Getting Started with Java Using Alice

Use Expressions

ORACLE ACADEMY

Objectives

This lesson covers the following objectives:

- Create an expression to perform a math operation
- Interpret a math expression

Using Expressions

Expressions are a combination of values that, when arranged correctly, result in a final value.

- Expressions are typically used in Alice 3 to solve timing and distance problems in your programs.
- Example: 2 + 2 = 4
  - Two values (2, 2) and the operator (+) result in the final value (4).

Expressions in Alice 3

Expressions are created in Alice 3 using the following built-in math operators:

- Add (+)
- Subtract (-)
- Multiply (*)
- Divide (/)

Location of Math Operators

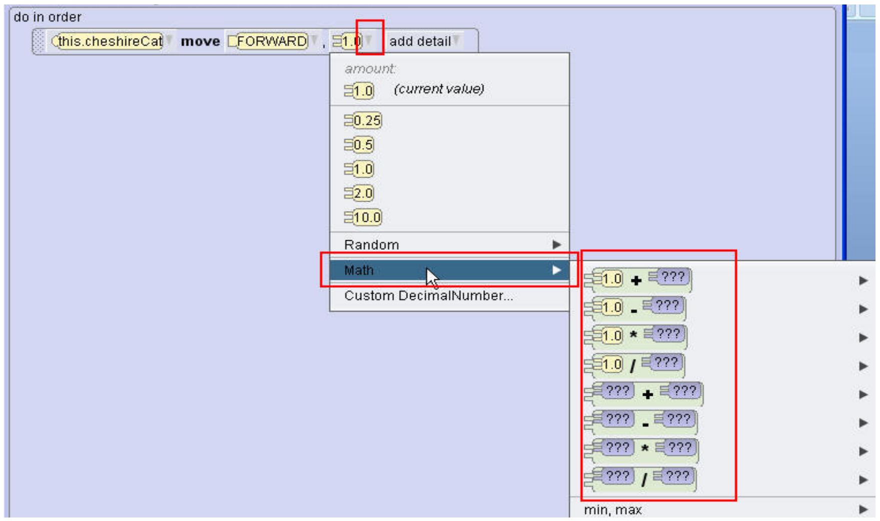
Math operators are available in the cascading menus where you select the argument values for:

