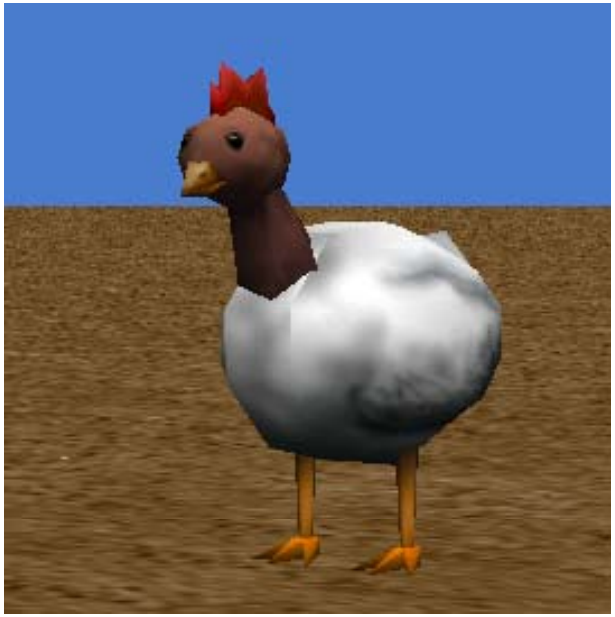
Making Events Better: Restricting an Event with a Conditional



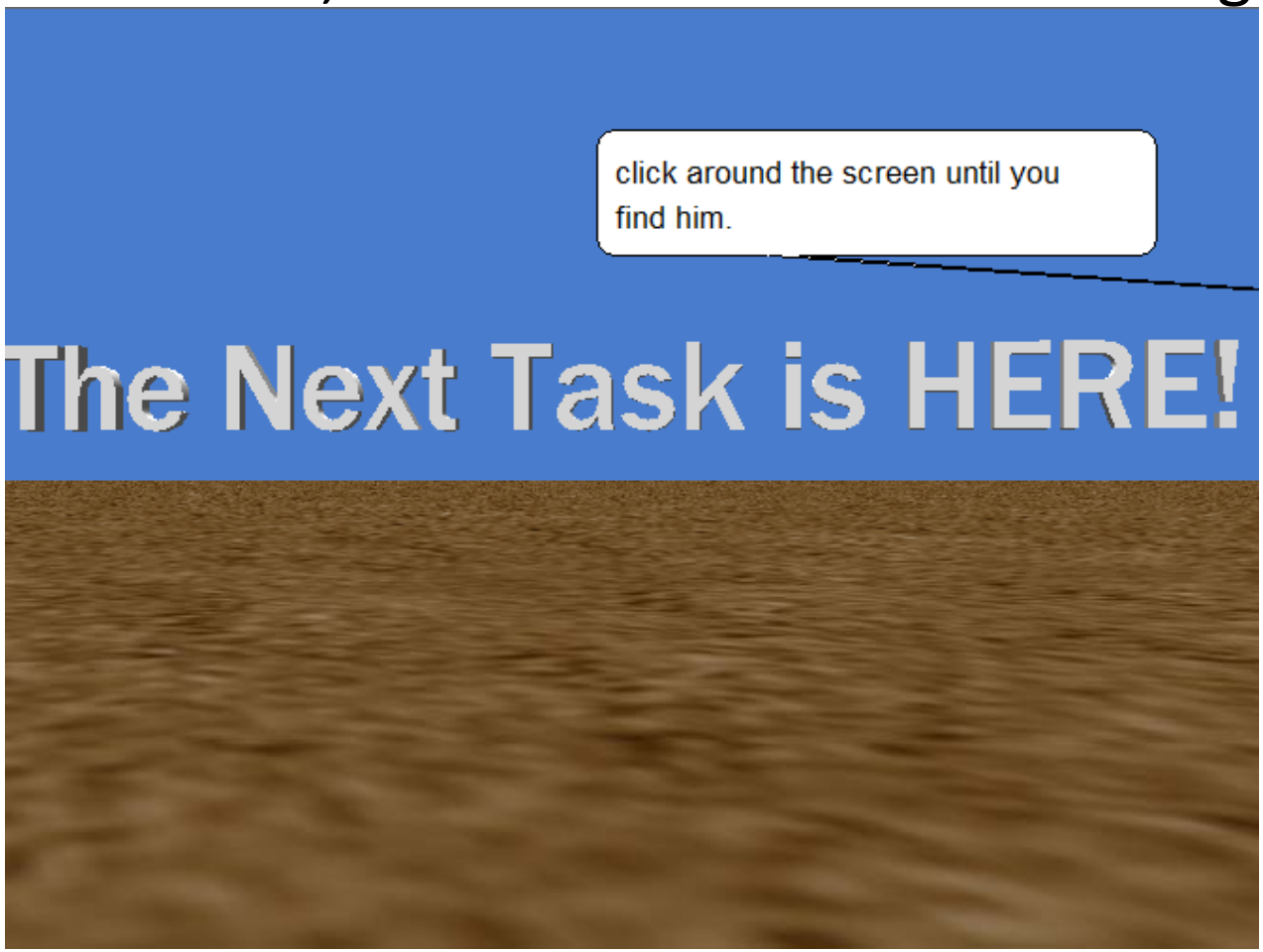
By Henry Qin, edited by Jenna Hayes
Under the direction of Professor Susan Rodger
Duke University, August 2008

