Getting Started with Java Using Alice

Use Functions

Objectives

This lesson covers the following objectives:

- Use functions to control movement based on a return value

Functions

Functions:

- Are used to ask questions about properties of an object.
- Are similar to procedures except that they return a value of a particular type.
- Can be used to compute a value.

Functions answer questions about an object, such as its height, width, depth, and even its distance to another object.

Functions Precisely Answer Questions

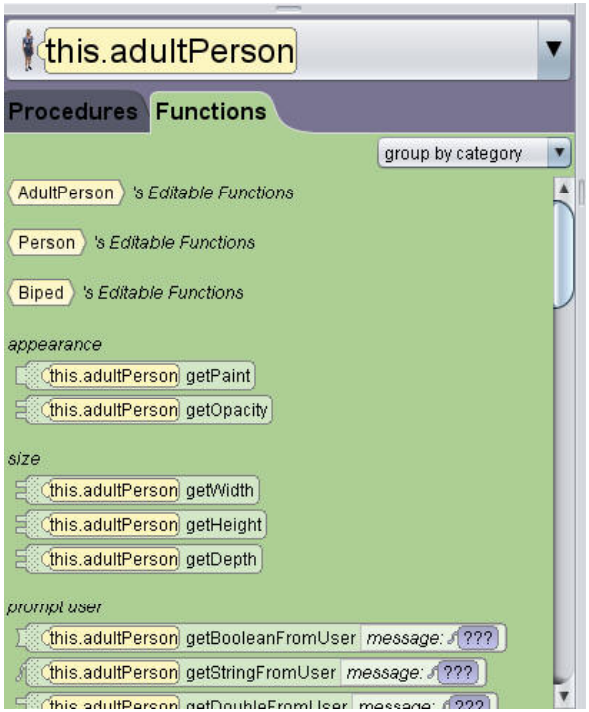
Functions provide precise answers to questions, such as:

- What is the distance between the helicopter and the ground?
- What is the height of the cat?
- What is the width of the owl?

A boolean function returns either a true or false value. For example, if the isFacing function is called to determine if the person object is facing the tree object, a true or false value will be returned.

Functions Tab

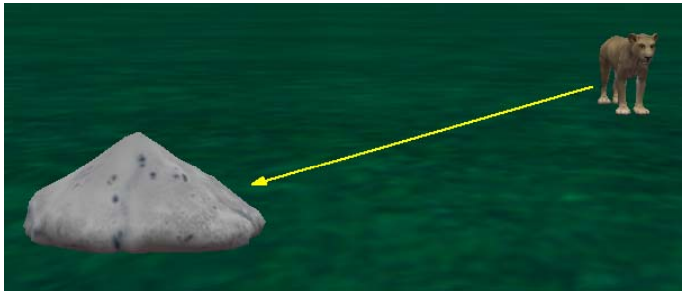
The Functions tab is in the methods panel. Select the object from the Instance menu, and then view its functions.



Functions Solve Distance Problems

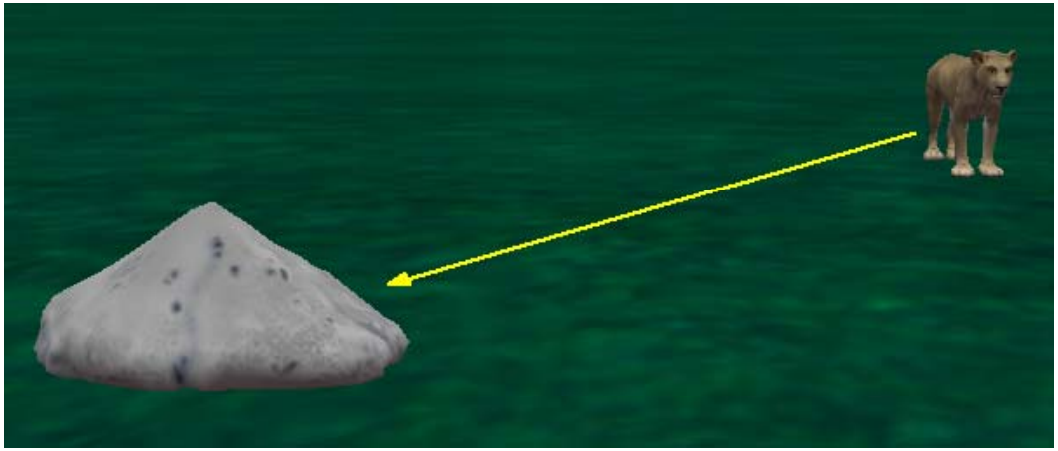
Suppose we want to move the lioness directly to the rock without having to manually determine, through trial and error, the distance between the lion and the rock.