- Amount and Duration
- getDistance functions

View Expressions in a Distance Argument

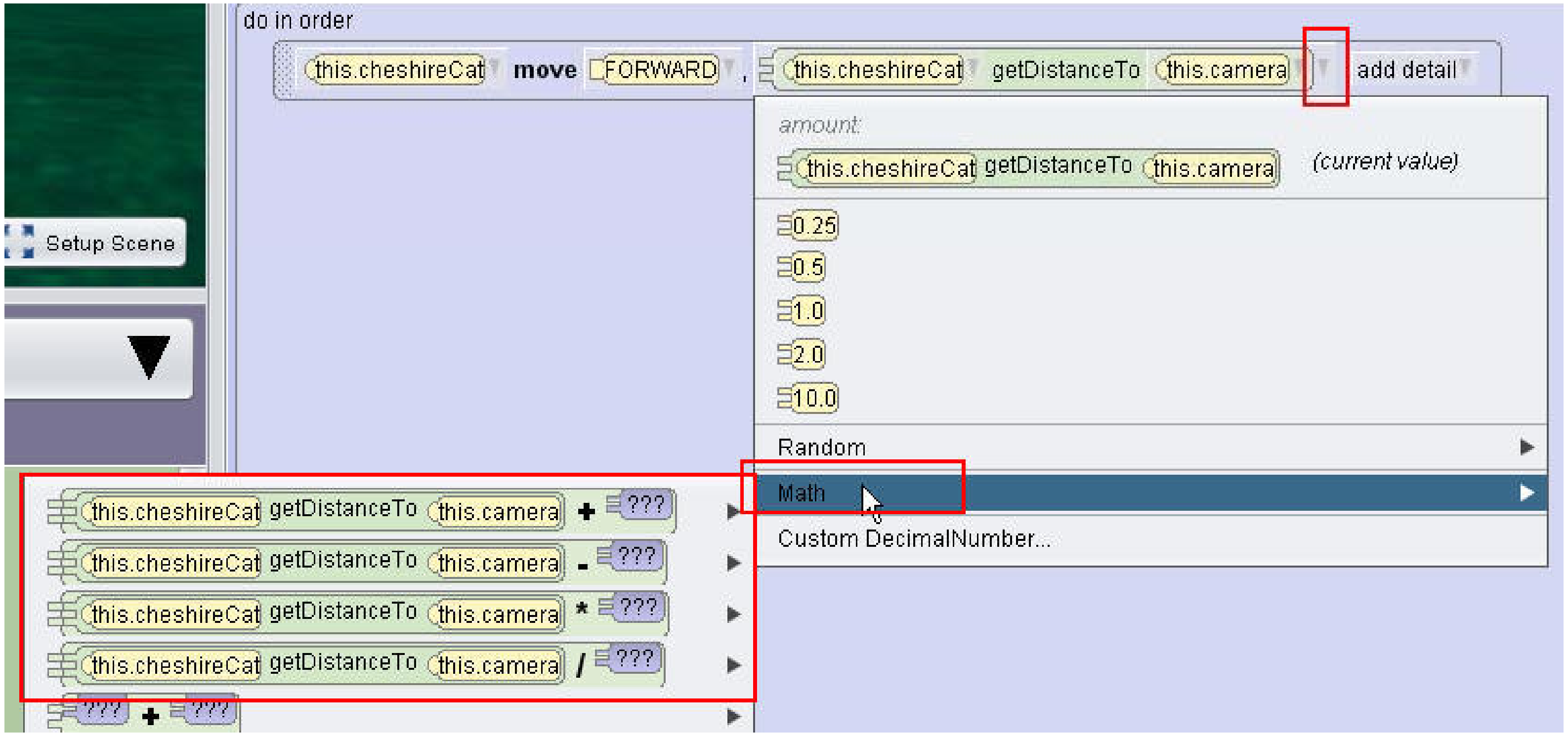
Select the Math option to view the available math operators in a procedure’s distance argument.

- Select from two different sets of math expressions.
  - The first set lets you specify the value of one operator
  - the second set lets you specify the values of both operators.



getDistanceTo Function Display

Select the Math option to view the available math operators for the getDistanceTo function's argument.

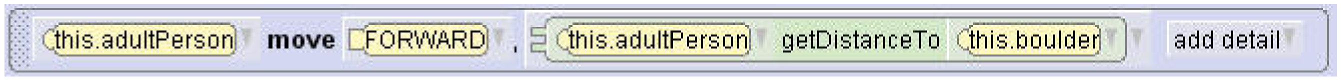


do in order
this.cheshireCat move FORWARD , this.cheshireCat getDistanceTo this.camera add detail
amount:
this.cheshireCat getDistanceTo this.camera (current value)
0.25
0.5
1.0
2.0
10.0
Random
Math
Custom DecimalNumber...
this.cheshireCat getDistanceTo this.camera + ???
this.cheshireCat getDistanceTo this.camera - ???
this.cheshireCat getDistanceTo this.camera * ???
this.cheshireCat getDistanceTo this.camera / ???
??? + ???
Setup Scene

Distance Problem

The problem: A person object moves to the center of the boulder, rather than near it.

- This is because the getDistanceTo function calculates the distance from the center of the person object to the center of the target object (boulder).
- We need to reduce the distance the person object moves so it does not collide with the boulder.
- A math calculation is used to reduce the distance the person object moves.



