

A Simple Quiz for Alice 3.2:



Introduction and Set-up

This tutorial will demonstrate how to create a simple quiz using the three different kinds of “ask user” functions: ask user for a number, ask user for true or false and ask user for a string.

Let's get started by setting up the world.

Open a new Alice **grass world**.

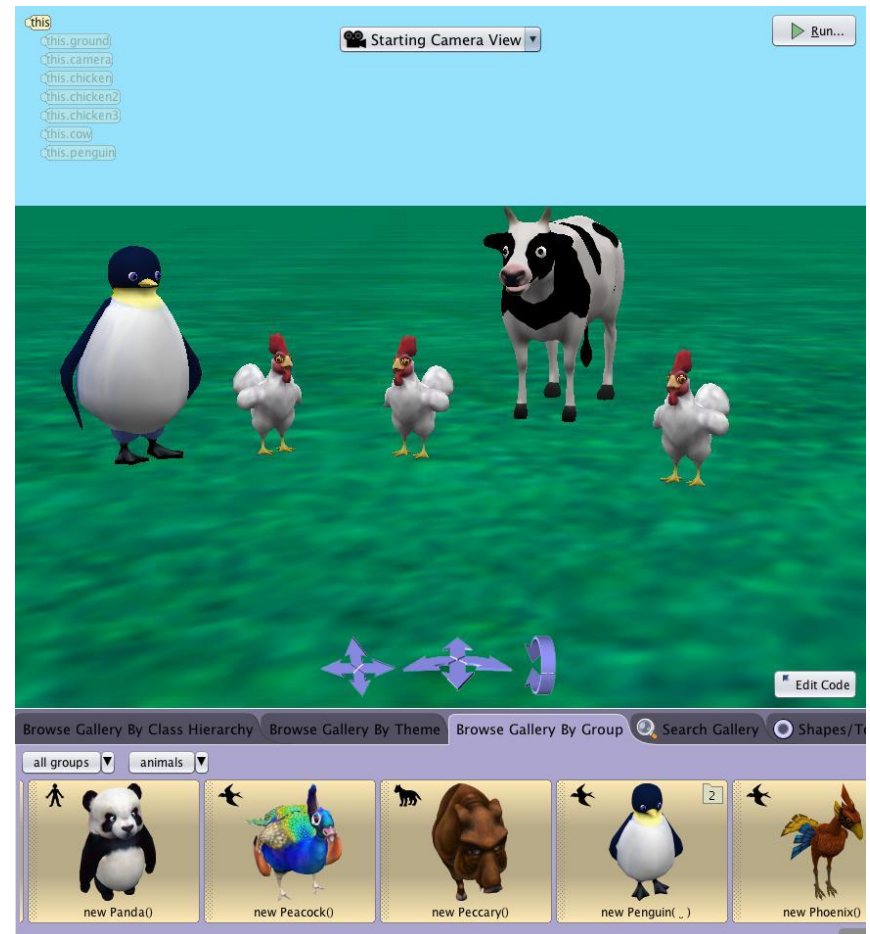
Click on **Setup Scene**.

Choose **Browse Gallery By Group**,
and click **Animals**.

Add one **Penguin**, one **Cow**, and
three **Chickens** into your world.

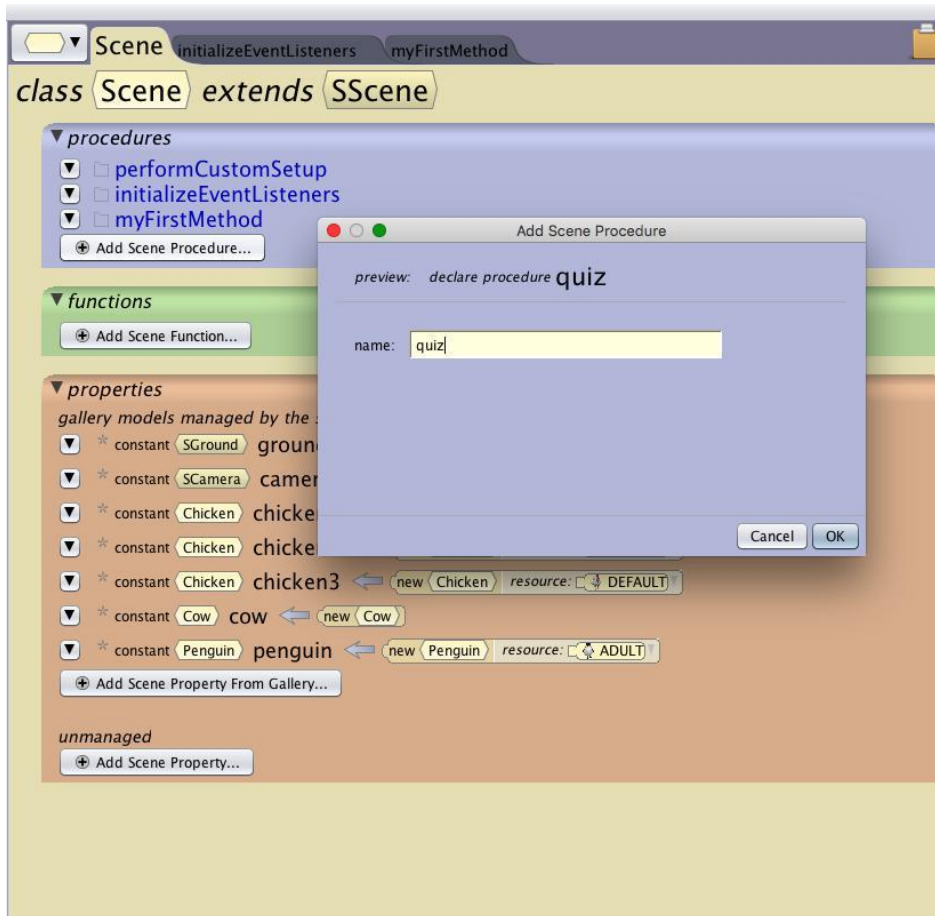
Arrange and resize the objects
so they are easy to see.

