Restricting Events
Creating Conditional Events in Alice

By Henry Qin
under the direction of
Professor Susan Rodger
Duke University
July 2008

Good Job! Now press enter to start the next task.
Step 1: Testing the World

Download the Alice World that goes along with this tutorial.

*The world contains the beginnings of a game in which you have to find a chicken’s invisible chick, click on it, and then press enter to go to the next task.*

Play the world and follow the instructions. Once you have found the chick and he has appeared, press enter. This is how the game should work. The problem is, if you press enter before finding the chick, it will go to the second task anyway.

Play the world again and press enter without even trying to find the chick. Your camera will go to the next task, even though the first task has not finished.

*This tutorial will teach you how to fix this problem, and make your chicken game work like a real game, so that you can’t move on until you’ve finished the first task.*
Step 2: Creating a Boolean Variable

First we need to create a boolean (true or false statement) variable that will be true if the first task is finished, and false if it is not.

Click on world in the object tree and then go to the properties pane and click on create new variable.

When the variable box pops up, name your variable taskOneCompleted. Make sure you have selected Boolean, and set your variable to false, because the variable should not be true until the first task is completed. Click OK when you’re done.
Now, we need to include code somewhere that changes `taskOneCompleted` to true when the chick is clicked on. For this, we want to look at the `foundChick` method, because as you can see in your events editor, this method runs when the chick is clicked on. Go to the world’s methods pane and click on edit next to `foundChick`.

Go back to the properties pane and find `taskOneCompleted`. Drag and drop it to the bottom of your `foundChick` method. Select true.

Now `taskOneCompleted` will be set to true when the chick is clicked on.
Now we need to use the information that the `boolean` variable tells us; we need a method that tells Alice to only go to the second task if `taskOneCompleted` is `true`. We will create a new method for this.

Go back to the `methods` pane, and click on `create new method`. Name it `reactToEnter`.

First, drag an `If Else` statement into `reactToEnter` and set it to `true`. 
Step 5: Building the Method

Now go back to the world properties pane and find the taskOneCompleted variable. Drag and drop it over where your If Else statement says true.

Go to the world methods pane and find the nextTask computer. Drag and drop it into your If Else statement.
Now when we hit **Enter** instead of doing **nextTask**, we want Alice to do **reactToEnter**.

Drag and drop **reactToEnter** from the **methods** pane to where it says **nextTask** onto the **Enter** event.

Now play the world, and try to press **Enter** without first clicking on the invisible chick. Nothing happens! Only after you have found the chick will you be able to continue.
You can use this tutorial to create other events that are restricted. The trick is to create a new variable and then a control method with an if/else that will only let the right method play if the conditions in the if are met. You can use the variables to keep track of which tasks have been completed by setting them to true from false after a task is complete. Enjoy creating games in Alice!