Getting Started

Download the starting world that goes 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. The problem is, if you press enter *before* finding the chick, it will go to the second task anyway. 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.

Testing it Out

Try playing the world and pressing enter without even trying to find the chick. Your camera will go to the next task, but the chicken will not even be finished telling you about the first task, which will result in something like this:

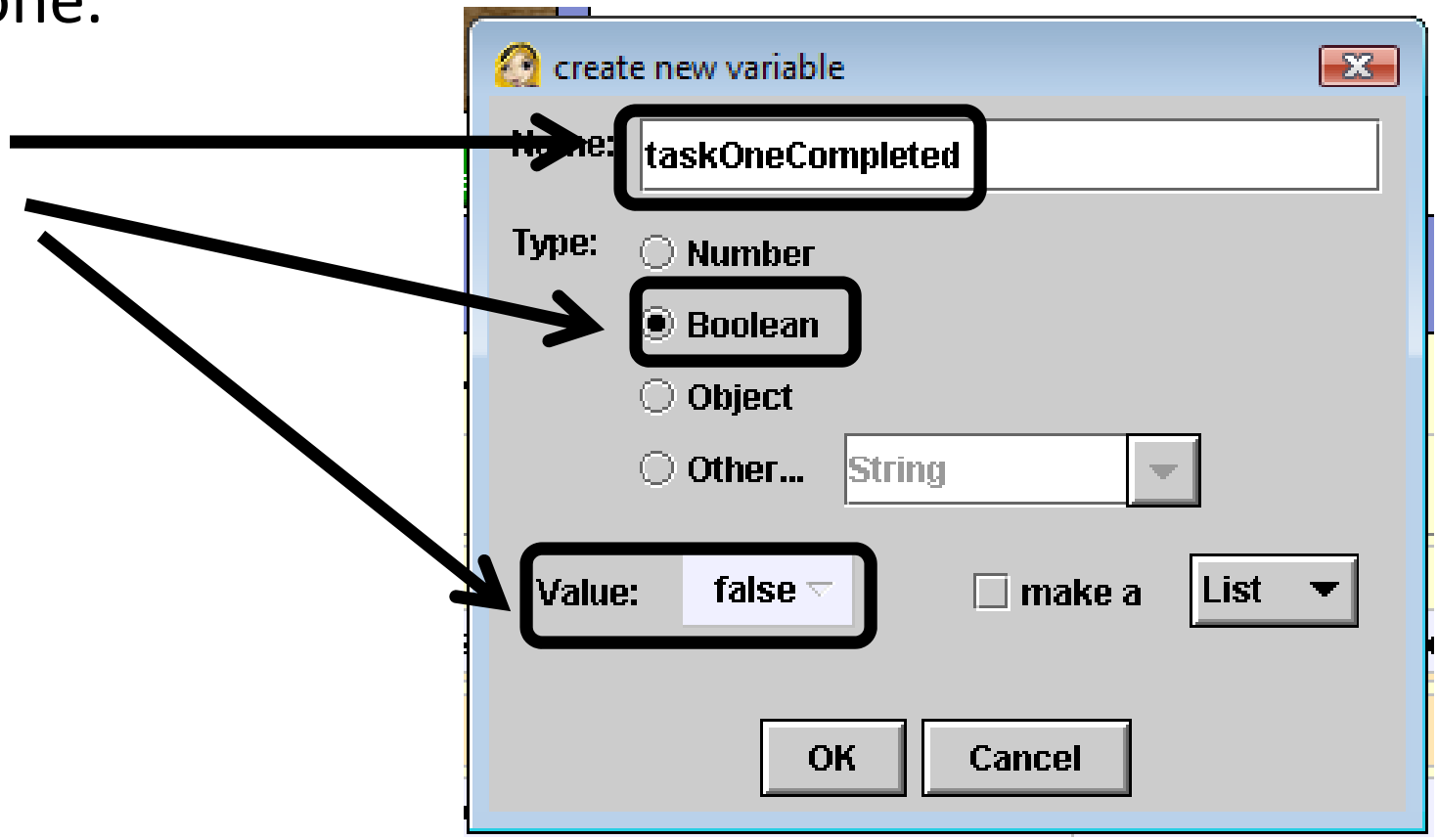


Creating a Boolean Variable

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.

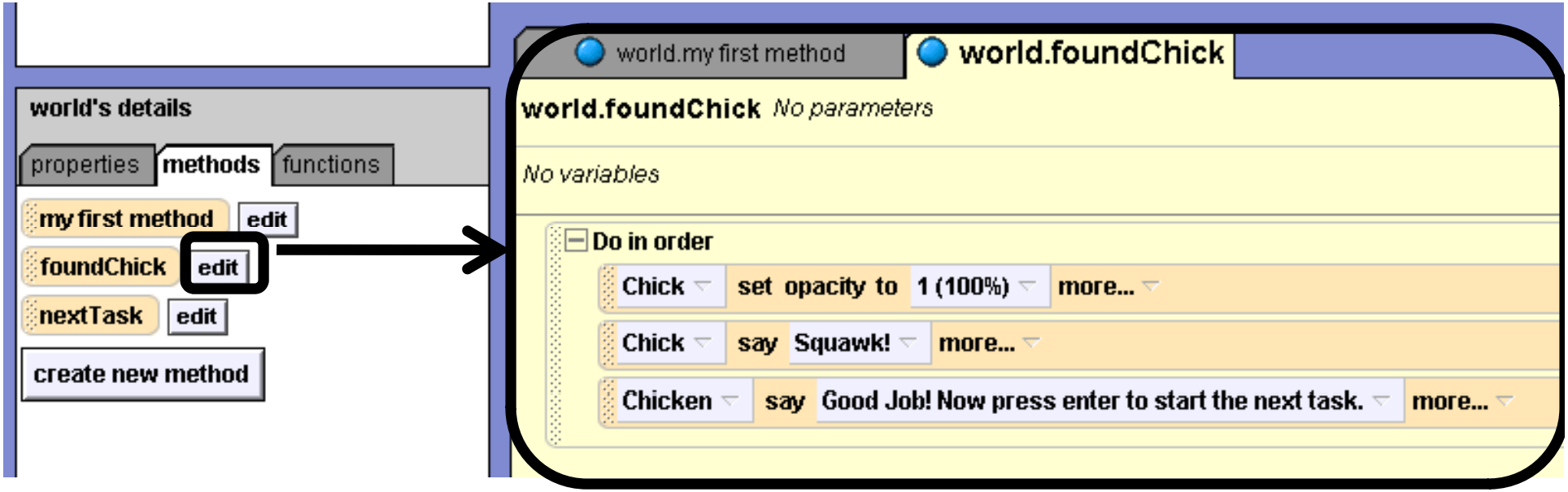
Creating a Boolean Variable

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 Okay when you're done.