Click **Edit Code** when finished.



Step 1: Create the Quiz Procedure

Create a new scene procedure that we will use later to hold the instructions for the quiz.



Lets start by creating a quiz procedure

Click on the **Scene** tab and choose **Add Scene Procedure**

Name the new procedure: **quiz**.

Click back on **myFirstMethod**.

Step 2: Set Up MyFirstMethod

Now we will add the quiz to **MyFirstMethod**.

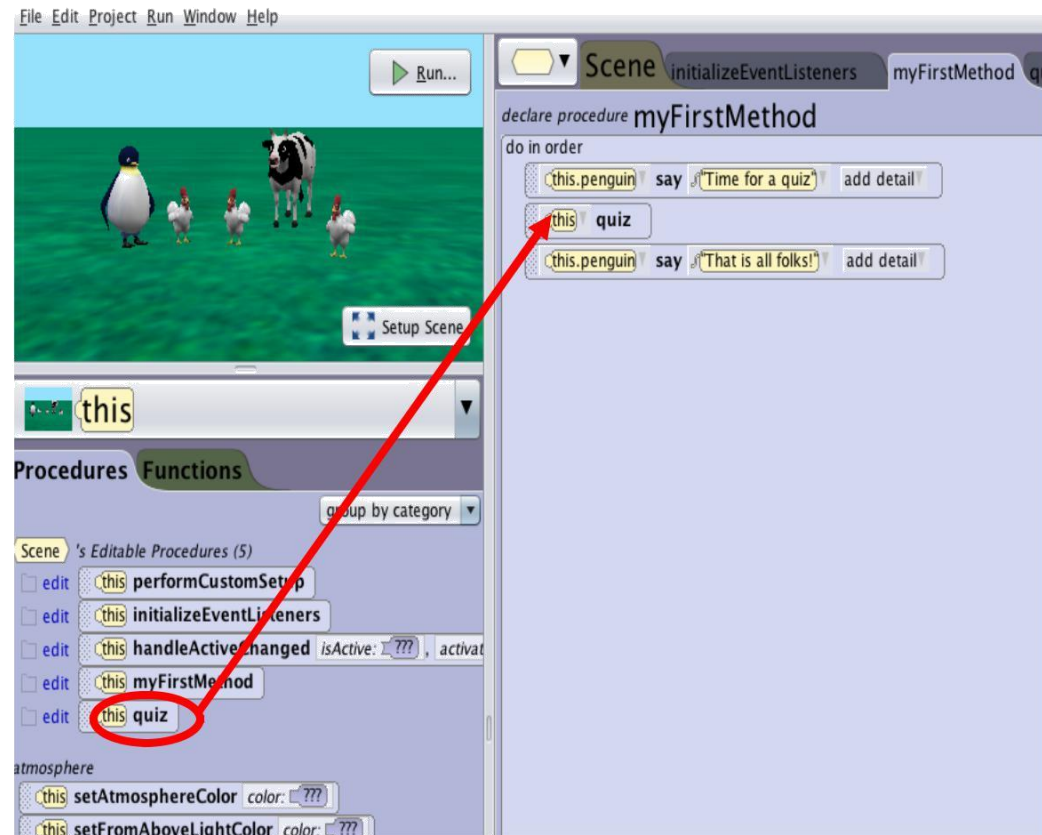
Make sure you are in **MyFirstMethod**.

Click on **this.penguin** in the object tree.

Drag the **say** procedure into the procedure editor and type in: *Time for a quiz*.

Drag in a second **say** procedure and type in: *That is all folks!*

Click on **this** in the object tree and drag the **quiz** procedure in between the two penguin say commands

