Getting Started with Java Using Alice

Get Started with Alice 3

Objectives

This lesson covers the following objectives:

- Identify scene components
- Create and save a new project
- Add an object to a scene
- Communicate the value of saving multiple versions of a scene
- Code a simple programming instruction
- Use the copy and undo command
- Understand the value of testing and debugging

Initial Scene

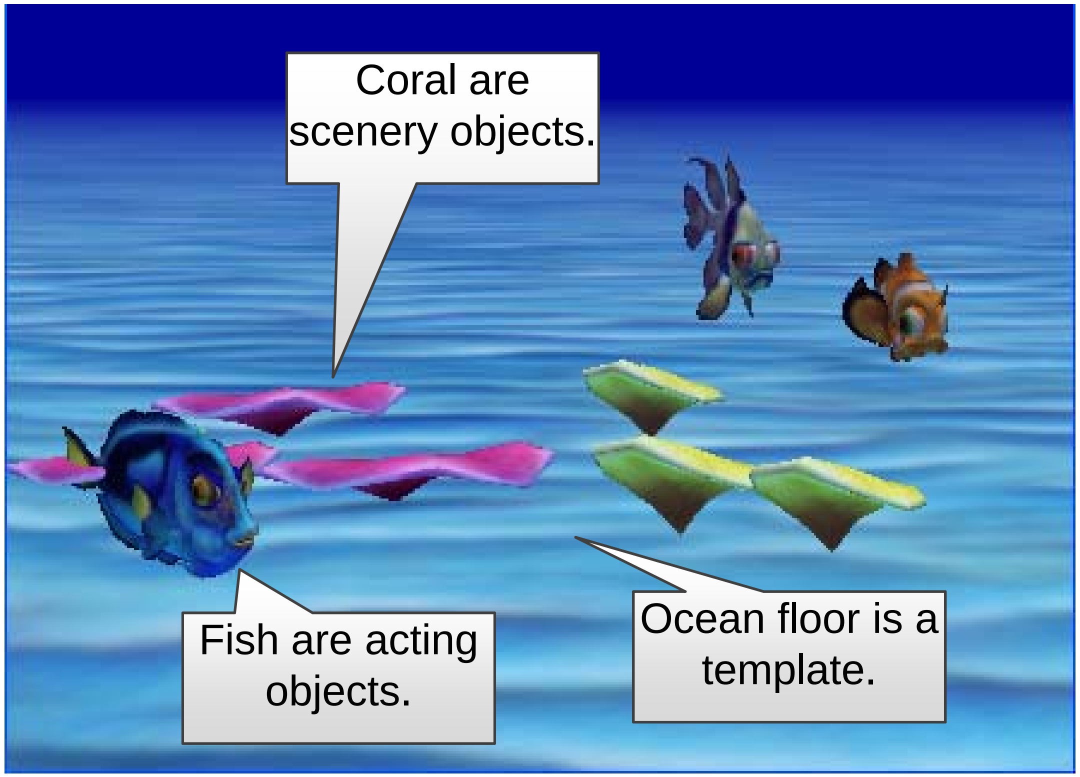
An initial scene is the starting point of your animation. It has three components:

- A background template which provides the sky, ground, and light.
- Non-moving scenery objects which provide the setting.
- Moving objects which provide the action.

The initial scene is the first scene of an animation where you select the background template and position the objects.

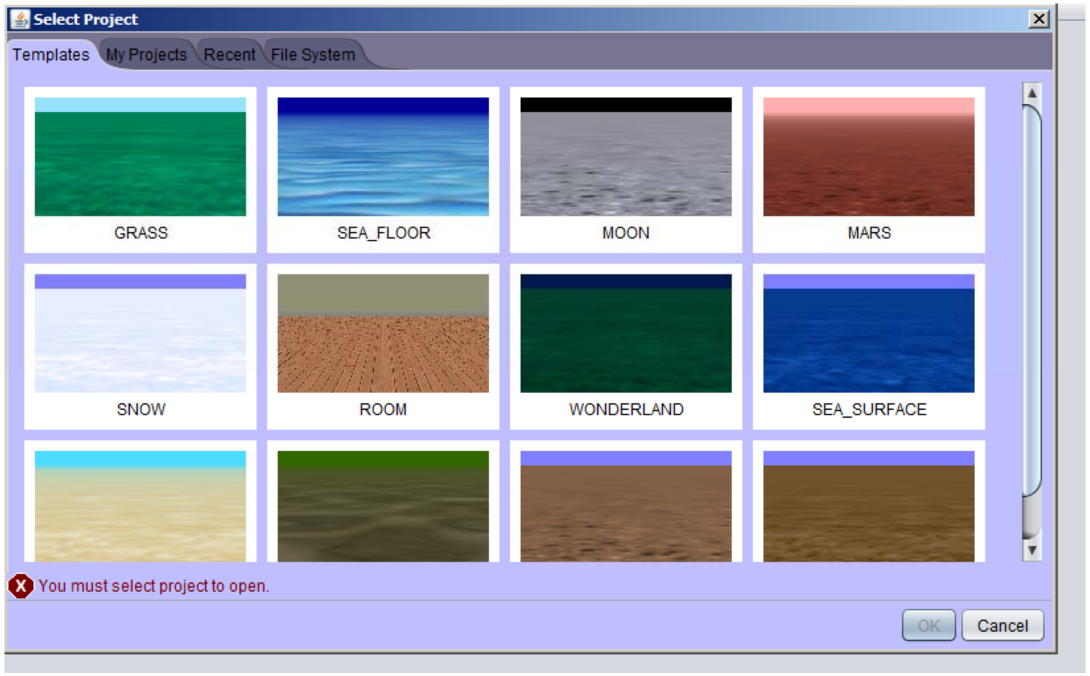
Initial Scene Components

Below are components of an ocean floor scene.



Steps to Create a New Project

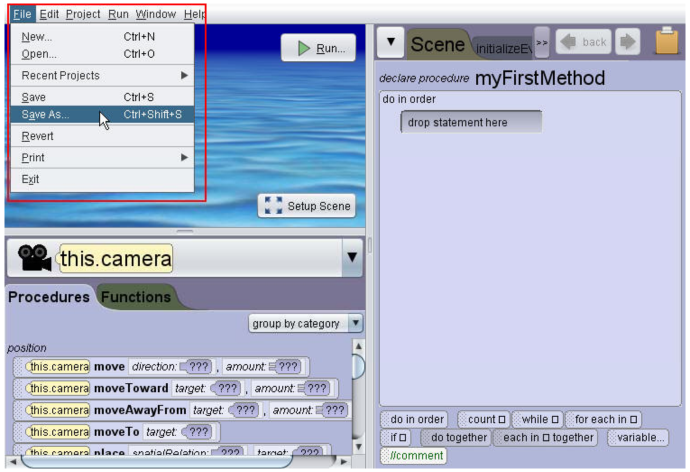
1. Launch Alice 3.
2. In the Welcome dialog box, select the Templates tab.
3. Select a template, and click OK.