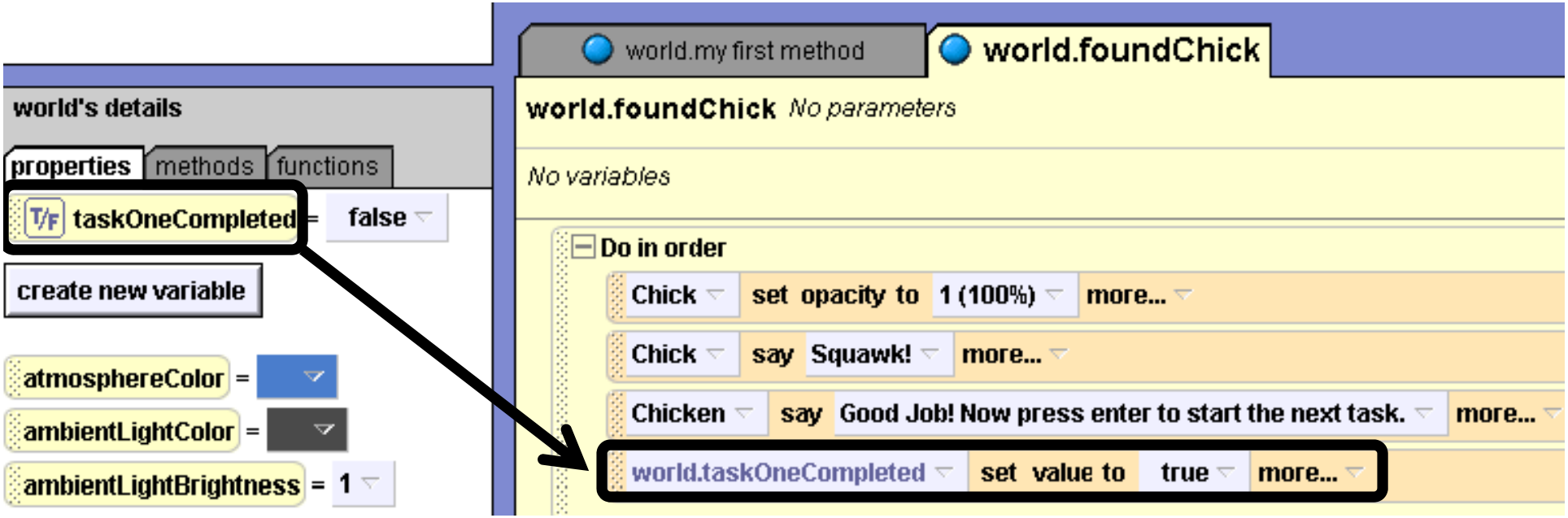
Setting the Variable

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.



Setting the Variable

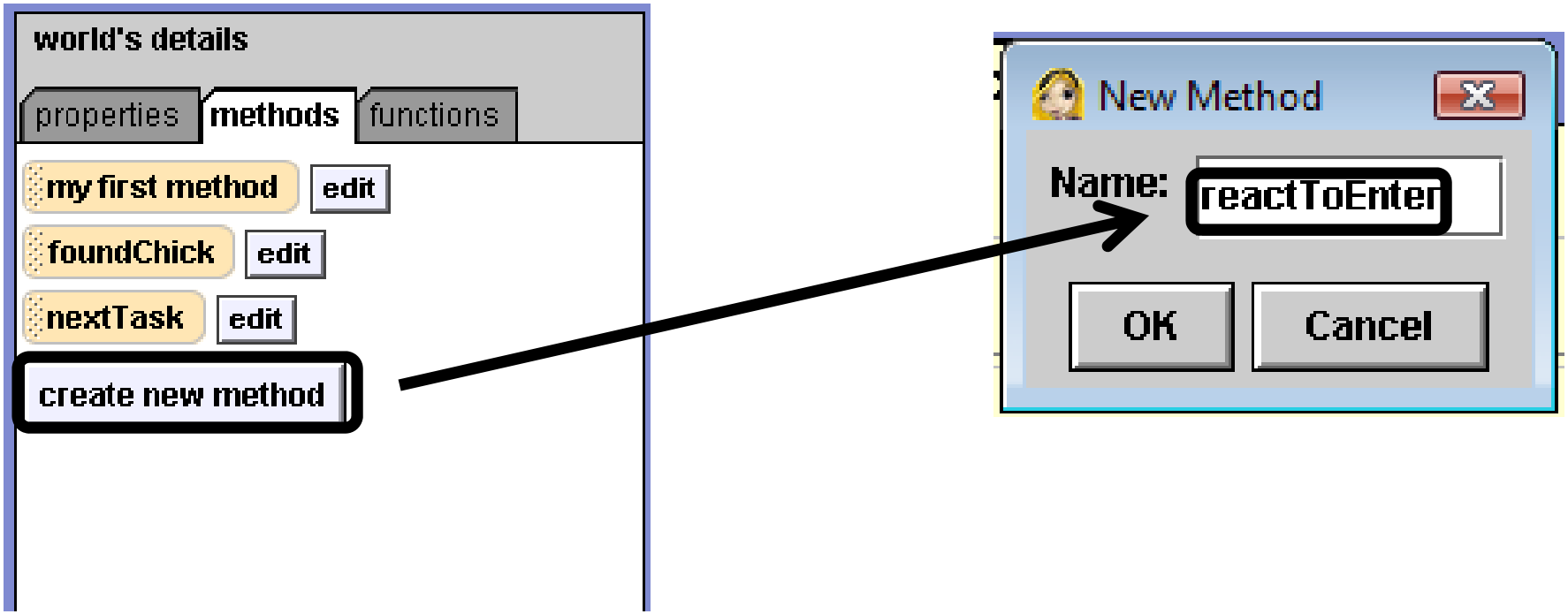
Go back to the properties pane and find taskOneCompleted. Drag and drop it to the bottom of your foundChick method. Select true.