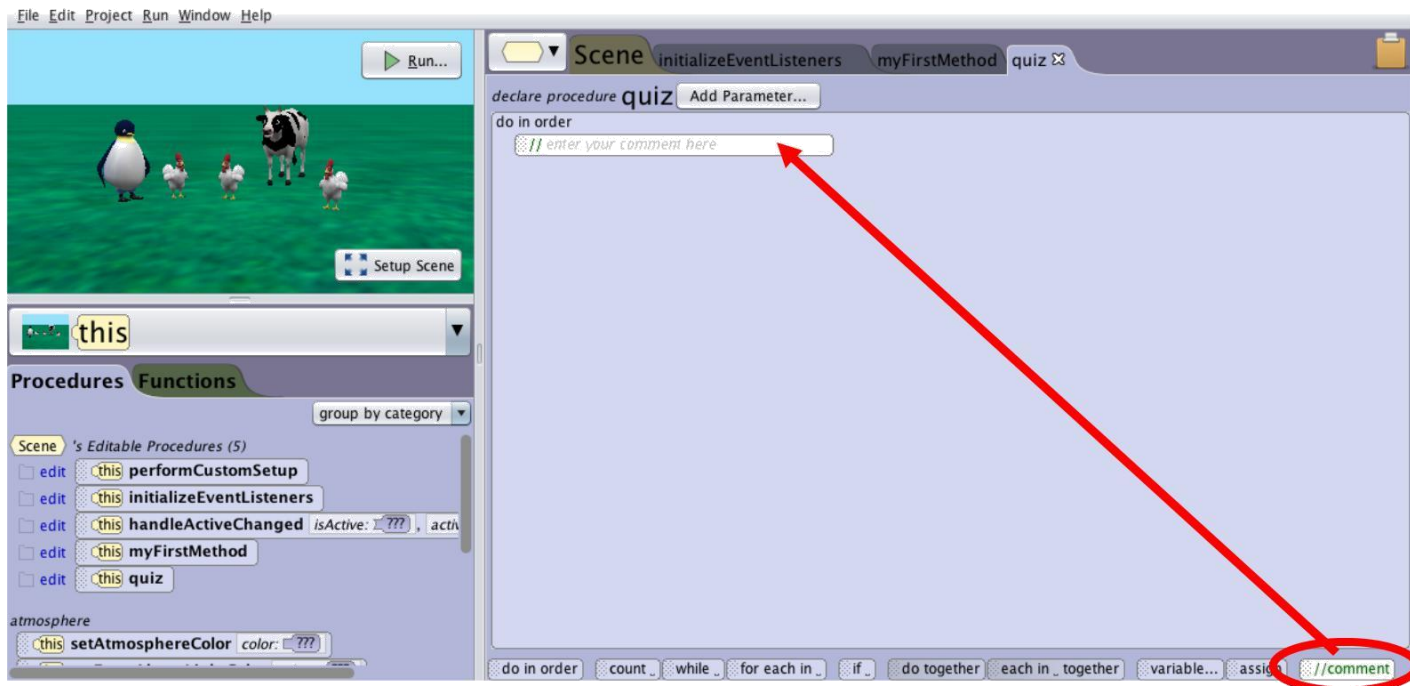


Step 3: Comments

Now we will start writing the quiz procedure. To make our code easier to read we will add in comments. Comments make it possible to add in notes about the code like, what it will do, or who wrote the code.

First click on the **quiz** procedure tab.

To add a comment drag and drop the double slash button `//comment` into the method



Step 4: Question #1

Type in the comment editor: *Question 1*

```
// Question 1
```

Click on **this.penguin** in the object tree, have it **say**: *How many animals in the world?* and set the duration to 2 seconds.

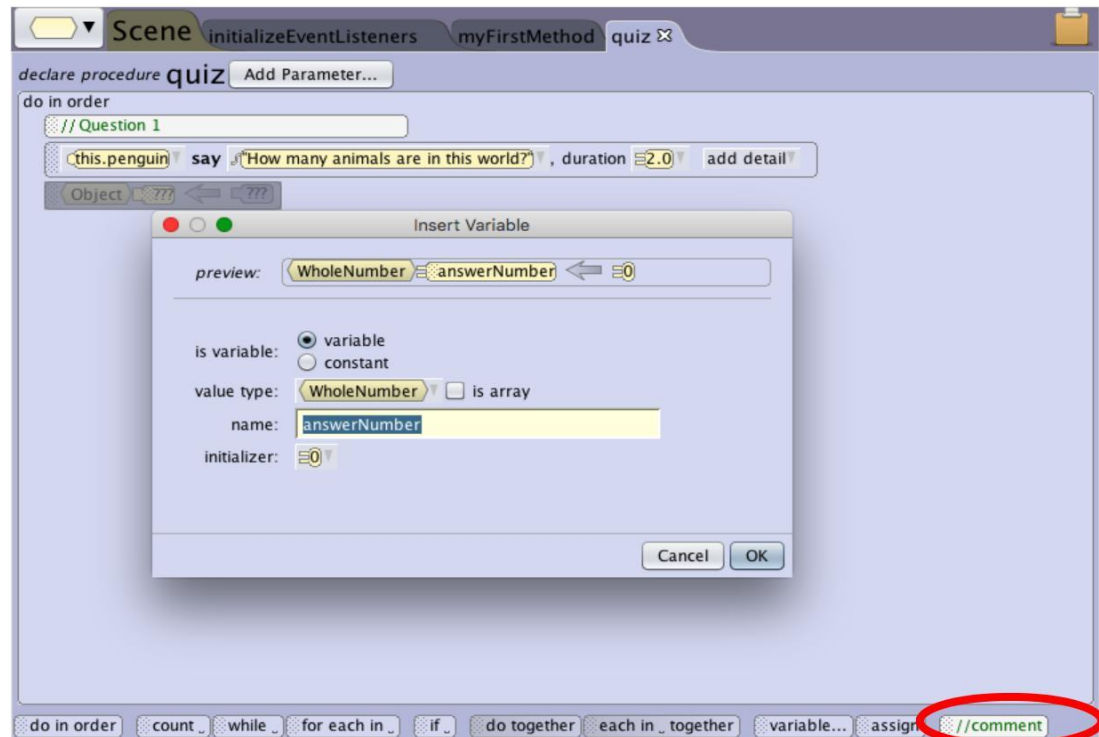
```
this.penguin say "How many animals are in this world?", duration 2.0 add detail
```

We need a variable to store the number information we get from the user.

Drag in the **variable...** button to open variable option box.

Name it *answerNumber*.

Select **WholeNumber** beside value type, initialize it to 0 and click **OK**.



Step 4: Question #1 Continued...

Click on the **world** in the object tree.