You must select project to open.

Steps to Save a Project

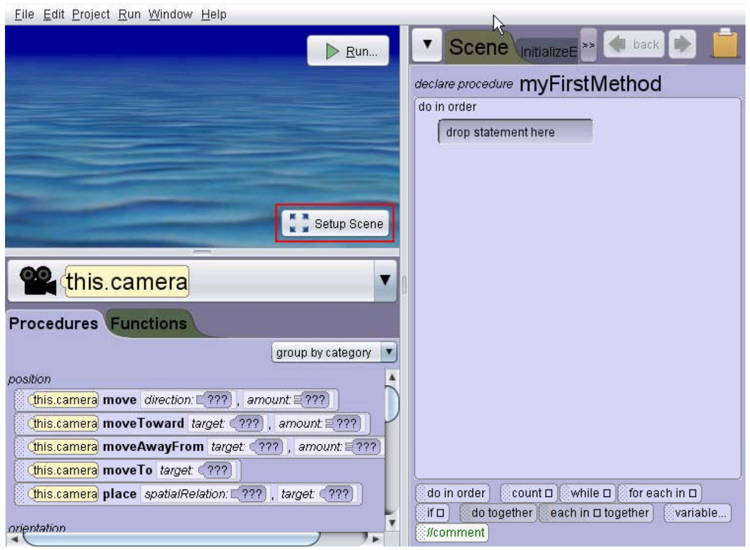
1. In the File menu, select Save As.
2. Select the location to save the project (e.g., computer, file server, memory stick).
3. Enter the project name.
4. Click Save.
5. Save projects frequently to avoid losing your work.

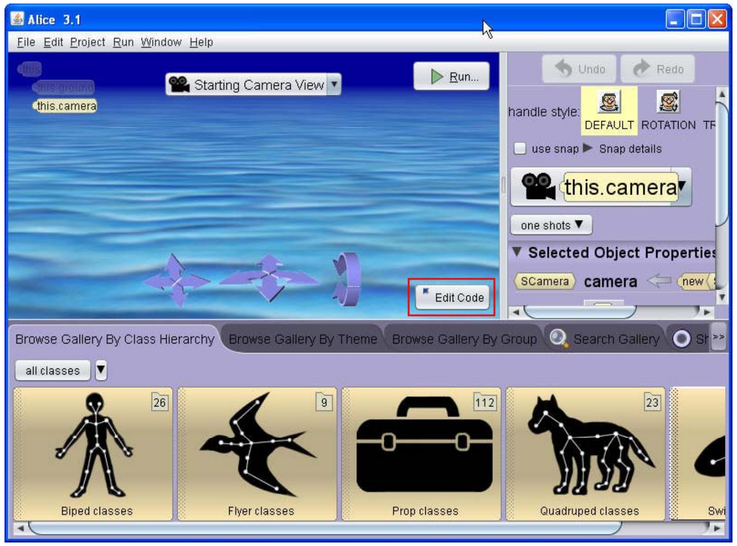


Navigating Between Editors

Alice provides two different workspace editors, called perspectives, that you will toggle between frequently as you build your project. The two editors are:

- Code editor (Edit Code perspective, shown on the left)
- Scene editor (Setup Scene perspective, on the right)

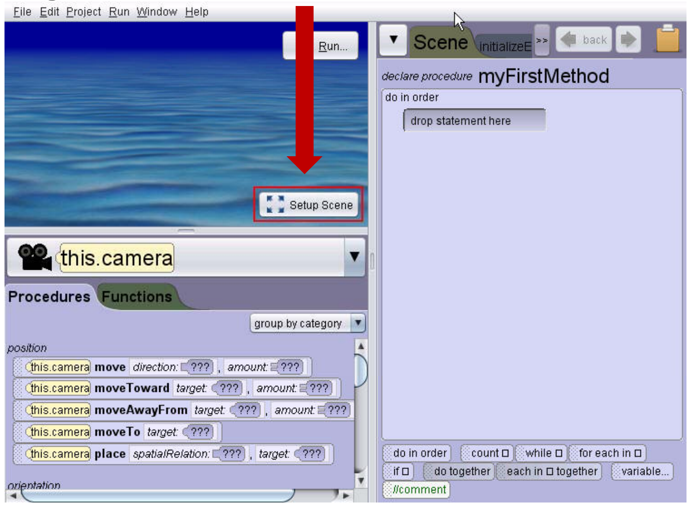

File Edit Project Run Window Help
Run...
Setup Scene
this.camera
Procedures
Functions
group by category
position
this.camera move direction: ??? , amount: ???
this.camera moveToward target: ??? , amount: ???
this.camera moveAwayFrom target: ??? , amount: ???
this.camera moveTo target: ???
this.camera place spatialRelation: ??? , target: ???
orientation
Scene
initializeE
back
declare procedure myFirstMethod
do in order
drop statement here
do in order
count
while
for each in
if
do together
each in together
variable...
//comment

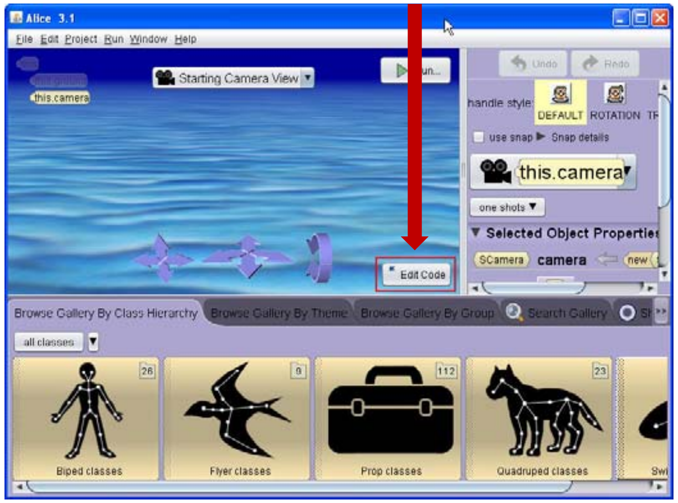


Navigating Between Editors

Switch between the two editors by using either the Edit Code button or Setup Scene button.