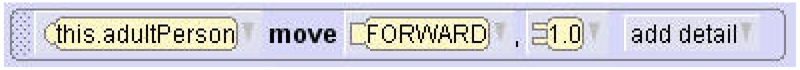
```
this.adultPerson move FORWARD , this.adultPerson getDistanceTo this.boulder add detail
```

Steps to Create an Expression

1. Summarize the timing or distance problem in your program.
2. Consider the expression that will solve the problem.
3. Code the expression.
4. Test and debug the expression until the animation works as intended.

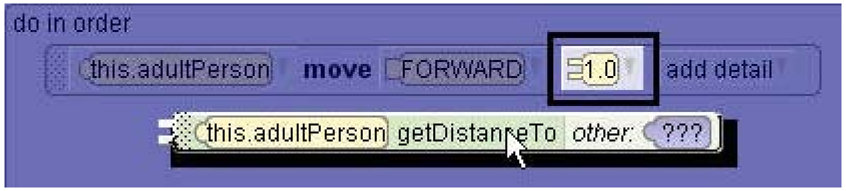
Steps to Move an Object the Distance to Another Object

1. Drag the move procedure for an object into the Code editor. Select forward and a distance placeholder value.



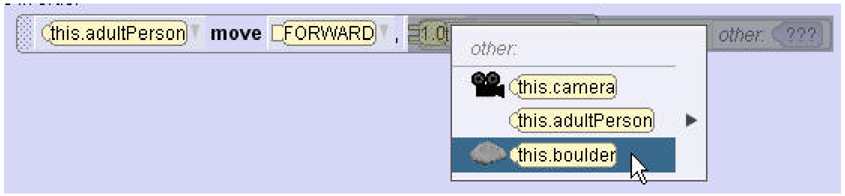
```
this.adultPerson move FORWARD , 1.0 add detail
```

2. From the Functions tab, drag the getDistanceTo function onto the distance argument placeholder.



```
do in order
  this.adultPerson move FORWARD 1.0 add detail
  this.adultPerson getDistanceTo other: ???
```

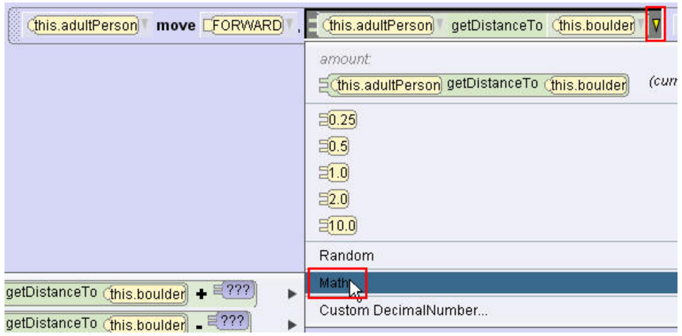
3. From the cascading menu, select the target object to which the object should move.



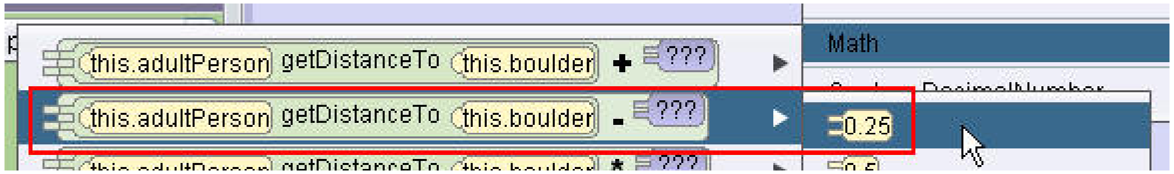
```
this.adultPerson move FORWARD , 1.0 other: ???
```

Steps to Modify the Distance Using a Math Operator

1. From the getDistanceTo tile, click the outer-most arrow to open the menu of distance values, and then select the Math option.


this.adultPerson
move
FORWARD
this.adultPerson getDistanceTo this.boulder
amount:
this.adultPerson getDistanceTo this.boulder
(curr
0.25
0.5
1.0
2.0
10.0
Random
Math
Custom DecimalNumber...
getDistanceTo this.boulder + ???
getDistanceTo this.boulder - ???