Under the world's details, **functions** find “ask user for a number.” Click and drag the **ask user for a number** over the 1 and select **other...**

Enter the string: *Enter the number of animals:*

The screenshot shows the Scratch IDE interface. On the left, a 3D scene with a penguin, chickens, and a cow is visible. Below the scene is the 'this.penguin' object tree. The 'Procedures' tab is active, showing a list of procedures. The 'Functions' tab is also visible. A red circle highlights the 'this.penguin.getIntegerFromUser' procedure. A red arrow points from this procedure to the 'Custom TextString...' dialog box in the main workspace. The dialog box shows a preview of the text 'Enter the number of animals:' and a text input field with the same text. The main workspace shows a procedure named 'quiz' with a 'do in order' block containing a 'say' block and a 'WholeNumber' block. The 'WholeNumber' block has a 'message:' field with a red circle around it, and a 'Custom TextString...' dialog box is open over it.

File Edit Project Run Window Help

Scene initializeEventListeners myFirstMethod quiz

declare procedure quiz Add Parameter...

do in order

```
// Question 1
```

this.penguin say "How many animals are in this world?" duration 2.0 add detail

WholeNumber answerNumber ← 0 this.penguin.getIntegerFromUser message: ???

message:
"hello"
Custom TextString...
"???" + "???"

Custom TextString

preview: "Enter the number of animals:"

value: Enter the number of animals:

Cancel OK

do in order count while for each in if do together each in together variable... assign

```
//comment
```

Step 4: Question #1 Continued...

Now we will determine what happens if the answer given is correct or incorrect.

Drag and drop an **If** block from the bottom of the window and set it to **true**.

The screenshot shows a Scratch-like programming environment. At the top, there's a 'declare procedure quiz' block with an 'Add Parameter...' button. Below it, a 'do in order' block contains a 'say' block with the text 'How many animals are in this world?', a duration of 2.0, and an 'add detail' button. This is followed by a 'WholeNumber' block with 'answerNumber' as the variable and a 'getIntegerFromUser' block with the prompt 'Enter the number of animals:'. Below these is an 'if' block with the condition 'true' and the text '(current value)'. The workspace is filled with various comparison and logical blocks. A red arrow points from the 'if' block in the bottom toolbar to the workspace.

Choose **Relational (WholeNumber)** in the true drop-down list.

Select “**??? == ???**” -> **answerNumber...**
-> **Custom WholeNumber** to enter in the value **5**.

5 is the correct answer to this question.

Note: The following is an explanation of all the comparison options

$a == b$	a is equal to b
$a \neq b$	a does not equal b
$a < b$	a is less than b
$a > b$	a is greater than b
$a \leq b$	a is less than or equal to b
$a \geq b$	a is greater than or equal to b

Step 4: Question #1 Continued...

For this question when the “If” statement is true, the answer is correct. So we will put our response to the correct answer first.

Click on **this.penguin** in the object tree.

Under the If:

Drag the **say** method and type: *Correct*

Drag the **turn** method and select **1**

When the if statement is false, the answer is incorrect.

Under the Else:

Drag the **say** method and type: *Sorry, that is not correct.*

```
if {answerNumber == 5} is true then
  this.penguin say "Correct" add detail
  this.penguin turn LEFT, 1.0 add detail
else
  this.penguin say "Sorry, that is not correct" add detail
```

Run your world to take the one question quiz.

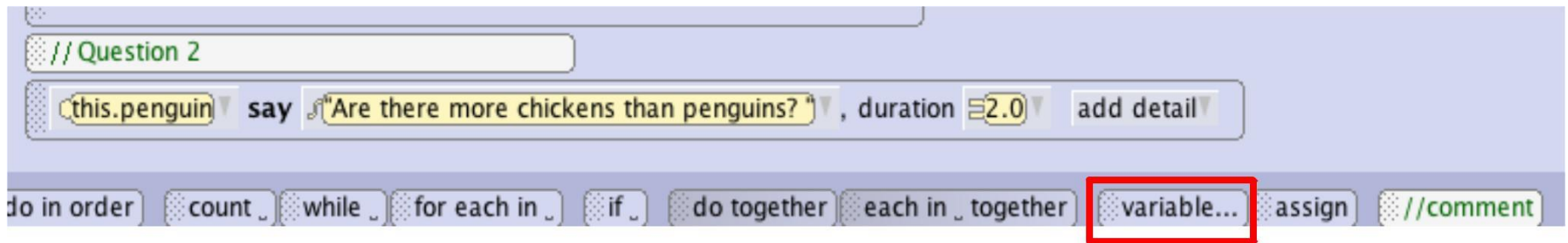


Step 5: Question #2

Now lets make a question that asks the user for a true or false response.

Drop in a comment and type in: *Question 2*

Click on **penguin** in the object tree, have it **say**: *Are there more chickens than penguins?* and set the duration to 2 seconds.

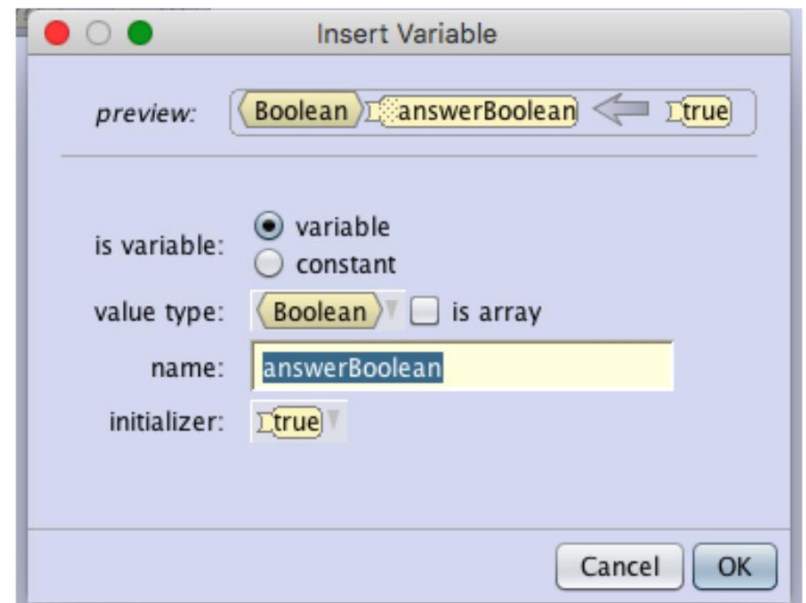


We need a variable to store the boolean information we get from the user.