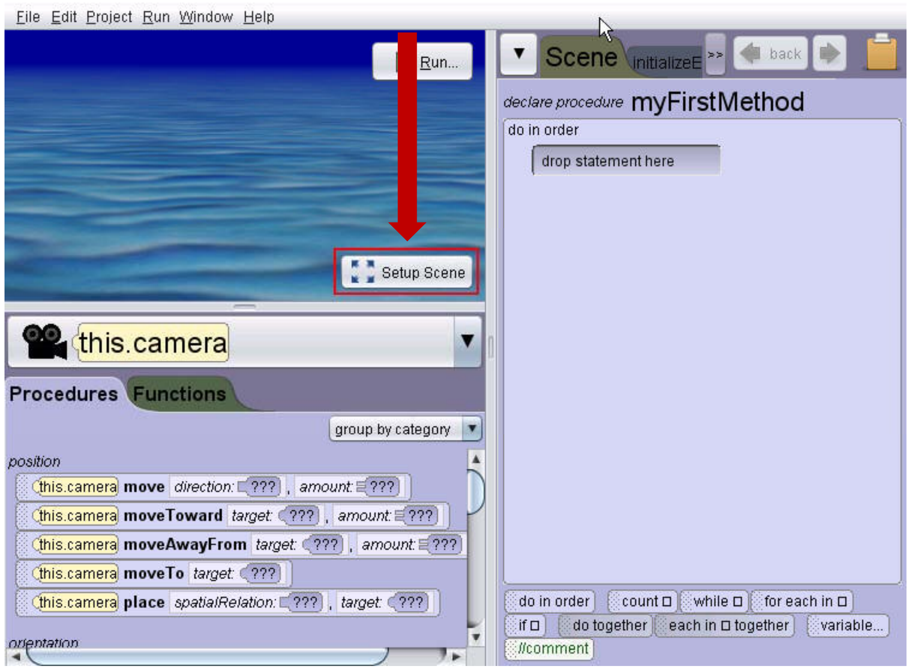
- You are in the Code editor when you see programming instructions in the left window pane.
- You are in the Scene editor when you see the gallery of objects.





The Default Editor

By default, Alice starts in the Code editor. Click the Setup Scene button to switch to the Scene editor.


File Edit Project Run Window Help
Run...
Setup Scene
this.camera
Procedures
Functions
group by category
position
this.camera move direction: ??? , amount: ???
this.camera moveToward target: ??? , amount: ???
this.camera moveAwayFrom target: ??? , amount: ???
this.camera moveTo target: ???
this.camera place spatialRelation: ??? , target: ???
orientation
Scene
initializeE
back
declare procedure myFirstMethod
do in order
drop statement here
do in order
count
while
for each in
if
do together
each in together
variable...
//comment

Add an Object to a Scene

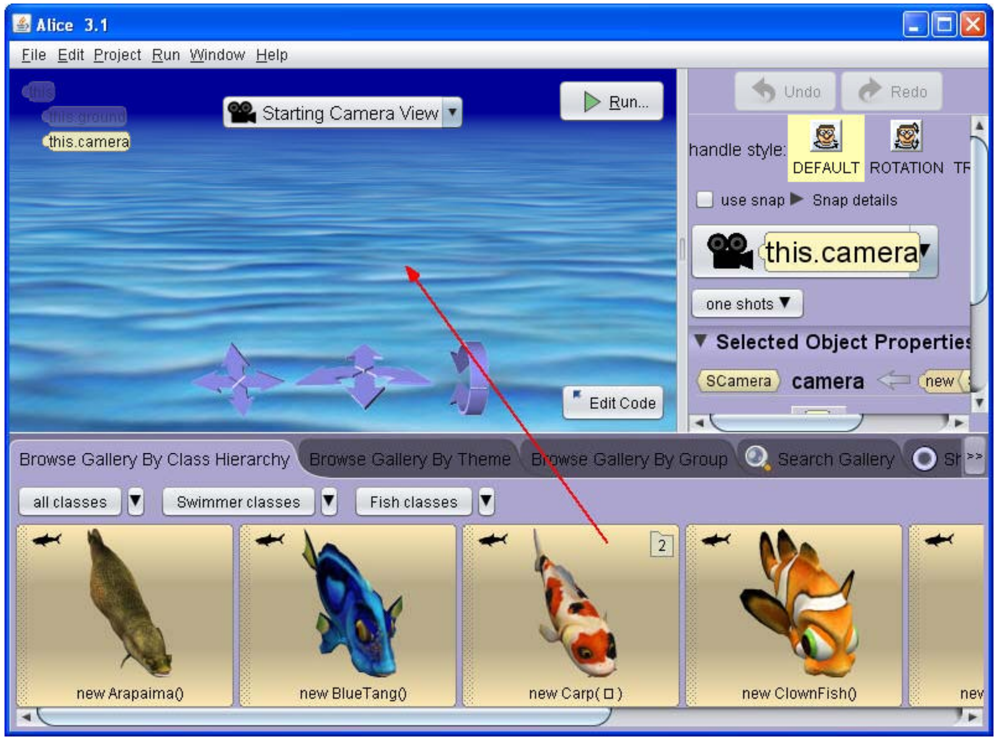
In the Scene editor, add an object (instance of a class) to a scene in one of two ways:

- Drag an object from the gallery into the scene with your mouse.
- Click the object, complete the dialog box, and let Alice 3 add the object to the center of the scene.

In programming terms, a class is a blueprint used to build an object, and an object is an instance of a class. After an object is added to a scene, it is referred to as an instance of the object. You can add many instances of the same object to a scene (multiple coral objects in the water, for example). Each instance must have a unique name.

