Using Functions in Alice

By Jenna Hayes
Under the direction of Professor Susan Rodger
Duke University
July 2008

Step 1: Understanding Functions

1. Download the starting world that goes along with this tutorial. We will be using functions. A function in Alice is basically a question about information in your Alice world that Alice answers. Click on the penguin object in the object tree. Then click on the functions tab. You will see a LONG list of functions. Scroll down and look at the functions under size.

2. Each of these functions asks a question about the penguin, and then keeps the answer so that you can use it in your Alice world. These functions can be very useful in Alice. What if, for example, you want to make something move up and stand on top of the penguin’s head? You don’t know exactly how tall the penguin is. But Alice does!

Step 2: Using the True or False Functions

Some functions in Alice are statements to which the answer is either true or false, like the one on the previous page that says, “penguin is taller than.” We want to know whether the penguin is taller than the normalGuy. It’s almost impossible to tell just by looking at them, because their heights are so close together. So we will use a function to know for sure. First drag a Do in order into your method editor. Then, inside that, drag in an If Else statement. Select true when you drop it. We will replace this later.

We are going to add commands so that when the Alice world starts, if the penguin is taller, it will say “Hah! I’m taller!”, but if the normalGuy is taller, he will say, “Hah! I’m taller!”.
Step 2: cont.

Here is where you put the question that is either true or false. For us, that will be `penguin is taller than normalGuy`. Since it is currently set at `true`, that means this If Else says, “If this statement right here is true, do whatever commands are right under it.”

This is where you put whatever you want to happen if the answer to your question is true. This is where we’ll tell the penguin to say “Hah! I’m taller!”

This is where you put whatever you want to happen if the answer to your question is NOT true. If it is not true, it will skip everything above the Else, and go straight to whatever is here. This is where we’ll tell the person to say “Hah! I’m taller!”, because if our statement is false, and the penguin is NOT taller, that means the person is taller!

Step 2: cont.

So let’s construct our question. Click on `penguin` in the object tree and then click on functions. Find the `penguin is taller than` function, and drag it over the word `true` in your If Else statement. When the little menu pops up, click on `normalGuy`. Your code will look like this:

This is where you put whatever you want to happen if the answer to your question is NOT true. If it is not true, it will skip everything above the Else, and go straight to whatever is here. This is where we’ll tell the person to say “Hah! I’m taller!”, because if our statement is false, and the penguin is NOT taller, that means the person is taller!

Step 2: cont.

Now click on `penguin` in the object tree, and then click on methods. Find `penguin say`, and drag it into the method editor under If. When the menu pops up, click on other, and type in “Hah! I’m taller!” Your code will look like this:

Now click on `normalGuy` in the object tree, and click on methods. Find `normalGuy say` and put under the Else part of your If Else statement. Tell him to say “Hah! I’m taller!” Your code will look like this:

Now play your world. Who is taller, the penguin or the person?
Step 3: Using Number Functions

Now we’re going to use one of the functions that is a question whose answer is a number. We’re going to make the penguin move right up to the man and give him a hug. The only problem is, we don’t know how far to tell him to move! That’s why we’ll use a function!

First we’ll tell the man to say something. He’s tired of competing with the penguin about their height, and he wants to be friends. Click on normalGuy in the object tree, then click on methods. Find normalGuy say, and drag it under your if else statement. Make him say “Let’s be friends. Give me a hug, penguin!” Your code will look like this:

```
world.my first method
world.my first method no parameters
No variables

Do in order

Do if

<table>
<thead>
<tr>
<th>condition</th>
<th>block</th>
</tr>
</thead>
<tbody>
<tr>
<td>penguin is taller than normalGuy</td>
<td>penguin say &quot;Hi! I’m taller! more...&quot;</td>
</tr>
<tr>
<td>else</td>
<td>normalGuy say &quot;Hi! I’m taller! more...&quot;</td>
</tr>
<tr>
<td>normalGuy say &quot;Let’s be friends. Give me a hug, penguin! more...&quot;</td>
<td></td>
</tr>
</tbody>
</table>
```

Your code will look like this:

```
world.my first method
world.my first method no parameters
No variables

Do in order

Do if

<table>
<thead>
<tr>
<th>condition</th>
<th>block</th>
</tr>
</thead>
<tbody>
<tr>
<td>penguin is taller than normalGuy</td>
<td>penguin say &quot;Hi! I’m taller! more...&quot;</td>
</tr>
<tr>
<td>else</td>
<td>normalGuy say &quot;Hi! I’m taller! more...&quot;</td>
</tr>
<tr>
<td>normalGuy say &quot;Let’s be friends. Give me a hug, penguin! more...&quot;</td>
<td></td>
</tr>
</tbody>
</table>
```

Now we want the penguin to move right up to the guy to hug him. Click on penguin in your object tree and then click on methods. Find the penguin move and drag it into your method editor under everything else. We don’t know exactly how far yet, so just put 1 meter for now. Your code will look like this:

```
world.my first method
world.my first method no parameters
No variables

Do in order

Do if

<table>
<thead>
<tr>
<th>condition</th>
<th>block</th>
</tr>
</thead>
<tbody>
<tr>
<td>penguin is taller than normalGuy</td>
<td>penguin say &quot;Hi! I’m taller! more...&quot;</td>
</tr>
<tr>
<td>else</td>
<td>normalGuy say &quot;Hi! I’m taller! more...&quot;</td>
</tr>
<tr>
<td>normalGuy say &quot;Let’s be friends. Give me a hug, penguin! more...&quot;</td>
<td></td>
</tr>
<tr>
<td>penguin move forward 1 meter</td>
<td>more...</td>
</tr>
</tbody>
</table>
```

Try playing your world. What happens? The penguin moves too far, into the body of the normalGuy. It would be nice if the penguin would stop about 0.5m before normalGuy. We can select math, followed by “-”, followed by a number. Click here to apply math.

```
world.my first method
world.my first method no parameters
No variables

Do in order

Do if

<table>
<thead>
<tr>
<th>condition</th>
<th>block</th>
</tr>
</thead>
<tbody>
<tr>
<td>penguin is taller than normalGuy</td>
<td>penguin say &quot;Hi! I’m taller! more...&quot;</td>
</tr>
<tr>
<td>else</td>
<td>normalGuy say &quot;Hi! I’m taller! more...&quot;</td>
</tr>
<tr>
<td>normalGuy say &quot;Let’s be friends. Give me a hug, penguin! more...&quot;</td>
<td></td>
</tr>
<tr>
<td>penguin move forward 1 meter</td>
<td>more...</td>
</tr>
</tbody>
</table>
```

Your code for this line will then look like this:

```
penguin move forward (penguin distance to normalGuy - 0.5) more...```
Step 4: Finishing Up

The last thing you need to do is make the penguin hug the guy. Click on penguin in the object tree, and then click on the methods tab. Find the hug method (near the top of the methods) and drag it into your method editor under everything else. Your final code will look like this:

```
world:my first method
world:my first method  No parameters     remote:use parameters
No variables

: if penguin = in front of normalGuy then
  penguin = say Hello! I'm tall! more...
else
  normalGuy = say Hello! I'm tall! more...
end

normalGuy = say Let's be friends. Give me a hug, penguin! more...

penguin = move forward ( penguin = distance to normalGuy = 0.5 ) more...
penguin: hug
```

Now play your world and watch the magic!

Now you know the basics of using functions. We only used two functions in this tutorial, but there are MANY functions in the Alice world that can be useful. Feel free to explore the uses of other functions.