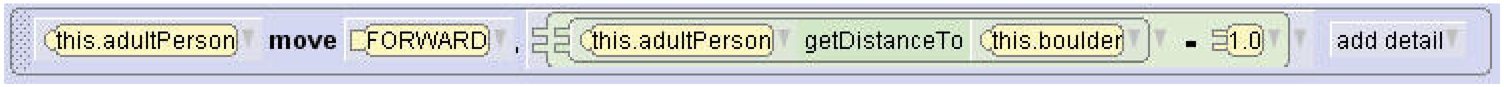
2. Select getDistanceTo - ???
3. Select a default value to reduce the distance by, or select Custom DecimalNumber… to enter a value.


this.adultPerson getDistanceTo this.boulder + ???
this.adultPerson getDistanceTo this.boulder - ???
Math
0.25

4. Test and debug the expression as necessary.

Expression Example

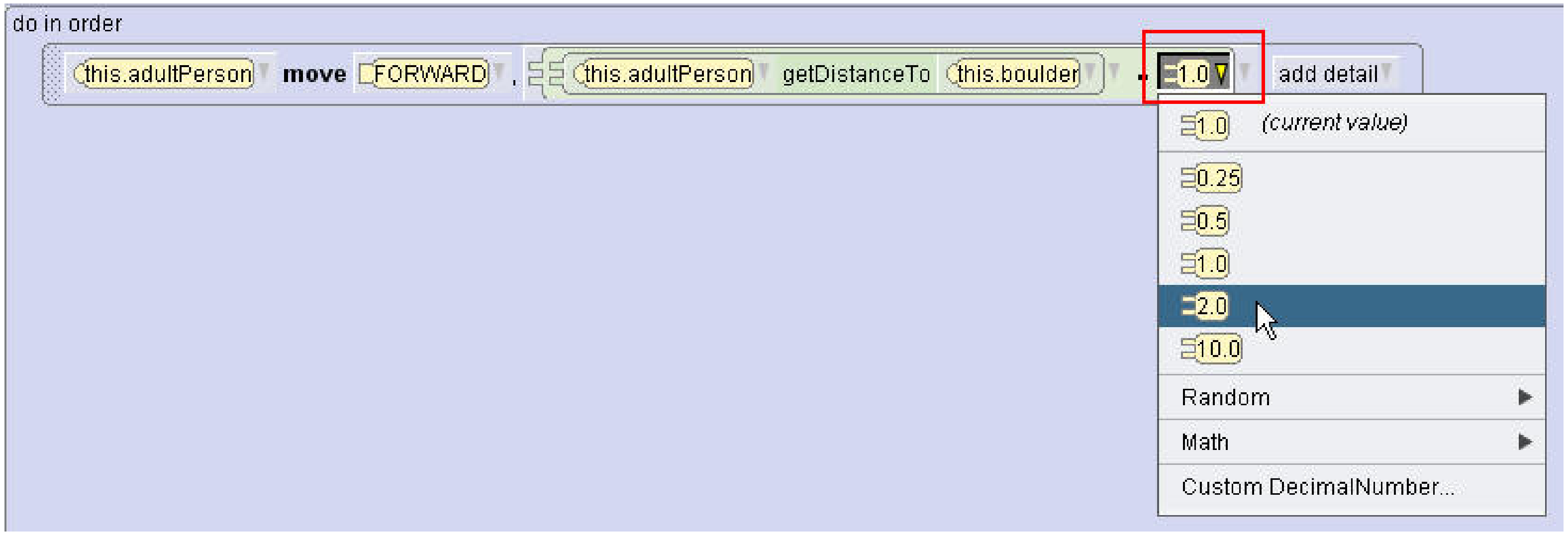
This expression reduces the distance that the person travels so the person does not collide with the boulder. This was tested and debugged several times until the correct expression was achieved.



```
this.adultPerson move FORWARD , this.adultPerson getDistanceTo this.boulder - 1.0 add detail
```

Editing the Expression

During the debugging process, you may need to adjust the value of the expression. Click the arrow next to the value, and select a new default value or use the Custom DecimalNumber... menu to select a more defined value.