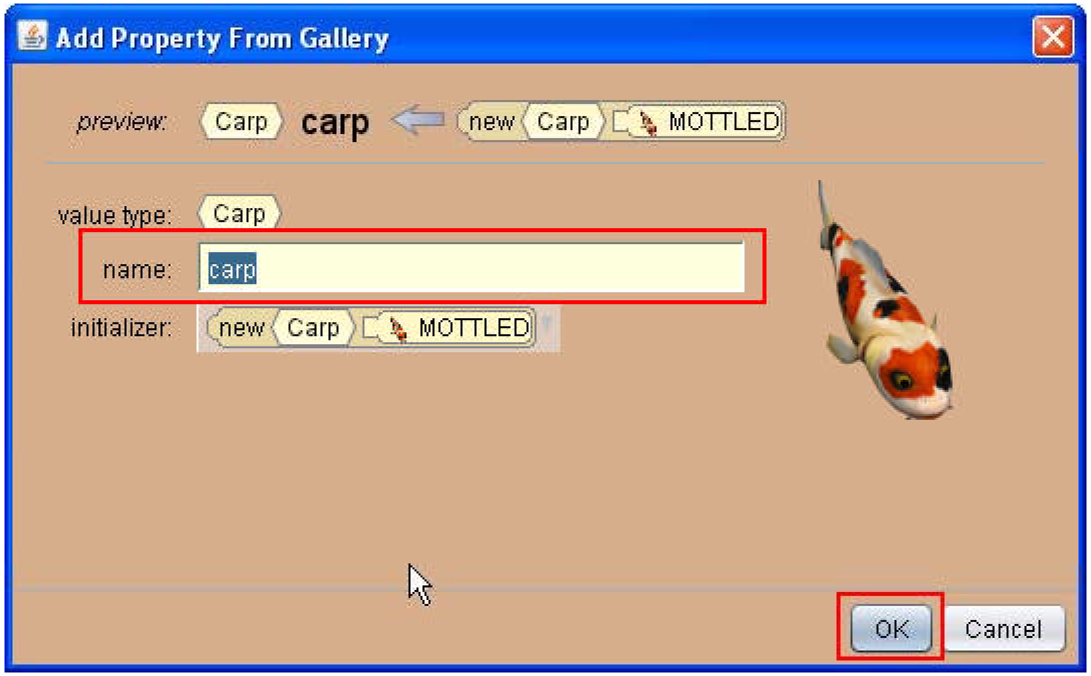
Add an Object to a Scene Display

Click the object once, or drag the object from the gallery into the scene with your mouse.



Naming the Object

Review the name provided for the object. Modify the name, or click OK to accept the name and add the instance to the scene.


Add Property From Gallery

preview: Carp carp ⇐ new Carp MOTTLED

value type: Carp

name: carp

initializer: new Carp MOTTLED

Scene Editor

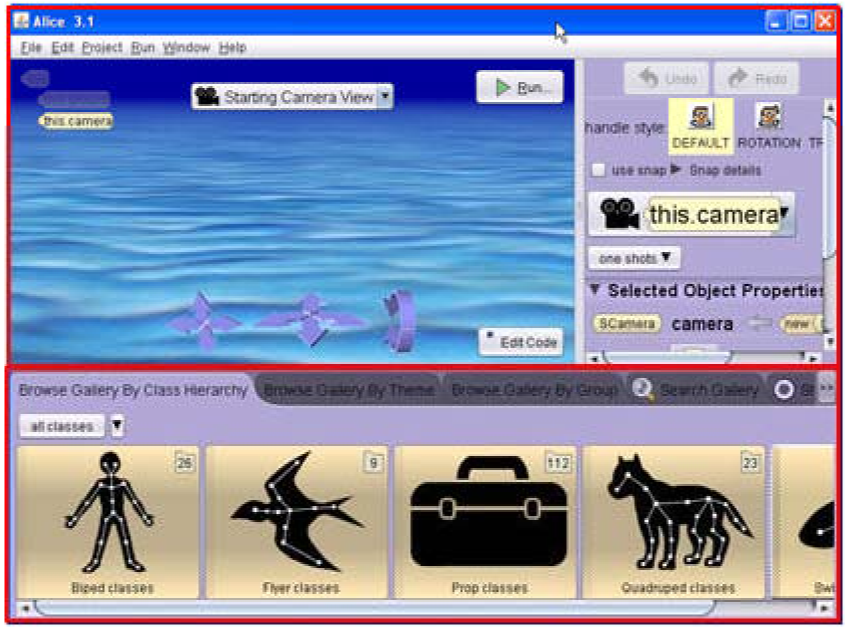
In the Scene editor, you can:

- Select objects from the gallery to add to the scene.
- Position objects in the scene using the Handles palette.
- Edit an object's properties using the Properties panel.
- Access the Code editor to add programming statements.
- Run the animation after the programming statements are added to the Code editor.

Scene Editor Display

The Scene editor contains two panels:

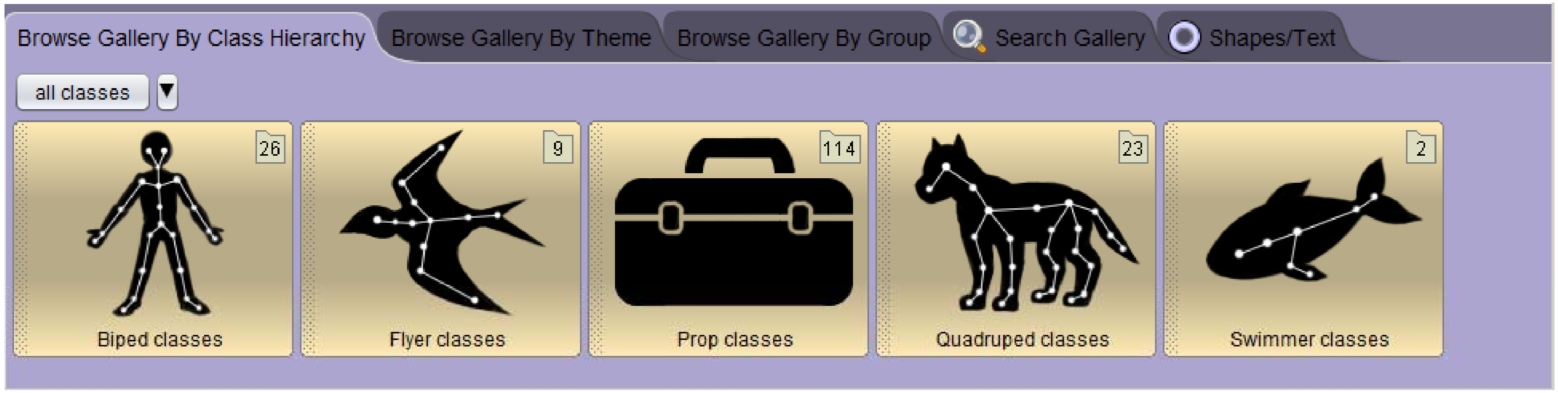
- Scene Setup at the top
- Gallery at the bottom



Gallery

The gallery is a collection of three-dimensional objects that you can insert into the scene.