We could guess the distance by specifying a placeholder value and testing the movement until we get close to the desired end result, but a more efficient way is to use a function to determine the exact distance to move.



Use getDistanceTo Function

Use the getDistanceTo function in the move procedure to solve this distance problem.



Steps to Use the getDistanceTo Function

1. Determine the moving object and target object.
2. In the Code editor, select the moving object from the Instance menu.
3. Drag the move procedure into the Code editor.
4. Select the direction and a placeholder argument for distance (the distance argument will be modified in the next step).

5. From the Functions tab, drag the getDistanceTo tile onto the highlighted distance value.
6. Select the target object.

Test the Function

Click the Run button to test the programming statement.

In the example below, the lioness moves to the center of the rock at run-time. This could be read out loud as "determine the distance from the center of the lioness to the center of the boulder and then move the lioness forward that amount."

Test the Function

The lioness moves to the middle of the rock. This is because the getDistanceTo function calculates the distance between the centers of both objects.

The function calculated the distance from the center of the lioness to the center of the rock, and moved the object using that value.

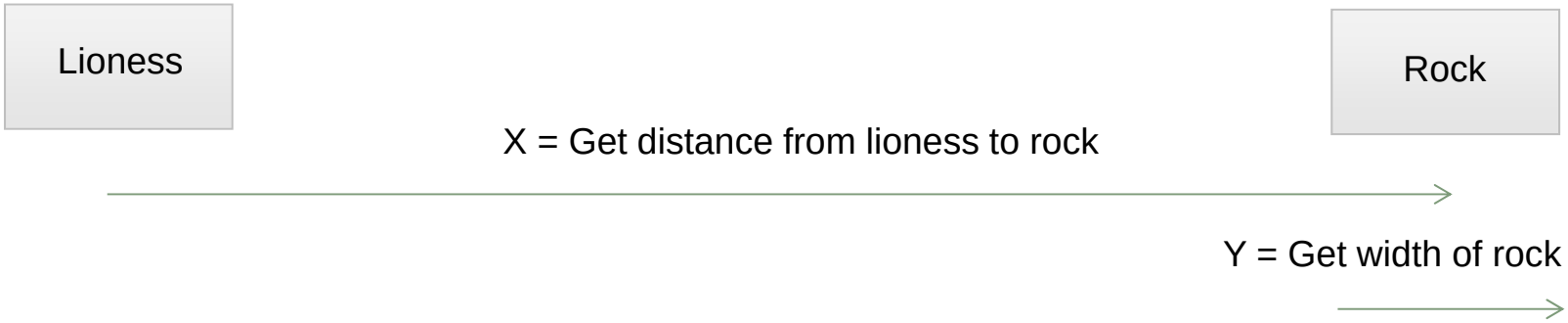
Avoid Collisions

You can enhance function calls using the math operators (+) addition, (-) subtraction, (*) multiplication, and (/) division.

For example, you can reduce the distance an object will move to avoid a collision.

Using Math Operators

A function determines the distance between the lioness and the rock. To reduce the value returned by the getDistance function, the subtraction operator subtracts a specified value. The specified value is determined by calling the getWidth function and dividing that value in half.