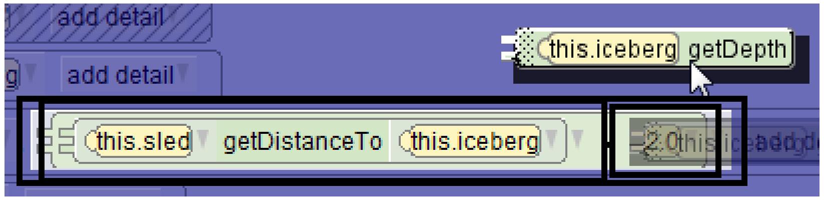

do in order
this.adultPerson move FORWARD , this.adultPerson getDistanceTo this.boulder - 1.0 add detail
1.0 (current value)
0.25
0.5
1.0
2.0
10.0
Random
Math
Custom DecimalNumber...

Subtract Depth from the Expression

Subtracting the depth of the target object from the expression is a more precise way to ensure that the moving object lands directly near the target object without going through its center.

Steps to Subtract Depth from Expression

1. Select the target object in the instance menu.
2. From the Functions tab, drag the getDepth tile onto the existing distance value in the expression.

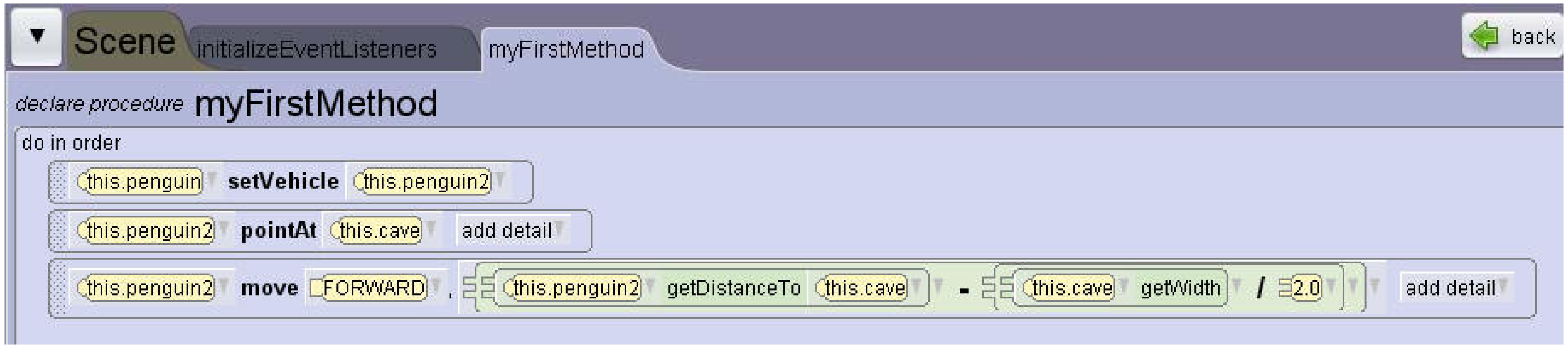


3. Test and debug the animation, and adjust the expression as necessary.

Interpret an Expression

To understand a programming statement that includes an expression, you often need to interpret the expression. To interpret an expression you can:

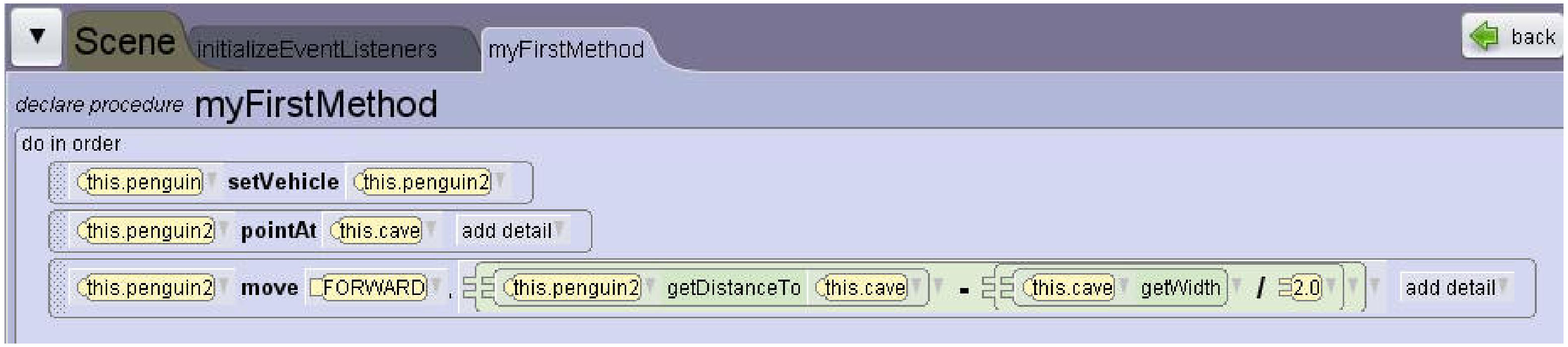
- Read it from left to right.
- Recognize the instances specified in the expression and describe what each one does.



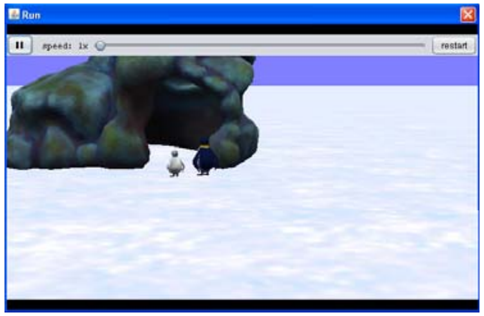
```
declare procedure myFirstMethod
do in order
    this.penguin setVehicle this.penguin2
    this.penguin2 pointAt this.cave add detail
    this.penguin2 move FORWARD , ( this.penguin2 getDistanceTo this.cave ) - ( this.cave getWidth / 2.0 ) add detail
```

Expression Example

Examine the visual associated with this expression. The adult and baby penguin instances are moving towards the cave. Will they go inside?



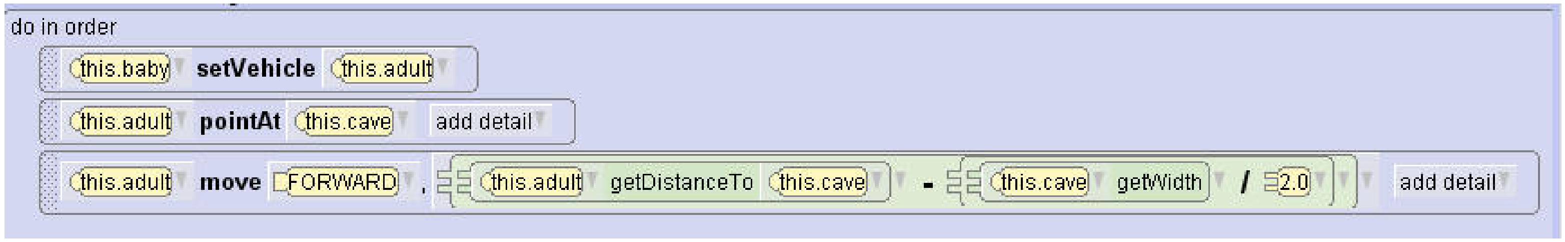
```
declare procedure myFirstMethod
do in order
    this.penguin setVehicle this.penguin2
    this.penguin2 pointAt this.cave add detail
    this.penguin2 move FORWARD , ((this.penguin2 getDistanceTo this.cave) - ((this.cave getWidth) / 2.0)) add detail
```