- The gallery is organized using tabs.
- To find objects, browse the gallery tabs or use the Search Gallery feature to search by keyword.
- Breadcrumb menus display as you select classes.



Gallery Tabs

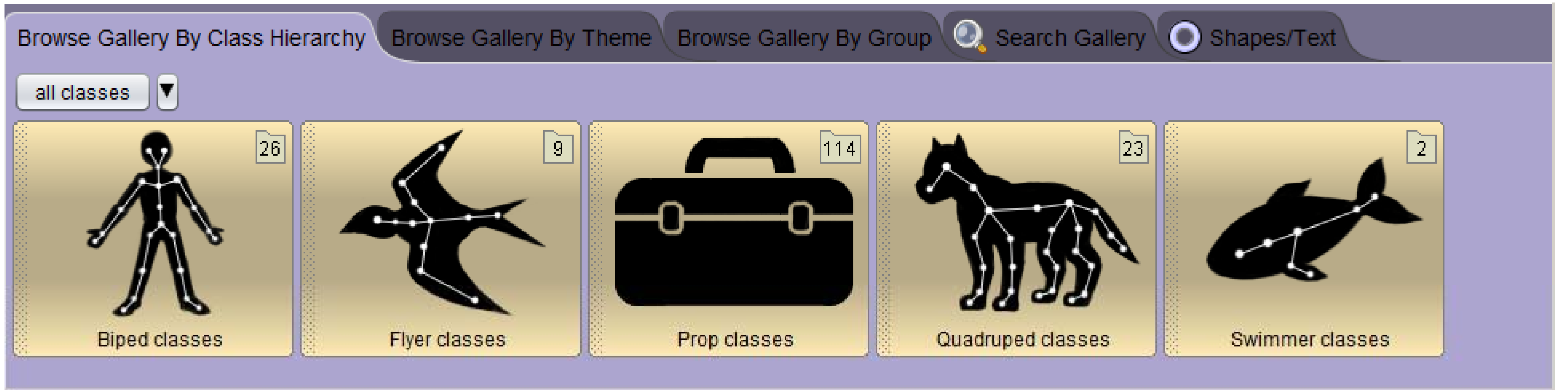
The gallery has five tabs:

Tab	Function
Browse Gallery by Class Hierarchy	Organizes objects by mobility.
Browse Gallery by Theme	Organizes objects by region and folklore context.
Browse Gallery by Group	Organizes objects by categories.
Search Gallery	Allows an object search by name.
Shapes/Text	Organizes object shapes, 3D text, and the billboard.

Select a Class

The Class Hierarchy tab groups objects by mobility type (biped, flyer, etc.).

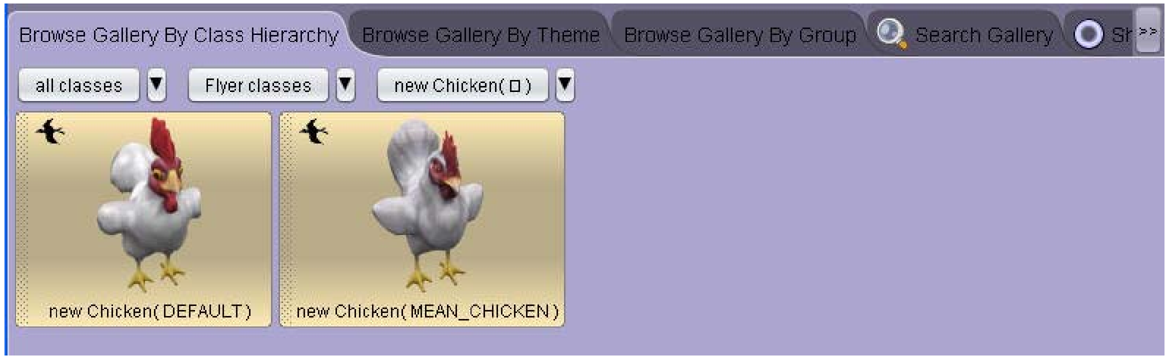
A class contains the instructions that define the appearance and movement of an object. All objects within a class have common properties. The class provides instructions to Alice 3 for creating and displaying the object when it is added to your scene.



Class Example

Classes can contain sub-classes. Consider the example below.

- The chicken class is a sub-class of the Flyer class. There are two chicken sub-classes in the gallery.
- Every chicken added to a scene inherits the properties that all Flyer objects have in common: two legs, two wings, the ability to fly, etc.



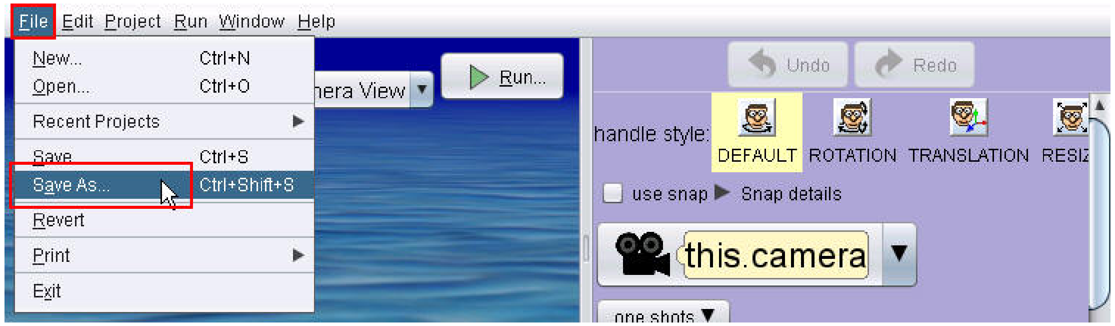
Save New Project Version

Save time by creating multiple versions of your project.

- After objects are positioned in the initial scene, save multiple versions of your project, giving each version a different name.
- Benefits of saving multiple versions of projects:
  - Use the same scene to create different animations.
  - Save time re-creating the scene if you encounter programming errors.