Now taskOneCompleted will be set to true when the chick is clicked on.

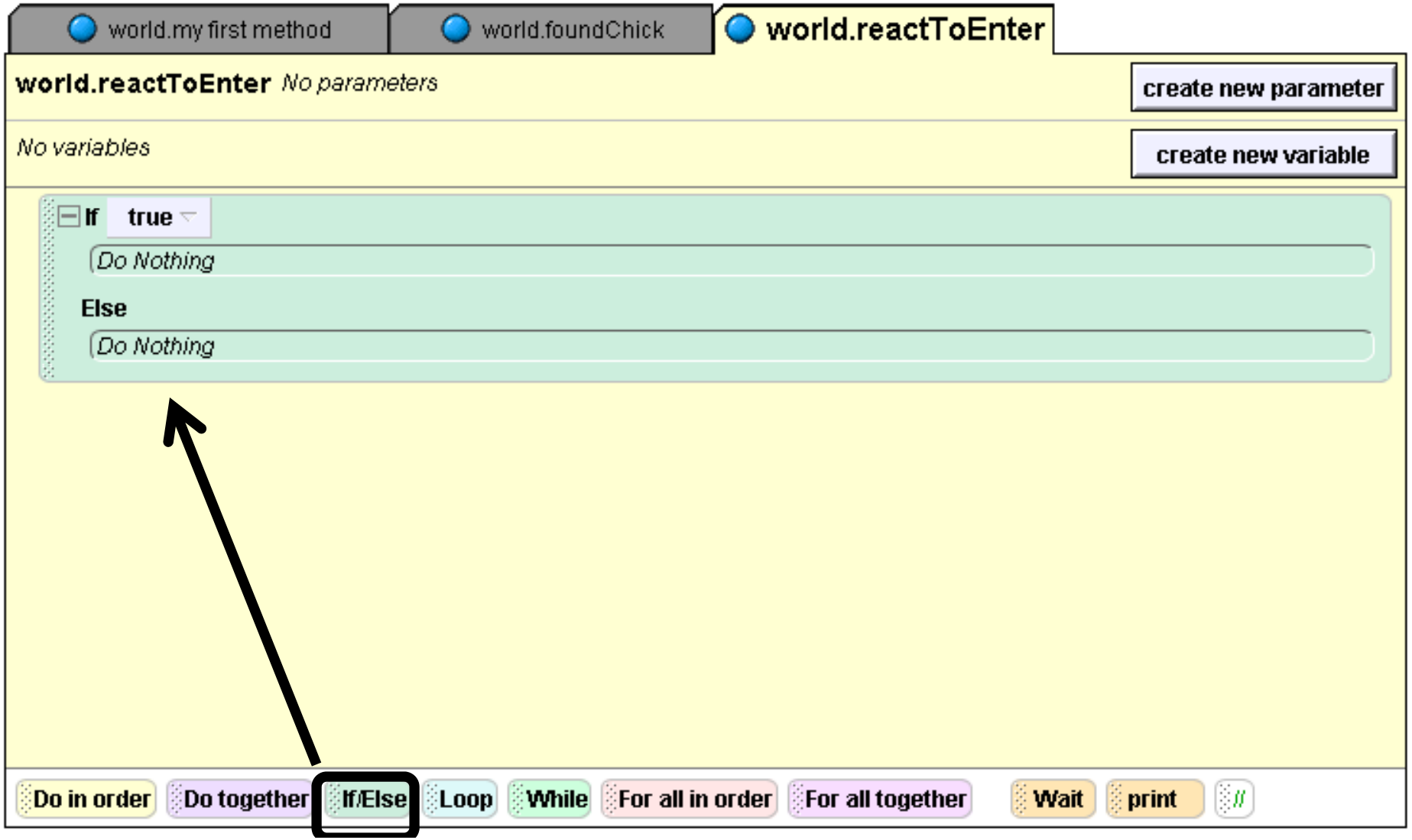
Using the Variable

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.