Interpretation of an Expression

This expression tells us the following:

- The adult penguin moves forward towards the cave.
- The distance between the adult penguin and cave is determined by the getDistanceTo function.
- The distance traveled is reduced by half the width of the cave.
  - The width of the cave is determined by the getWidth function.



```
do in order
    this.baby setVehicle this.adult
    this.adult pointAt this.cave add detail
    this.adult move FORWARD , this.adult getDistanceTo this.cave - this.cave getWidth / 2.0 add detail
```

Formulating the Expression

To interpret an expression, it is helpful to draw a picture or write the values you know before formulating the expression.

Example:

Z = X – (a / b)

Z = Distance moved

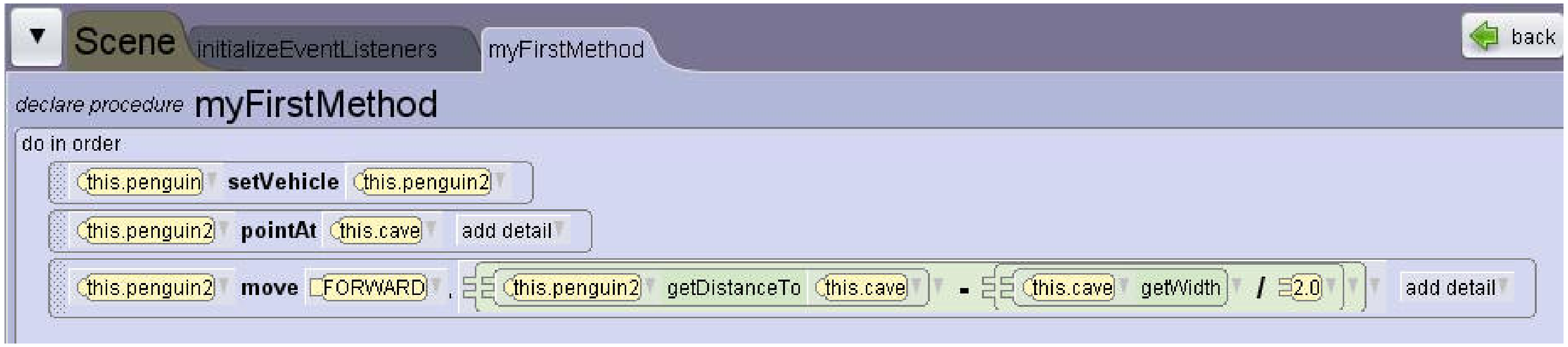
X = Distance from adult penguin to cave

a = Cave width

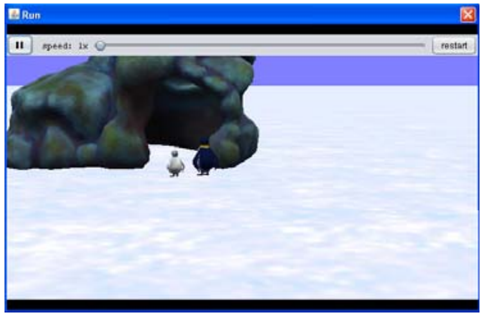
b = 2

Expression Example Answer

Examine the visual associated with this expression. The adult and baby penguin instances are moving towards the cave. Will they go inside? No, they stop outside the cave.



```
declare procedure myFirstMethod
do in order
    this.penguin setVehicle this.penguin2
    this.penguin2 pointAt this.cave add detail
    this.penguin2 move FORWARD , ((this.penguin2 getDistanceTo this.cave) - ((this.cave getWidth) / 2.0)) add detail
```



Summary

In this lesson, you should have learned how to:

- Create an expression to perform a math operation
- Interpret a math expression