods do. You should create a new Alice world in which a penguin will announce what action it is going to perform, and then perform the action. For example, the penguin would say "I am now going to flap my wings," then it would flap its wings, as shown in Figure 1-83.

In the Lab

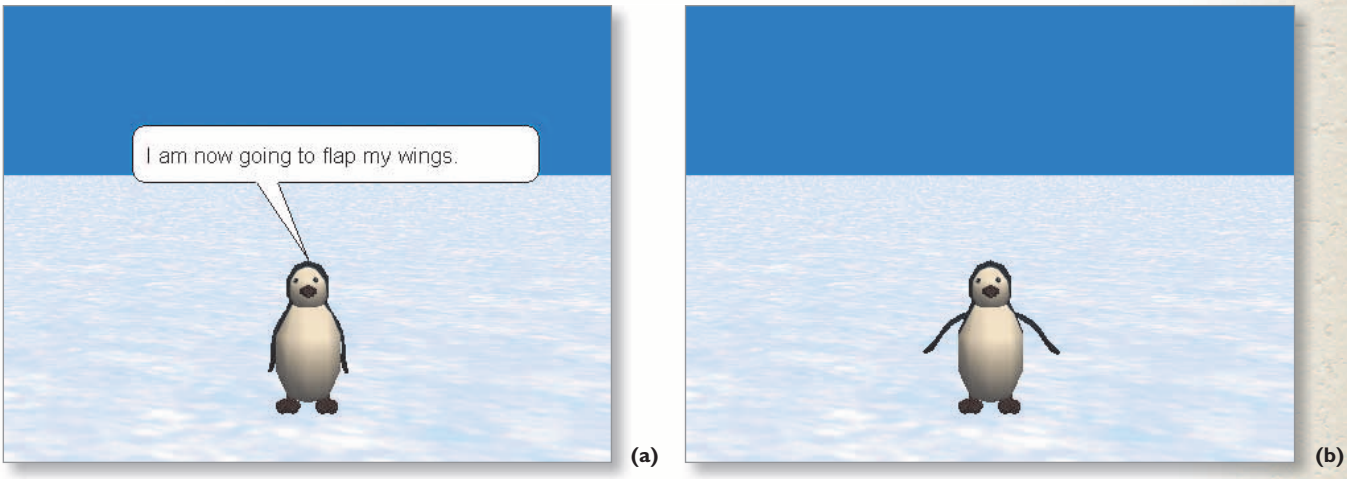

FIGURE 1-83

Instructions:

1. Start Alice and begin a new world with a snow template.
2. Enter Scene Editor mode and add a penguin to the world from the Animals folder. The tile for the penguin class of objects is shown in Figure 1-84.

FIGURE 1-84

3. Place the penguin in the middle of the screen, facing the camera. Click the DONE button to exit Scene Editor mode.
4. Click the penguin tile in the object tree. You will see several user-defined methods on the methods tab in the Details area. You will start with the wing_flap method.
5. First, make the penguin say "I am now going to flap my wings." The duration for the instruction should be 2 seconds.
6. Next, the penguin should actually flap its wings. Use the wing_flap tile from the methods tab to create an instruction to make the penguin flap its wings two times.

(continued)

In the Lab

A Penguin Demonstration *(continued)*

7. Following this pattern, have the penguin demonstrate some of the other things that it can do. The Penguin class built-in user methods are shown on the Penguin class tile in Figure 1-84.
8. By changing the penguin's color property in a manner similar to the way you changed the IceSkater's pose property during Project 1, you can make the penguin demonstrate that he can change color. For example, he could say "I am now going to turn blue," then do so.
9. The penguin should also execute several moves together at the same time. For example, the penguin could say "I am now going to jump up and down, flap my wings, and spin around all at the same time." You will need to use a Do together tile and adjust the instructions' parameters to coordinate such actions.
10. Finally, the penguin should turn to face the camera, say "Now, I'm leaving," and then turn to the side and glide out of the scene. You can use the penguin glide method to make the penguin glide out of the scene.
11. When you are finished creating the code for the world, save your world with the filename, Lab 1-2 penguin methods.a2w before playing the world.

3 American Chicken Farmers Association

Problem: The American Chicken Farmers Association needs your help. For several months, the Amalgamated Beef Producers of America have been running a television advertising campaign that has been cutting into the profits of chicken farmers. They want you to create an animated advertisement to respond to the Beef Producers' ads. Their catch phrase is "Don't even think beef."