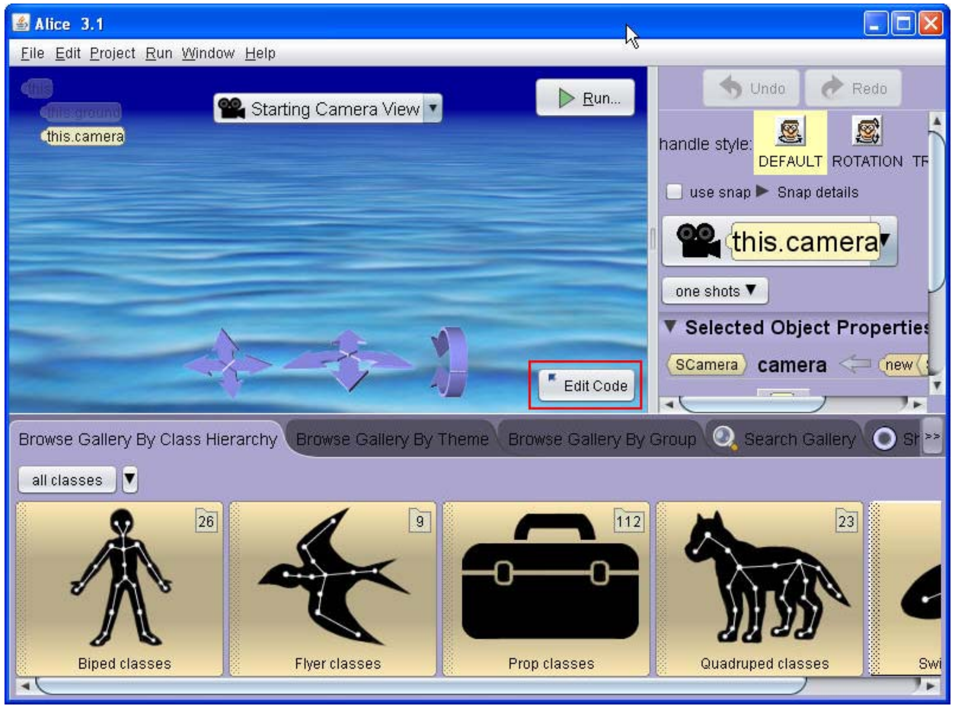
Steps to Save a Project Version

1. Select File.
2. Select *Save As...*
3. Select the location to save the project (e.g., computer, file server, memory stick).
4. Enter the project name.
5. Click Save.
6. Save projects frequently in case of a power failure or a computer crash.



Code Editor

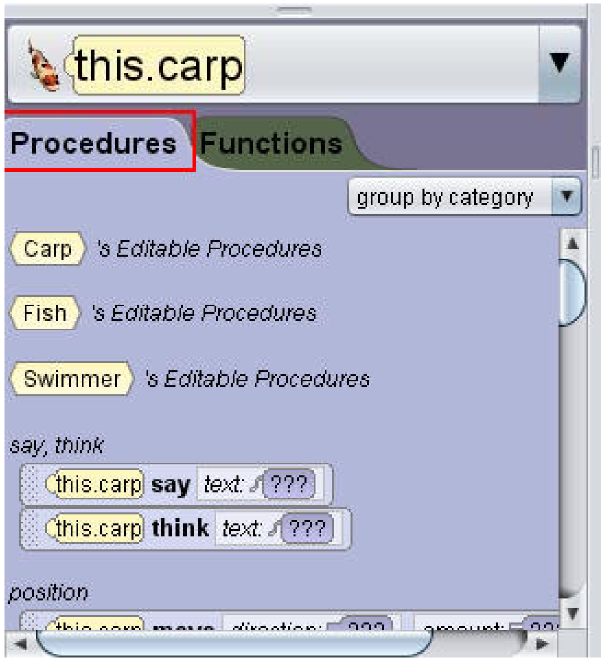
Click the Edit Code button to display the Code editor. The Code editor is where you add the programming instructions to program your animation.



Methods Panel

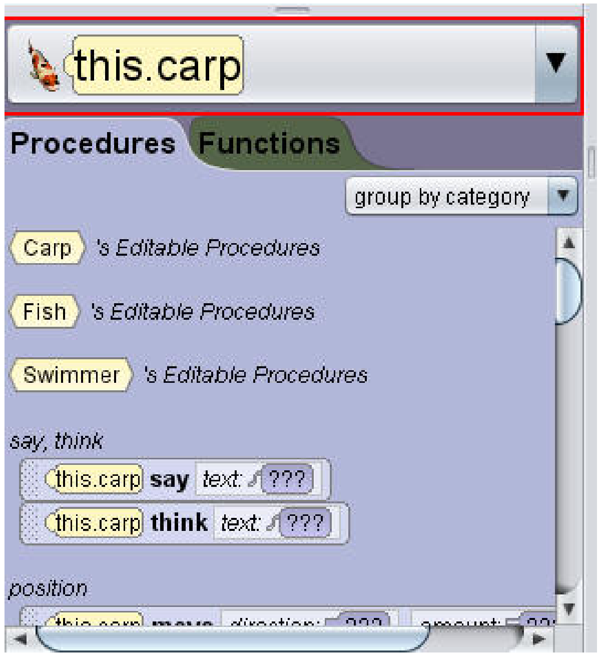
The Procedures tab, located within the Methods Panel in the Code editor, displays pre-defined methods for the selected instance, as well as methods defined for the class of objects.

A procedure is a piece of program code that defines how the object should execute a task. Alice 3 has a set of procedures for each class; however, users can create or "declare" new procedures.


this.carp
Procedures
Functions
group by category
Carp 's Editable Procedures
Fish 's Editable Procedures
Swimmer 's Editable Procedures
say, think
this.carp say text: ???
this.carp think text: ???
position