Drag in **variable** button from the bottom of the window .

Name it *answerBoolean*.

Select **Boolean** and Click **OK**.



Step 5: Question #2 Continued...

In the penguin's functions, drag and drop the **“getBooleanFromUser”** over the **true**.

Type in: *Click on true and false*.

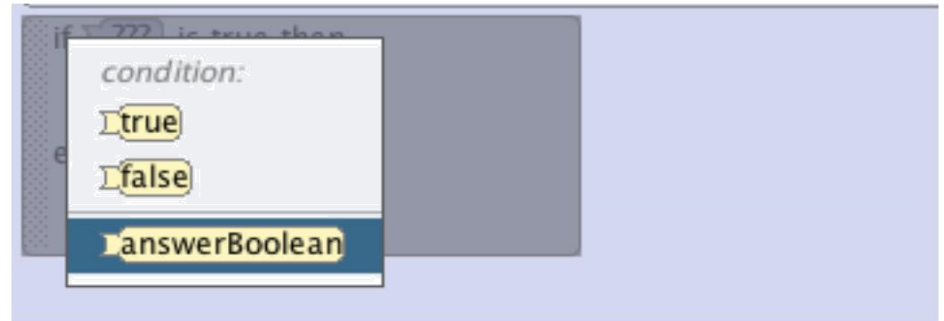
The screenshot shows the Scratch IDE interface. On the left, the 'Functions' list for 'this.penguin' is visible, with the function 'this.penguin getBooleanFromUser message: ???' circled in red. A red arrow points from this function to a 'Boolean' variable in the 'quiz' procedure script. The script contains the following code:

```
declare procedure quiz
do in order
  // Question 1
  this.penguin say "How many animals are in this world?", duration 2.0 add detail
  WholeNumber answerNumber ← this.penguin getIntegerFromUser "Enter the number of animals."
  if answerNumber == 5 is true then
    this.penguin say "Correct" add detail
    this.penguin turn LEFT, 1.0 add detail
  else
    this.penguin say "Sorry, that is not correct" add detail
  // Question 2
  this.penguin say "Are there more chickens than penguins?", duration 2.0 add detail
  Boolean answerBoolean ← this.penguin getBooleanFromUser "Click on true or false"
```

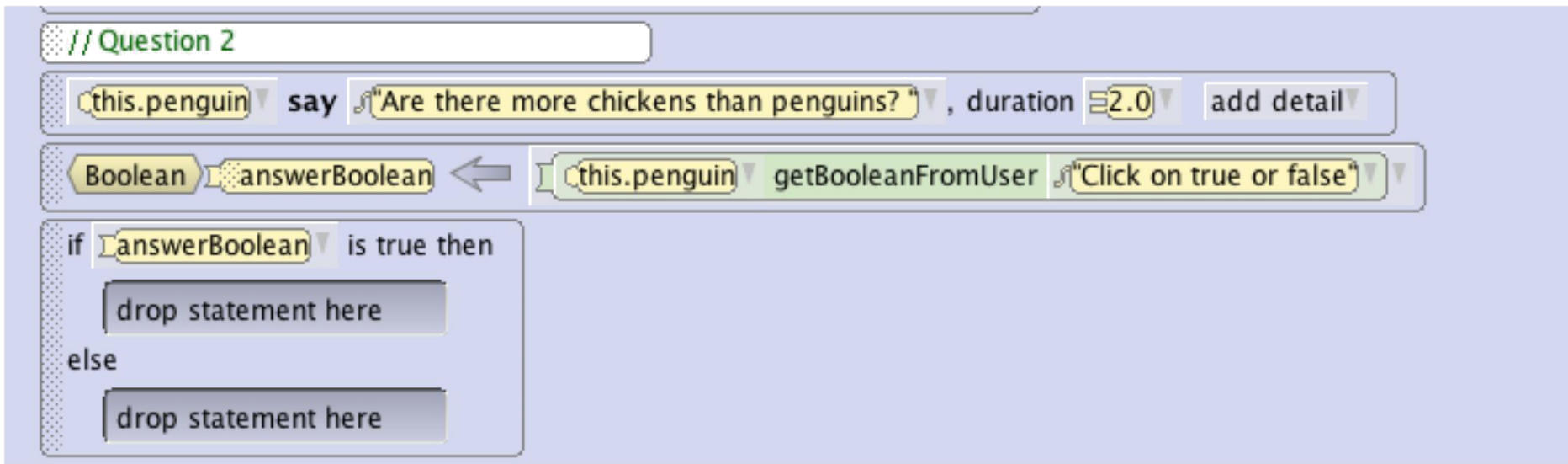
Step 5: Question #2 Continued...

Now we will add in our responses to the answer in another If/Else statement.

Drag and drop an **If/Else** statement and select **answerBoolean**.



The If question accepts just the variable because answerBoolean is a boolean type.



Step 5: Question #2 Continued...