The American Chicken Farmers Association advertisement will feature a cow that walks to the middle of the screen, turns its head to face the camera, and then says the following, one at a time: "Chicken is healthy.", "Real healthy.", "And tasty, too." The cow will then turn its head to face forward, start to walk off the screen, then stop, look at the camera again, and say "Don't even think beef." Finally, it will turn its head to face forward again and continue to walk off the screen.

Instructions:

1. Start Alice and begin a new world with a grass template.
2. Enter Scene Editor mode and add a cow to the world from the Animals folder.
3. Turn the cow so that it is perpendicular to the camera, facing to the left as seen by the camera, as shown in Figure 1-85.
4. Using the blue camera control arrows, move the camera to the left just enough so that the cow is off screen on the right, then exit Scene Editor mode and return to the standard Alice interface.

FIGURE 1-85

In the Lab

5. Click the cow tile in the Object tree to select the cow. Several user-defined methods are visible for the cow in the Details area, including walk, walkTowards, and tailSwish.
6. You will need to create the animation sequence by dragging instruction tiles from the methods tab into world.my first method in the Editor area. You will need the following 15 instructions:

   cow.walk times = 10 speed = 2
   cow.tailSwish times = 2 speed = 2
   cow.neck turn left 1/4 revolution
   cow say "Chicken is healthy." duration = 2 seconds
   cow say "Real healthy." duration = 2 seconds
   cow say "And tasty, too." duration = 2 seconds
   cow.neck turn right 1/4 revolution
   cow.tailSwish times = 2 speed = 2
   cow.walk times = 2 speed = 2
   cow.tailSwish times = 2 speed = 2
   cow.neck turn left 1/4 revolution
   cow say "Don't even think beef." duration = 2 seconds
   cow.neck turn right 1/4 revolution
   cow.tailSwish times = 2 speed = 2
   cow.walk times = 11 speed = 2

   Most of the instructions use the walk, say, or tailSwish tiles found on the cow's methods tab, so you simply need to drag those instructions into world.my first method in the proper order and set the parameters as indicated. Four of the instructions are turn instructions for the cow's neck, which is a subpart of the cow. For each of these, you will need to click the plus sign next to the cow tile in the Object tree to see the neck tile, then you can drag the neck turn instruction from the methods tab into world.my first method.
7. When you are finished creating the code, save your world with the file name, Lab 1-3 chicken is tasty.a2w, before playing the world.

Cases and Places

1 The Caribbean School of Computer Programming would like a 10-second video advertisement in the form of an animated virtual world. They wish to show a beach scene with a lighthouse, two beach houses, a pier, a sailboat, and several palm trees, as shown in Figure 1-86.

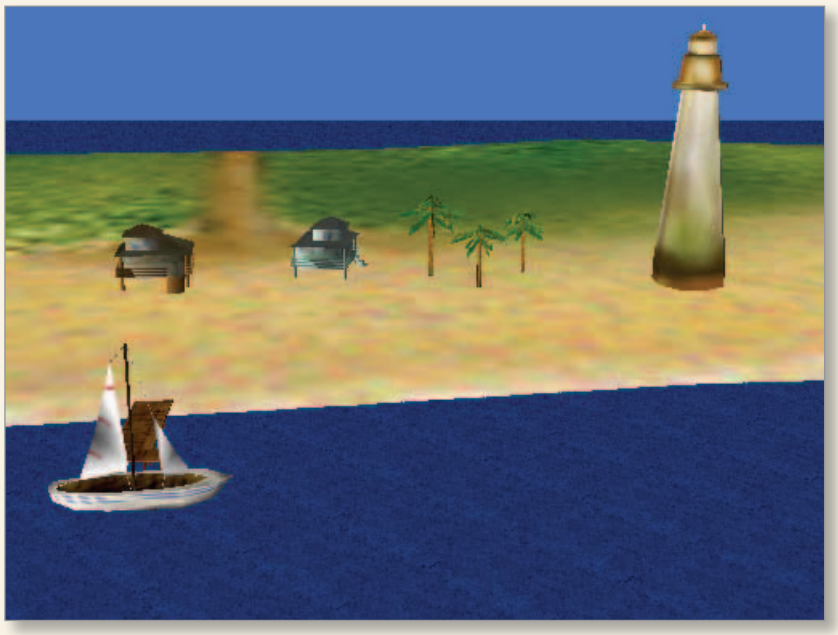

FIGURE 1-86

When the advertisement starts, the sailboat sails away, then two messages appear on the screen one at a time. One of the beach houses should say "Come to the Caribbean School of Computer Programming." Then the other should say "Where the summer semester never ends." Each message should stay on the screen for several seconds.