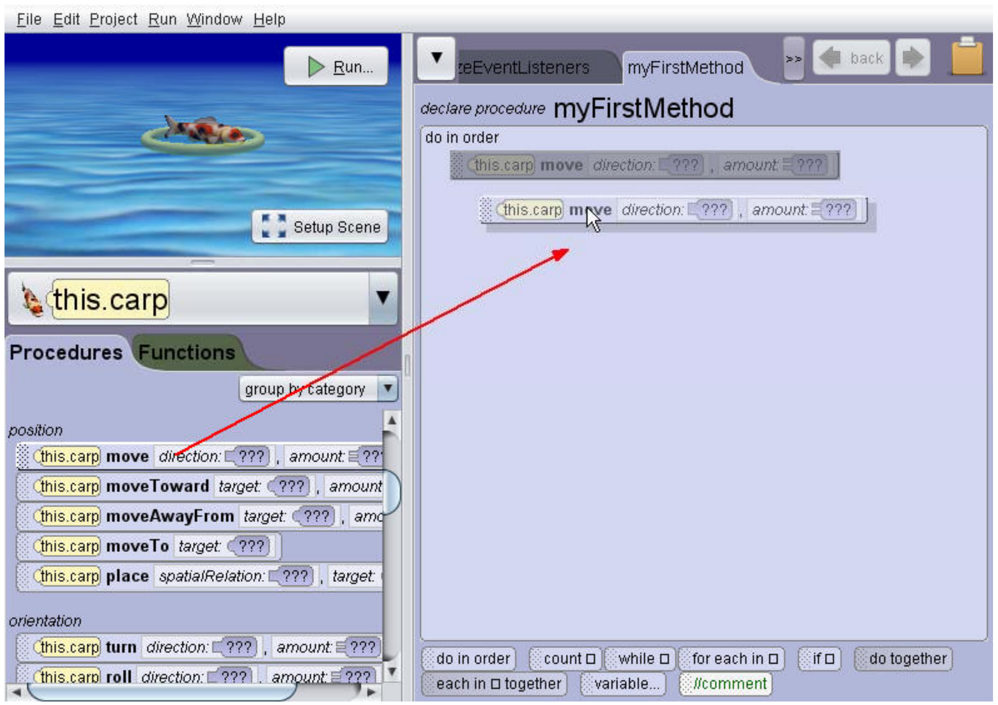
Instance Menu in Methods Panel

The instance menu displays above the Procedures tab. The down pointing triangle on the right side of the menu indicates that the menu drops down when selected.


this.carp
Procedures
Functions
group by category
Carp 's Editable Procedures
Fish 's Editable Procedures
Swimmer 's Editable Procedures
say, think
this.carp say text: ???
this.carp think text: ???
position

Create a Programming Instruction

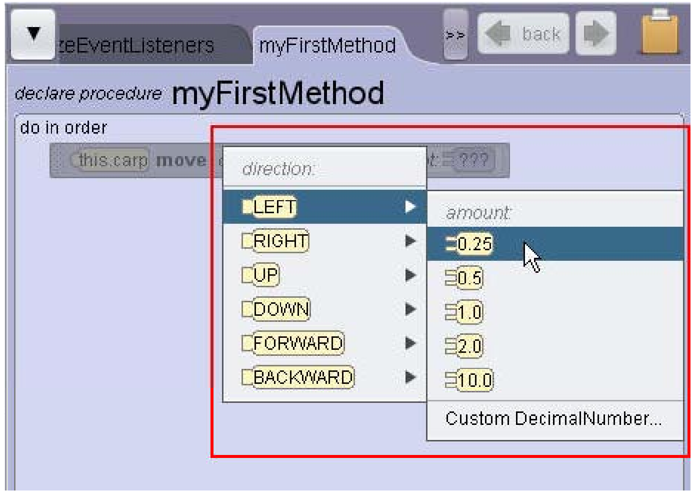
From the Methods Panel, click and drag the desired programming instruction into the myFirstMethod tab of the Code editor.


File Edit Project Run Window Help
Run...
Setup Scene
this.carp
Procedures
Functions
group by category
position
this.carp move direction: ??? , amount: ???
this.carp moveToward target: ??? , amount
this.carp moveAwayFrom target: ??? , amo
this.carp moveTo target: ???
this.carp place spatialRelation: ??? , target:
orientation
this.carp turn direction: ??? , amount: ???
this.carp roll direction: ??? , amount: ???
teEventListeners
myFirstMethod
back
declare procedure myFirstMethod
do in order
this.carp move direction: ??? , amount: ???
do in order
count
while
for each in
if
do together
each in together
variable...
//comment

Select Values for Method Arguments

After you drag the programming instruction into the myFirstMethod tab, use the cascading menus to select the value for each argument used in the method.

An argument is a value that is used by the method to perform an action.



Procedure Argument Types

Argument types may include:

- Direction
- Amount
- Duration
- Text

Alice 3 recognizes how many arguments are needed for each programming instruction. It presents you with the correct number of cascading menus to specify the values for each of those arguments.

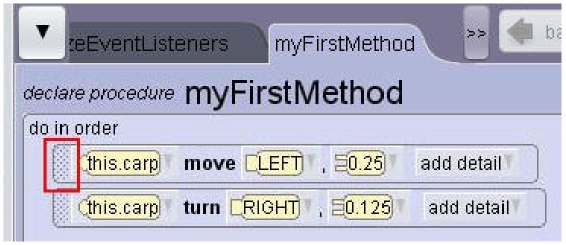
Copy Programming Instructions

To copy a programming instruction, you may use any one of these methods:

- The CTRL + Drag method.
- Right-click and use the Copy to Clipboard option.
- Click and drag the programming instruction to the clipboard.

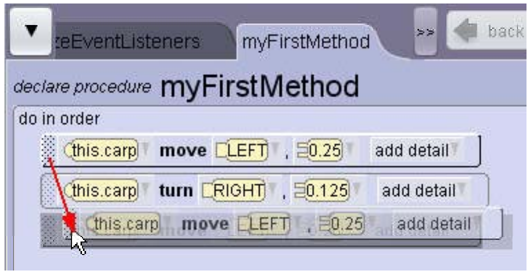
Steps to Use the CTRL + Drag Method

1. Hold down the CTRL key on your keyboard.
2. Click on and hold the programming instruction handle.



```
declare procedure myFirstMethod
do in order
    this.carp  move LEFT , 0.25  add detail
    this.carp  turn RIGHT , 0.125  add detail
```

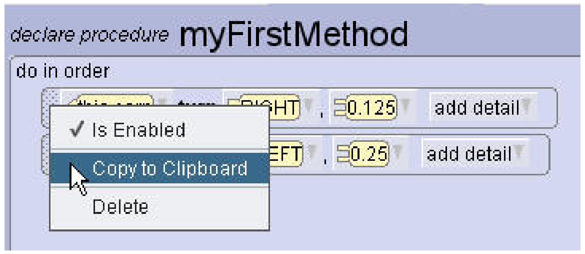
3. Drag the handle to the desired location in the code, or to the clipboard. Release the mouse button before releasing the CTRL key.



```
declare procedure myFirstMethod
do in order
    this.carp  move LEFT , 0.25  add detail
    this.carp  turn RIGHT , 0.125  add detail
    this.carp  move LEFT , 0.25  add detail
```

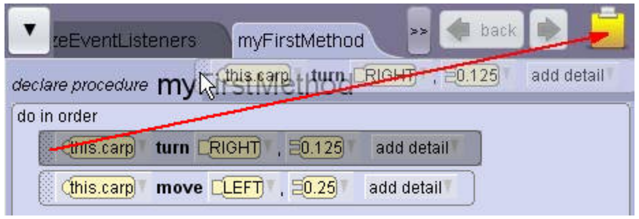
Steps to Use the Right-Click Copy Method