Now complete the question #2 by adding in the procedures as shown below:

```
if answerBoolean is true then
  this.penguin say "Correct" add detail
  this.penguin move UP, 0.5 add detail
  this.penguin move DOWN, 0.5 add detail
else
  this.penguin say "Sorry, that is not correct" add detail
```

Run your world to take the two question quiz.



Step 6: Question #3

For our final question we will ask the user to enter a string.

Drop in a comment and type in: *Question 3*

Click on **penguin** in the object tree, have it **say**: *What building are we in?* and set the duration to 2 seconds.

Note: A string is a set of letters or characters. A space is considered a character so be careful if you have a space at the end of your word or sentence.

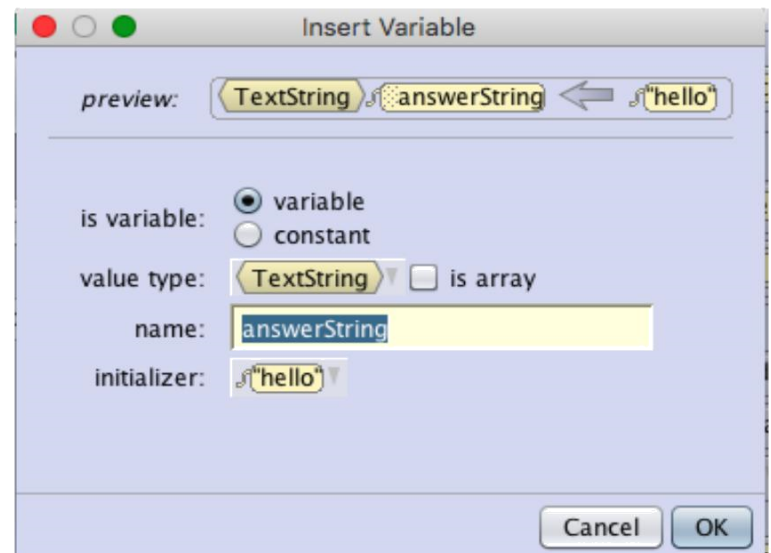


We need a variable to store the string information we get from the user.

Drag in the **variable** button.

Name it *answerString*.

Select **TextString** and **'hello'** in initializer Click **OK**.



Step 6: Question #3 Continued...

In the penguin's functions,
drag and drop the **getStringFromUser** over the 'hello' string.

Type in: *Enter acronym*.

The image shows a programming environment with two main panels. The left panel, titled 'this.penguin', has a 'Functions' tab selected. It lists several functions: 'getOpacity', 'getWidth', 'getHeight', 'getDepth', 'getBooleanFromUser', 'getStringFromUser', 'getDoubleFromUser', and 'getIntegerFromUser'. The right panel shows a script area with two question blocks. 'Question 2' contains a 'say' block with the message 'Are there more chickens than penguins?', a 'duration' of 2.0, and an 'add detail' block. Below it is a 'Boolean' variable named 'answerBoolean' connected to a 'getBooleanFromUser' block with the message 'Click on true or false'. An 'if' block follows, with a condition 'answerBoolean is true then'. Inside the 'if' block are three blocks: 'say' with 'Correct', 'move' with 'UP' and duration 0.5, and 'move' with 'DOWN' and duration 0.5. An 'else' block contains a 'say' block with 'Sorry, that is not correct'. 'Question 3' contains a 'say' block with 'What building are we in?' and a 'duration' of 2.0, followed by a 'TextString' variable named 'answerString' connected to a 'getStringFromUser' block with the message 'Enter acronym'.

Step 6: Question #3 Continued...

Now we will add in our responses to the answer in another If/Else statement.

Drag and drop an **If/Else** statement and select **true**.

Click the down arrow in **true** and select **TextString Comparison** -> **???.equalsIgnoreCase???** -> **answerString** -> **Custom TextString** then type in: **LSRC**

With **equalsIgnoreCase**, the answer is not case-sensitive

The image shows a Scratch code editor with the following code:

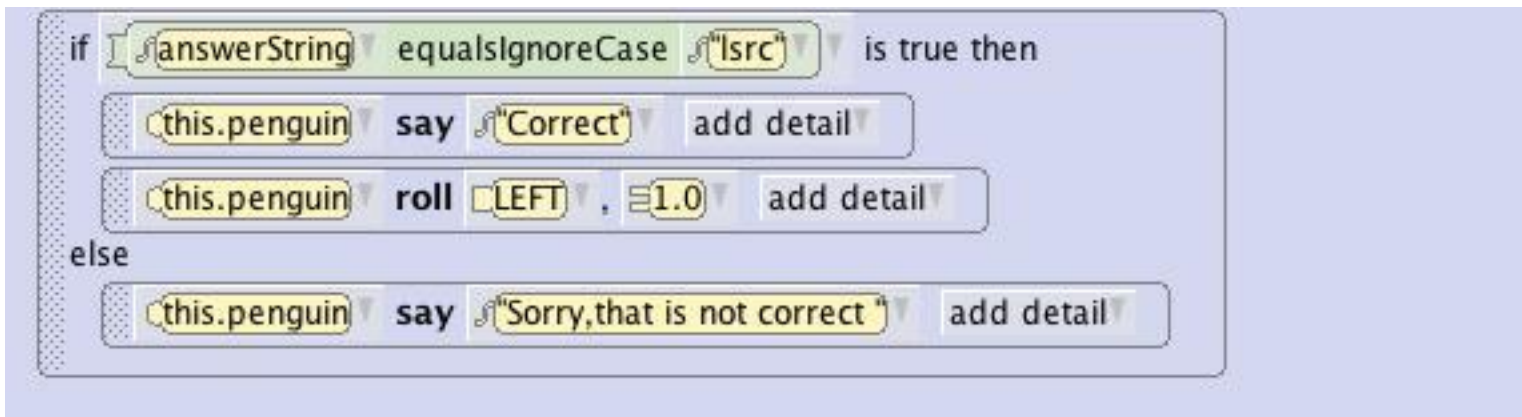
```
// Question 3
this.penguin say "What building are we in?", duration 2.0 add detail
TextString answerString ← this.penguin getStringFromUser "Enter acronym"
if true (current value)
  true
else
  nextRandomBoolean
  NOT true
  NOT ???
  BOTH true AND ???
  EITHER true OR ???
  BOTH ??? AND ???
  EITHER ??? OR ???
  Relational (DecimalNumber) { ==, !=, <, <=, >=, > }
  Relational (WholeNumber) { ==, !=, <, <=, >=, > }
  Relational (SThing) { ==, != }
  TextString Comparison
do in
  answerBoolean
```