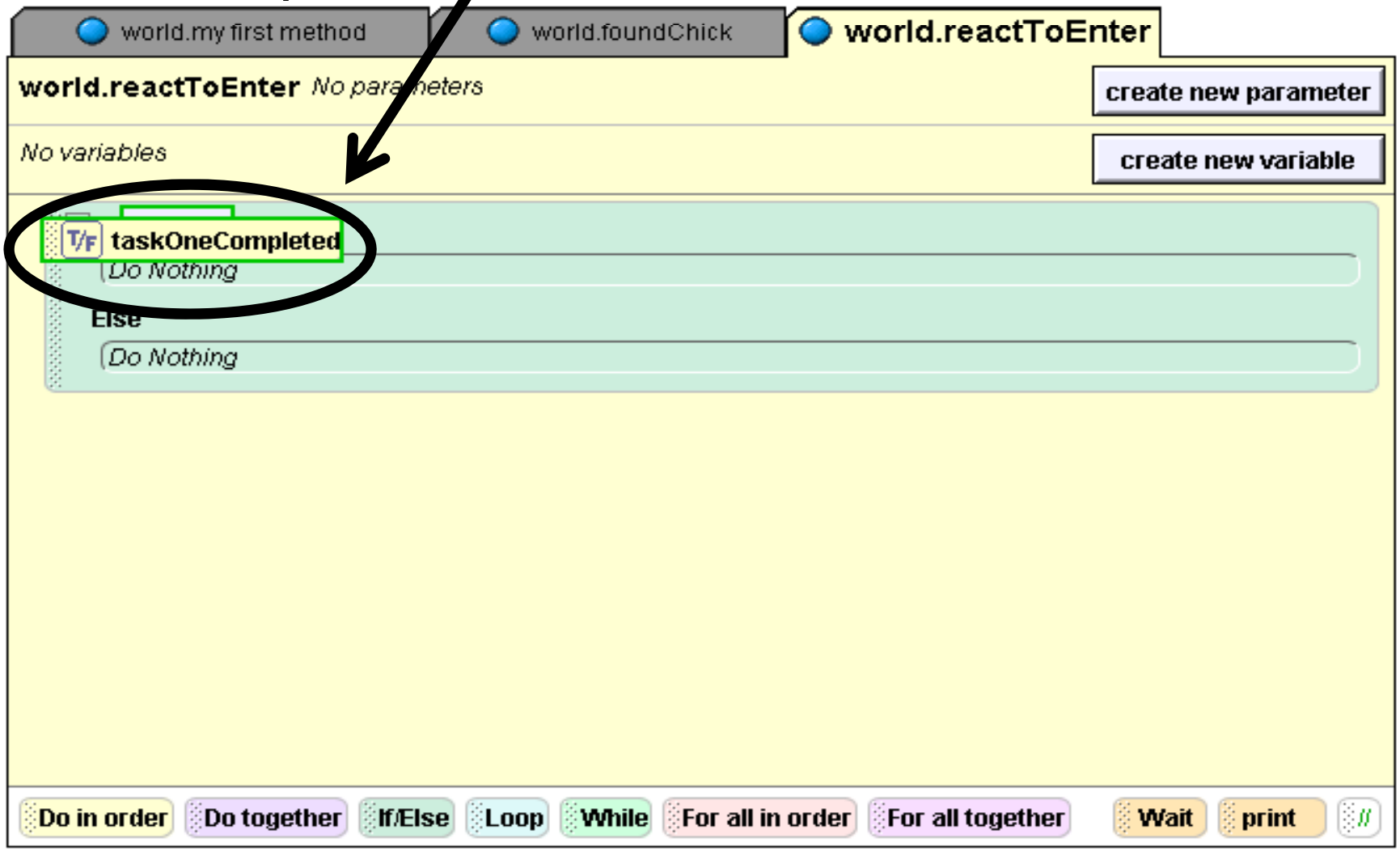
Building reactToEnter

First, drag an If Else statement into reactToEnter and set it to true.