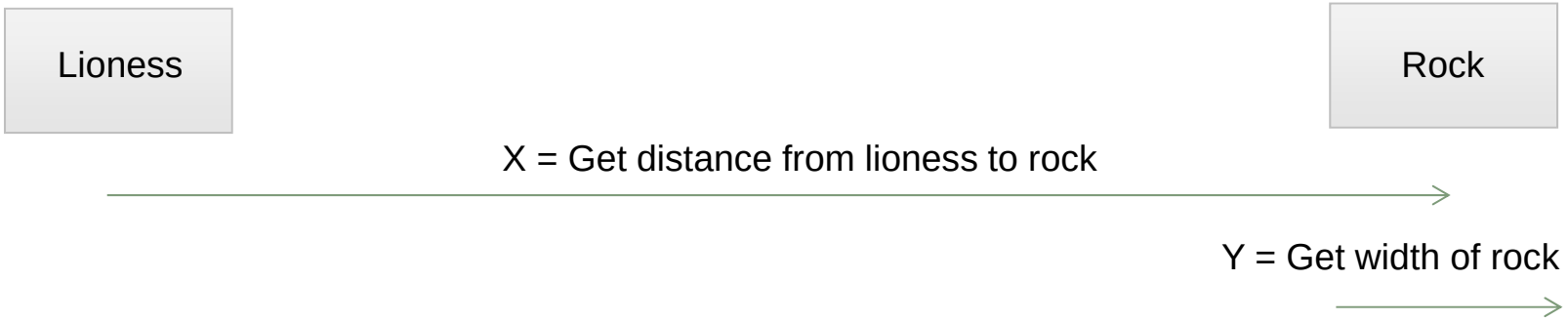


$$Z = X – (Y / 2)$$

Examine the Math Calculation

Let's examine the math calculation Z = X - (Y / 2):

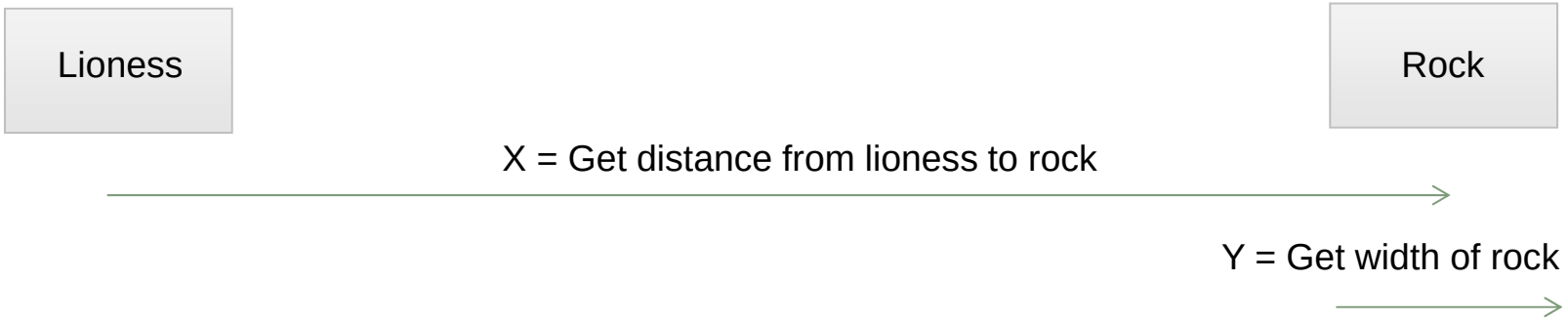
- Z represents the total distance the lioness will move.
- X represents the distance between the lioness and rock.
- Y represents the width of the rock.
- Y / 2 represents the width of the rock divided by 2.
- ( ) represent the order of precedence.



Math Operator Tip

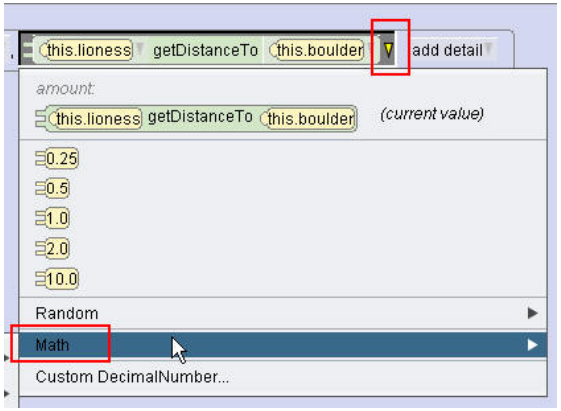
Why did we divide the width of the rock in our calculation?