1. Right-click on the programming instruction handle.
2. Select the Copy to Clipboard option.



Steps to Use the Click + Drag To Clipboard Method

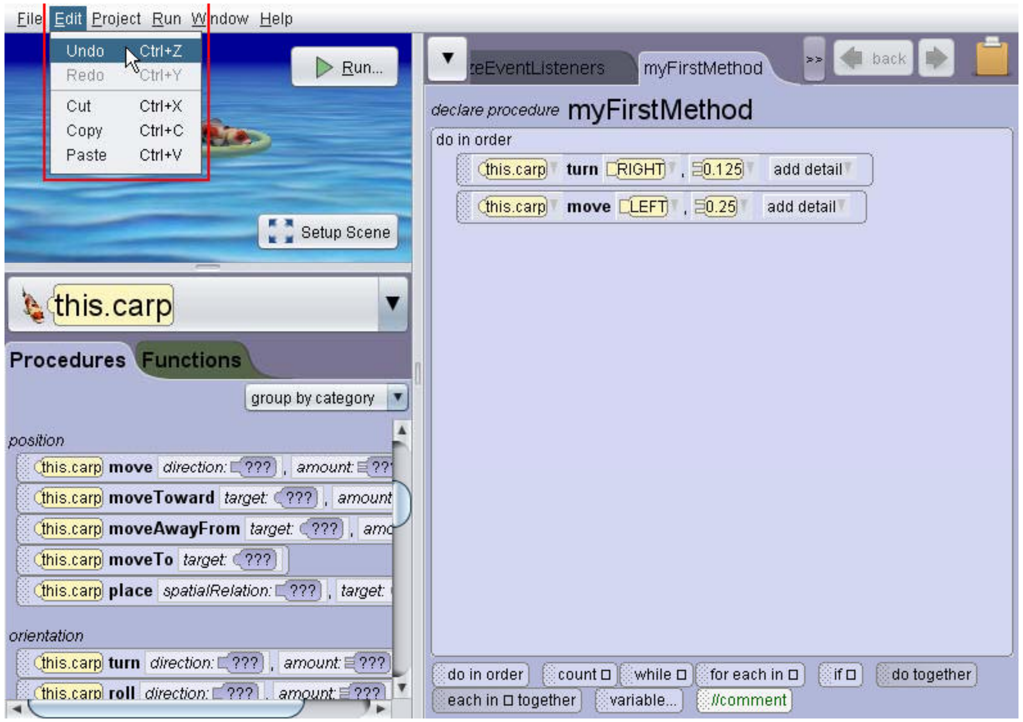
1. Click and drag a copy of the programming instruction to the clipboard icon. The clipboard changes color when the mouse pointer makes contact with the clipboard icon. Use this method when copying programming instructions between tabs.



this.carp turn RIGHT , 0.125 add detail
this.carp move LEFT , 0.25 add detail

Undo an Action

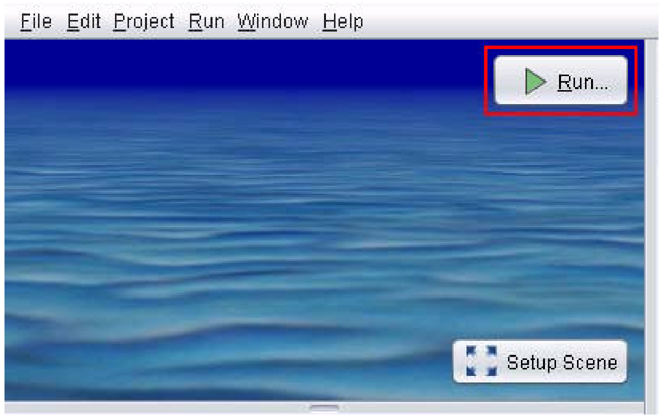
Undo an action using the Undo option on the Edit menu, or the keyboard shortcut CTRL + Z.


File Edit Project Run Window Help
Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Run...
Setup Scene
this.carp
Procedures Functions
group by category
myFirstMethod
declare procedure myFirstMethod
do in order
this.carp turn RIGHT , 0.125 add detail
this.carp move LEFT , 0.25 add detail
do in order count while for each in if do together each in together variable... //comment

Test and Debug Your Animation
Testing

Once you create the programming instructions for your animation, you need to test your program.

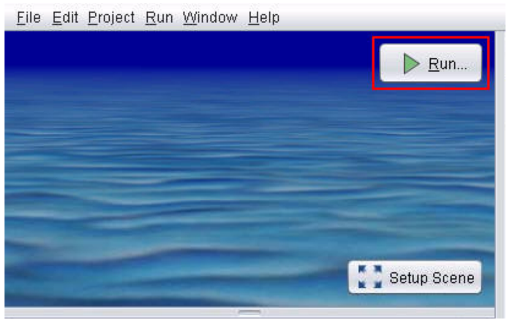
- To test your program, click the Run button.
- Run the animation to test that it functions properly and executes as planned and without error.
- Test the animation frequently during development.



Test and Debug Your Animation
Testing the Limits of Your Program

Testing the limits of your program is an important part of the process. For example, change the value of an argument in a procedure in an effort to intentionally "break" the code to prove that it works under extreme conditions .

- What happens if a number is very large? or negative?
- Test the limits of the animation frequently during development.



Test and Debug Your Animation
Debugging

The cycle of testing your program, identifying errors or unintended results, rewriting the code, and re-testing is a process referred to as debugging your program.

Software programs, such as animations, are tested by entering unanticipated commands to try and "break" the code. When something is broken or doesn't work as intended in a software program, it is often referred to as a "bug". Debugging is the process of finding bugs in a software program.

Testing and Debugging Techniques

Use some of the following techniques as you program the animation in Alice 3:

- Adjust the arguments that specify the direction, distance, and duration that objects move.
- Adjust the mathematical expressions that manipulate the direction, distance, and duration that objects move.
- Refine or replace instructions in the code that do not work as intended.
- Resolve errors created by the programmer.

Summary

In this lesson, you should have learned how to:

- Identify scene components
- Create and save a new project
- Add an object to a scene
- Communicate the value of saving multiple versions of a scene
- Code a simple programming instruction
- Use the copy and undo commands
- Understand the value of testing and debugging