Your task is to add the necessary objects to an existing beach scene and then create the advertisement. You can find most of the items you will need in the **Beach** folder in Alice's local object gallery. The sailboat is in the **Vehicles** gallery and the palm tree is in the Nature folder. You can resize, turn, and rotate copies of the palm tree so that they do not all look the same.

You should begin by starting Alice and opening the file, Case 1-1 beach scene.a2w, from the Data Files for Students. See the inside back cover of this book for instructions for downloading the Data Files for Students or see your instructor for information about accessing the files required in this book. Once you have a properly working world, save it with the file name, Case 1-1 Caribbean School of Computer Programming.a2w.

Cases and Places

2 Daphne Belle, the Director of the Shelly Cashman Drama Society, has been trying to create an animation to show students the movement and dialogue for their upcoming production of the 1970's revival "Joe Cool!". Scene II, Act III opens with Mary standing in a room between classes. The door opens, Joe enters the room and, trying to be cool, says in French, "Ehhh! Mon Ami. Je t'adore." To which Mary says "Shut the door? Shut it yourself." She then walks away.

The problem is that a director like Daphne often is quite busy, and she hasn't finished creating the Alice animation to show her actors what to do. Your task is to finish working on the animation so that it functions as described above.

Start Alice and open the world Case 1-2 Joe Cool.a2w from the Data Files for Students. See the inside back cover of this book for instructions for downloading the Data Files for Students or see your instructor for information about accessing the files required in this book. Play the world to see how far Daphne got, decide what changes need to be made, then complete the world. When you are finished, save the world with the file name, Case 1-2 Joe Cool Revival.a2w.

3 Alice has two tools that you can use to build characters of your own for an Alice virtual world: the **hebuilder** and **shebuilder** classes of objects found in the **People** folder in the Alice Local Gallery. The new characters created with **hebuilder** and **shebuilder** have built-in, user-defined methods to walk, move, and show various emotions. Working together with several other students, create a simple Alice world with two characters created using **hebuilder** and **shebuilder** that engage in a short dialogue.

You should experiment with **hebuilder** and **shebuilder** before you start on your final world to become familiar with the different options available for new characters. You should also experiment with the various methods for the new characters to see how they function. Once you are familiar with the **hebuilder** and **shebuilder** characters, you can work as a team to plan and build your new world. You might want to start the planning process by sketching the sequence of events in your new world, either with a series of storyboards or with an outline. Storyboards can often give you a better feel for the movement and placement of objects, as well as camera angles, when planning a new Alice world.

Keep in mind McGinley's Rule for new programmers: K.I.S.S. – Keep It Short and Simple. Do not get too carried away planning a world that will be difficult and time consuming to implement. Start with just two characters, perhaps a few background objects, and a short and simple sequence of events.

When you are finished, save the new world with the file name, Case 1-3 new characters.a2w.

Learning Exercises

1 The URL for the official Alice Website is www.alice.org. The page contains links to many items about Alice. Visit the site and try the links to these or other features:

- **Alice Gallery.** An online gallery of objects that can be saved as data files with the extension "a2c" and then later imported into an Alice world. To save an object, click on the link for the desired object and save it to disk. Do not change its file name. To import a saved character, start Alice and open an existing world, or create a new world and then select Import from the Alice File menu.
- **Alice FAQ.** A large collection of answers to the most frequently asked questions by Alice users.
- **Export as a webpage tutorial.** A short tutorial on saving an Alice world as an interactive Web page.
- **Alice community.** A link to the Alice Community Forums, where users of Alice exchange questions, information, and ideas. You may read the messages posted on the forum without joining, but to post questions or comments to the site you will need to join. The forum is free, and is operated by faculty, staff, and students at Carnegie Mellon University. There are sections on the forum for announcements, students, and educators, and for sharing Alice objects and worlds.

2 The Alice software includes a set of built-in tutorials to introduce you to Alice. Open the Alice software and complete the first two tutorials. The third tutorial involves music, which requires a computer with a sound card.

3 Briefly describe how to complete each of the following Alice tasks:

A. view the subparts of an Alice object

B. view the properties of an Alice object

C. add an instance of an object to an Alice world

D. delete an object from an Alice world

E. delete an instruction tile from an Alice method

F. print the code for an Alice world