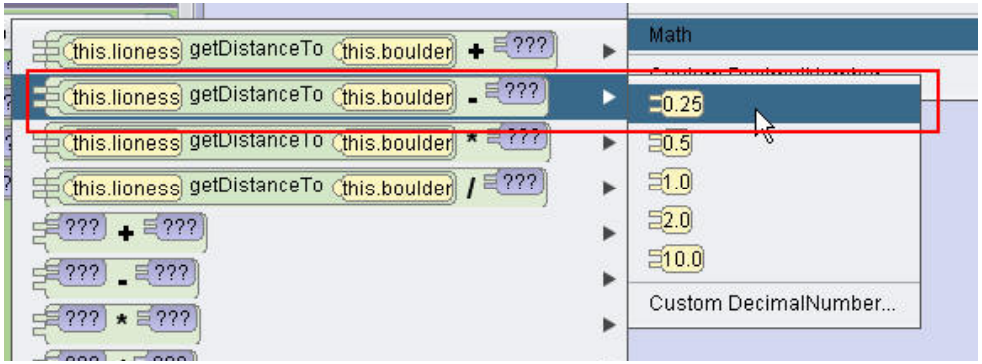
Because we want the animation to appear as though the lioness is moving to the very edge of the rock. If we used the entire width of the rock, the lioness would stop further from the rock than desired.



Steps to Use Math Operator to Avoid Collision

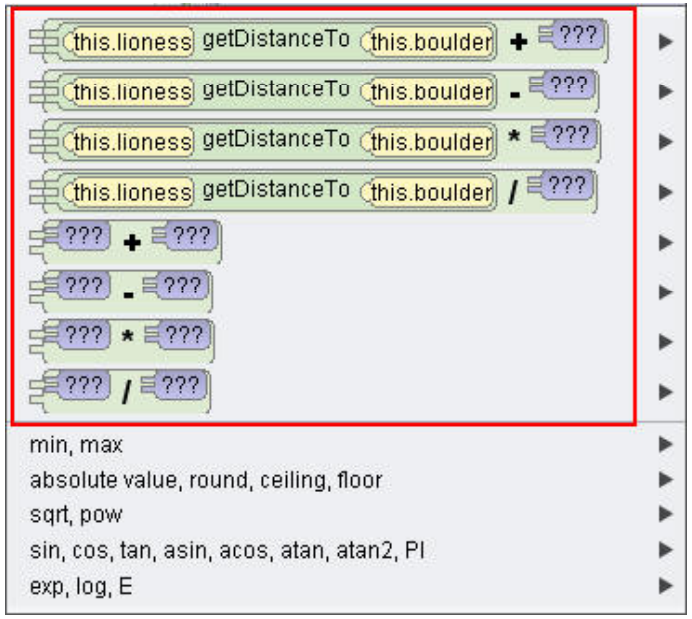
1. Click the outer arrow next to the function argument.
2. Select Math.
3. Select the getDistanceTo subtraction option.
4. Select the distance amount.
5. Run the animation to test how the object moves at run-time.





Understanding the Math Menus Example 1

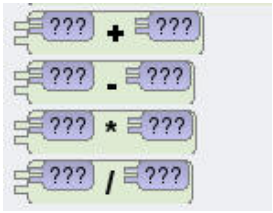
The following image displays the math operators (+ - * /) requiring one or two arguments. Each option will provide one or two cascading menus to specify the argument values.



Understanding the Math Menus Example 2

The following image displays an (+) addition operator requiring a single argument.

The following image displays math operators requiring two arguments.



Remember, you can select placeholder values for the arguments. Placeholder values can always be edited.

Remove Object's Depth from Function

Another precise way to avoid collisions is to remove the depth (length) of the moving object from the function. In the example below, the lioness will move the distance to the rock, minus the depth of the lioness.

Lion

Rock

Distance from lioness to rock

Lioness depth

Depth is Measured from Object's Center

When a distance value is calculated, it is measured from one object's center to another object's center. The same is true for math calculations. When the depth of the lioness is subtracted from the rock, it is actually subtracted from the center of the rock.

Lioness

Rock

Distance from lioness to rock

Lioness depth

Steps to Remove Depth from Function

1. In the Functions tab, drag the moving object's getDepth function onto the highlighted distance value.
2. Run the animation to test how the object moves at run-time. Adjust with additional math calculations if necessary.

Summary

In this lesson, you should have learned how to:

- Use functions to control movement based on a return value