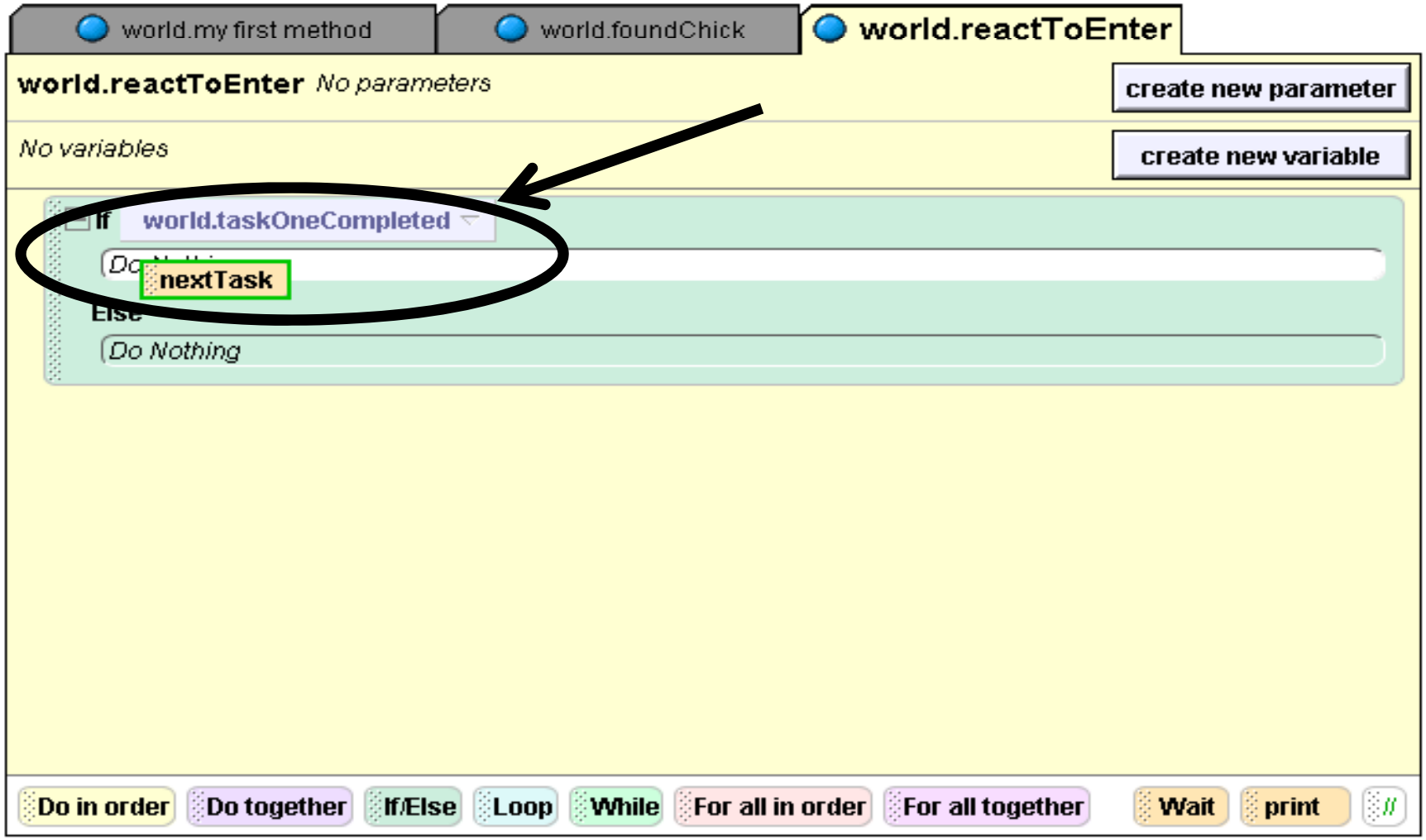
Building reactToEnter

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



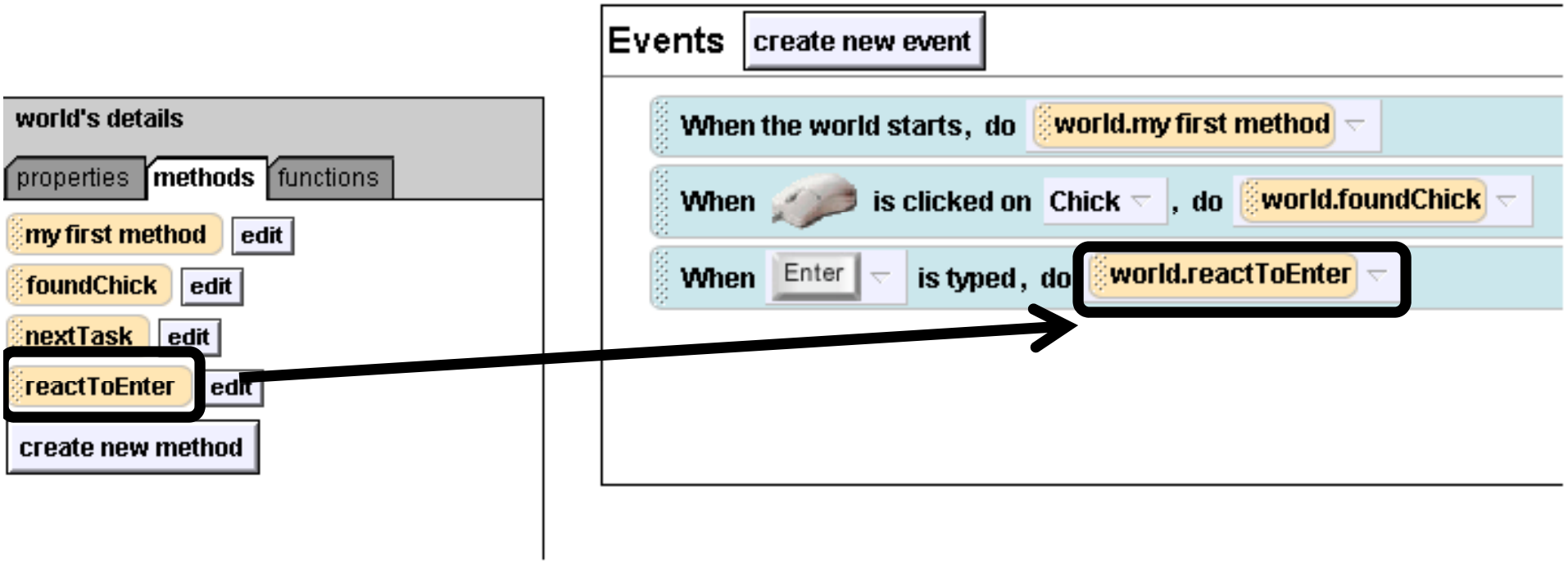
Building reactToEnter

Go back to the methods pane and find nextTask. Drag and drop it into your If Else statement.



Changing the Enter Event

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 on the Enter event.



Testing it Out

Now play the world, and try to press Enter without first clicking on the invisible chick. It won't work! Only after you have found the chick will you be able to continue.