The **TextString Comparison** block is expanded, showing the following configuration:

- Operator: **equalsIgnoreCase**
- Input 1: **hello**
- Input 2: **Custom TextString...**
- Variable: **answerString**

Step 6: Question #3 Continued...

Click on penguin in the object tree and add in the responses for when the answer is correct or incorrect as shown below.



```
if [answerString equalsIgnoreCase "Isrc"] is true then
  this.penguin say "Correct" add detail
  this.penguin roll LEFT, 1.0 add detail
else
  this.penguin say "Sorry, that is not correct" add detail
```

The image shows a Scratch code editor snippet with a light blue background. It features an 'if-else' conditional block. The 'if' condition is '[answerString equalsIgnoreCase "Isrc"] is true then'. The 'then' branch contains two actions: 'this.penguin say "Correct" add detail' and 'this.penguin roll LEFT, 1.0 add detail'. The 'else' branch contains one action: 'this.penguin say "Sorry, that is not correct" add detail'. Each action is contained within a rounded rectangular box with a light yellow background and a small 'add detail' button on the right.

Now play your world and take the quiz.

Part 2: Forcing the Answer

In Question 1, if the user mistypes the answer, we want the program to ask the user for the answer again until it is correct. Rewrite question 1 to add a while loop and force the user into the loop with a wrong value to start with.

```
// Question 1
this.penguin say "How many animals are in this world?" , duration 2.0 add detail
WholeNumber answerNumber ← 1
while answerNumber ≠ 5 is true
  answerNumber ← this.penguin getIntegerFromUser "Enter the number of animals:"
  if answerNumber == 5 is true then
    this.penguin say "Correct" add detail
    this.penguin turn LEFT 1.0 add detail
  else
    this.penguin say "Sorry, that is not correct" add detail
loop
```

The image shows a Scratch script for 'Question 1'. The script starts with a comment '// Question 1'. The first block is a 'say' block from 'this.penguin' with the text 'How many animals are in this world?', a duration of 2.0, and an 'add detail' button. The second block is a 'WholeNumber' block for 'answerNumber' with a value of 1. The third block is a 'while' loop with the condition 'answerNumber ≠ 5'. Inside the loop, there is a 'getIntegerFromUser' block from 'this.penguin' with the text 'Enter the number of animals:'. Below this is an 'if' block with the condition 'answerNumber == 5'. If true, it contains two blocks: a 'say' block with the text 'Correct' and a 'turn' block with the text 'LEFT' and a value of 1.0. If false, it contains a 'say' block with the text 'Sorry, that is not correct'. The loop is labeled 'loop' at the bottom left. Two values, 2.0 and 1.0, are highlighted with black boxes.

Part 3: Limiting Number of Trials

To limit the number of trials to 3 for example, create a new variable named **numberOfTrial**, create a while loop that allows the question to show up as long as the students tries less than 3 times. Each time they answer incorrectly, increment **numberOfTrial** by 1. If they answer, correctly, set **numberOfTrial** to a number greater than 3 to break out of the while loop (here we choose 5)

```
// Question 3 Limit to 3 Trials Only

this.penguin say "What building are we in?", duration 2.0 add detail

WholeNumber numberOfTrial ← 1

while numberOfTrial ≤ 3 is true
  TextString answerString ← this.penguin getStringFromUser "Enter acronym"
  if answerString equalsIgnoreCase "Isrc" is true then
    this.penguin say "Correct" add detail
    this.penguin roll LEFT, 1.0 add detail
    numberOfTrial ← 5
  else
    this.penguin say "Sorry, that is not correct" add detail
    numberOfTrial ← 1 + numberOfTrial

loop
```

The image shows a Scratch script designed to limit the number of trials for a question. It starts with a comment: "// Question 3 Limit to 3 Trials Only". The first block is a "say" block from the "this.penguin" sprite, with the text "What building are we in?", a duration of 2.0 seconds, and an "add detail" option. This is followed by a "WholeNumber" block that initializes a variable named "numberOfTrial" to the value 1. A "while" loop is then created, which will execute as long as the condition "numberOfTrial ≤ 3" is true. Inside the loop, a "TextString" block asks the user for an "acronym" using the "getStringFromUser" method. An "if" block checks if the user's answer, "answerString", equals "Isrc" (ignoring case). If true, the penguin says "Correct", rolls the dice to the left for 1.0 seconds, and sets "numberOfTrial" to 5, which will break the loop. If false, the penguin says "Sorry, that is not correct" and increments "numberOfTrial" by 1. The loop is labeled "loop" at the bottom.

