A Simple Quiz for Alice 3.2:
Ask User Functions.

By Lana Dyck
under the direction of Professor Susan Rodger
Duke University
June 2009, added Part 2 July 2011,
Edited for Alice 3 use and Part 3,4 added by Anh Trinh July 2016
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.
Create a new scene procedure that we will use later to hold the instructions for the quiz.

Let's 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.
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.
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 into the method.
**Step 4: Question #1**

Type in the comment editor:  

```// 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.

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:
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`.

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 ≠ b`  a does not equal b
- `a < b`  a is less than b
- `a > b`  a is greater than b
- `a ≤ b`  a is less than or equal to b
- `a ≥ b`  a is greater than or equal to b
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.*

Run your world to take the one question quiz.
Step 5: Question #2

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

Type in: Click on true and false.
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.

```cpp
// Question 2

this.penguin say "Are there more chickens than penguins?", duration = 2.0

Boolean answerBoolean

if answerBoolean is true then
    drop statement here
else
    drop statement here
```
Now complete the question #2 by adding in the procedures as shown below:

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.

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.

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.
In the penguin's functions, drag and drop the `getStringFromUser` over the ‘hello’ string.

Type in: **Enter acronym.**
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
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 {  // is false then
    this.penguin.say("Sorry, that is not correct", add detail);
}
```

Now play your world and take the quiz.
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.
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).
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.
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"
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”
declare procedure quiz Add Parameter...

do in order

WholeNumber := score ← 0

// Question 1 Force Answer

text := "How many animals are in this world?" add detail

while (answerNumber ≠ 5) is true

answerNumber ← this.penguin getIntgerFromUser ("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 ← score + 1

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?" 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

score ← score + 1

else

this.penguin say "Sorry, that is not correct" add detail

loop

// Question 3 Limit to 3 Trials Only

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

WholeNumber := numberOfTrial ← 3

while (numberOfTrial ≤ 3) is true

TextString := answerString ← this.penguin getStringFromUser ("Enter acronym")

if (answerString equalsIgnoreCase ("QSC") is true then

this.penguin say "Correct" add detail

this.penguin roll (LEFT), 1.0 add detail

numberOfTrial ← 5

score ← score + 1

else

this.penguin say "Sorry, that is not correct" add detail

loop

(this.penguin say "You got " + score) + ("question correct") add detail