Part 4: Score Reporting

Challenge: To report score to user, we have to create a local variable **score** for the quiz procedure, initialize it to 0, and increment it by 1 every time the student answer a question correctly. At the end, have the penguin report the score.

Part 4: Score Reporting

To report the score in the end, drag the penguin **say** procedure in and type “You got ”.

Click on the down arrow next to the string you just typed and select “You got +???-> WholeNumber -> score

The screenshot shows a Scratch-like programming environment with a script for a penguin character. The script includes a loop for rolling a die and updating a score. A context menu is open over the 'say' block, showing the path 'You got + ??? -> WholeNumber -> score'.

The script contains the following blocks:

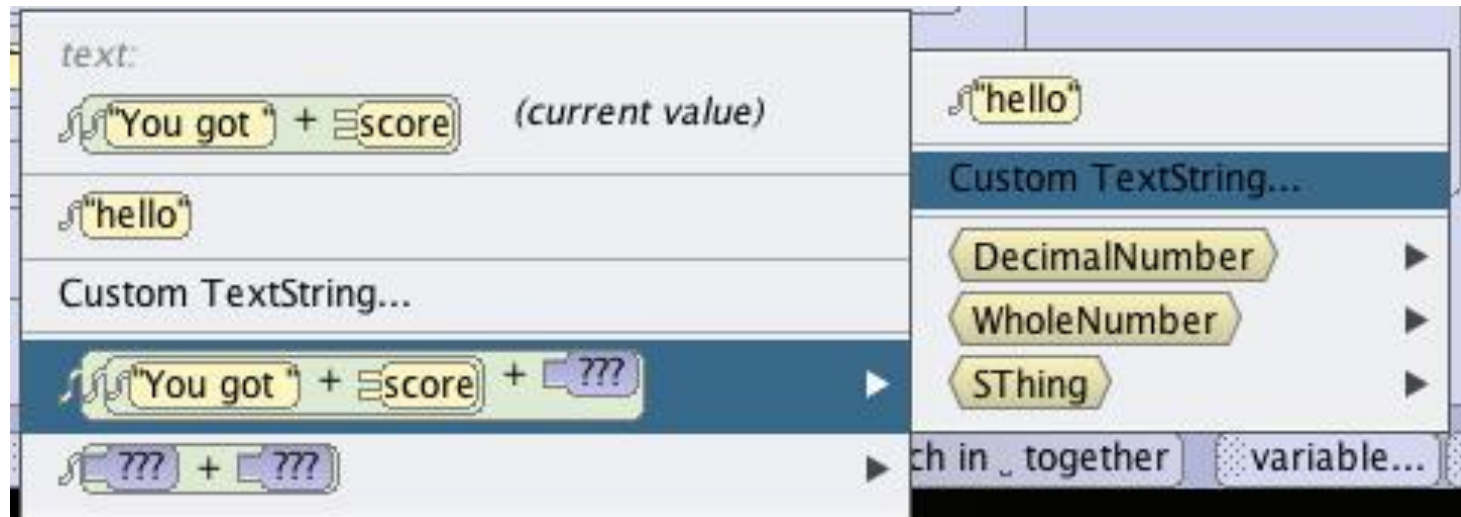
- if [answerString equals ignore case] is true then
 - this.penguin say "Correct" add detail
 - this.penguin roll LEFT, 1.0 add detail
 - numberOfTrial ← 5
 - score ← 1 + score
- else
 - this.penguin say "Sorry that is not correct" add detail
 - numberOfTrial ← 5
- loop
- this.penguin say "You got " + ???

The context menu for the 'say' block shows the following options:

- text:
 - "You got " (current value)
 - "hello"
 - Custom TextString...
 - "You got " + ???
 - ??? + ???
- Custom WholeNumber...
- numberOfTrial
- answerNumber
- score
- SThing

Part 4: Score Reporting

Click on the down arrow next to the “You got “ + score string and select “You got “ + score + ??? -> Custom TextString and Type “ questions correct”



Quiz Procedure code

```
declare procedure quiz Add Parameter...
do in order
  WholeNumber score ← 0
  //Question 1 Force Answer
  this.penguin say "How many animals are in this world?", duration 2.0 add detail
  WholeNumber answerNumber ← 1
  while answerNumber ≠ 5 is true
    answerNumber ← this.penguin getIntegerFromUser "Enter the number of animals."
    if answerNumber == 5 is true then
      this.penguin say "Correct" add detail
      this.penguin turn LEFT, 1.0 add detail
      score ← 1 + score
    else
      this.penguin say "Sorry, that is not correct" add detail
  loop
  //Question 2 One Time Try
  this.penguin say "Are there more chickens than penguins?", duration 2.0 add detail
  Boolean answerBoolean ← this.penguin getBooleanFromUser "Click on true or false"
  if answerBoolean is true then
    this.penguin say "Correct" add detail
    this.penguin move UP, 0.5 add detail
    this.penguin move DOWN, 0.5 add detail
    score ← 1 + score
  else
    this.penguin say "Sorry, that is not correct" add detail
  //Question 3 Limit to 3 Trials Only
  WholeNumber numberOfTrial ← 1
  while numberOfTrial ≤ 3 is true
    TextString answerString ← this.penguin getStringFromUser "Enter acronym"
    if answerString equalsIgnoreCase "lsrc" is true then
      this.penguin say "Correct" add detail
      this.penguin roll LEFT, 1.0 add detail
      numberOfTrial ← 5
      score ← 1 + score
    else
      this.penguin say "Sorry, that is not correct" add detail
      numberOfTrial ← 1 + numberOfTrial
  loop
  this.penguin say "You got " + score + " question correct" add detail
```