Creating Functions

Deborah Nelson

Duke University

Professor Susan Rodger

July 22, 2008

Loading the World

- Download the file that we'll be working with today
- It is named fireStart.a2w
- Save it in a directory that you can find again, and then start Alice and open the world. NOTE: You cannot double-click the file to open it;
- Windows will not know what to use, and even if you select Alice from a list of programs, the loading will fail.

Play the World

- In this world, the user is supposed to burn everything that is in the junkyard.
- Play the world
- There is an error. Because whatever the user clicks on burns – even the animals and the ground.
- We only want the things in the junkyard to be burned. So we need to write a function

Part 1: Functions

- Information about the world or the objects in the world are all stored in properties
- A function is used to ask questions about these properties
- For example, some of the built in functions of an object are: color, height and depth.
- A function is not a behavior or an action. It simply returns the information that we need.

Function

- In this world, we want to know: is the object that was clicked on one of the objects in the junkyard?
- If it is one of the objects in the junkyard, we want the fire to go to it.
- If it isn’t, we don’t want anything to happen

Types of Functions

- Depending on the question, a function can return any type of value.
- Some of the options of a return type are:
  - Number
  - Boolean (true or false)
  - Object
  - String
- But there are more types

Part 2: Creating a function

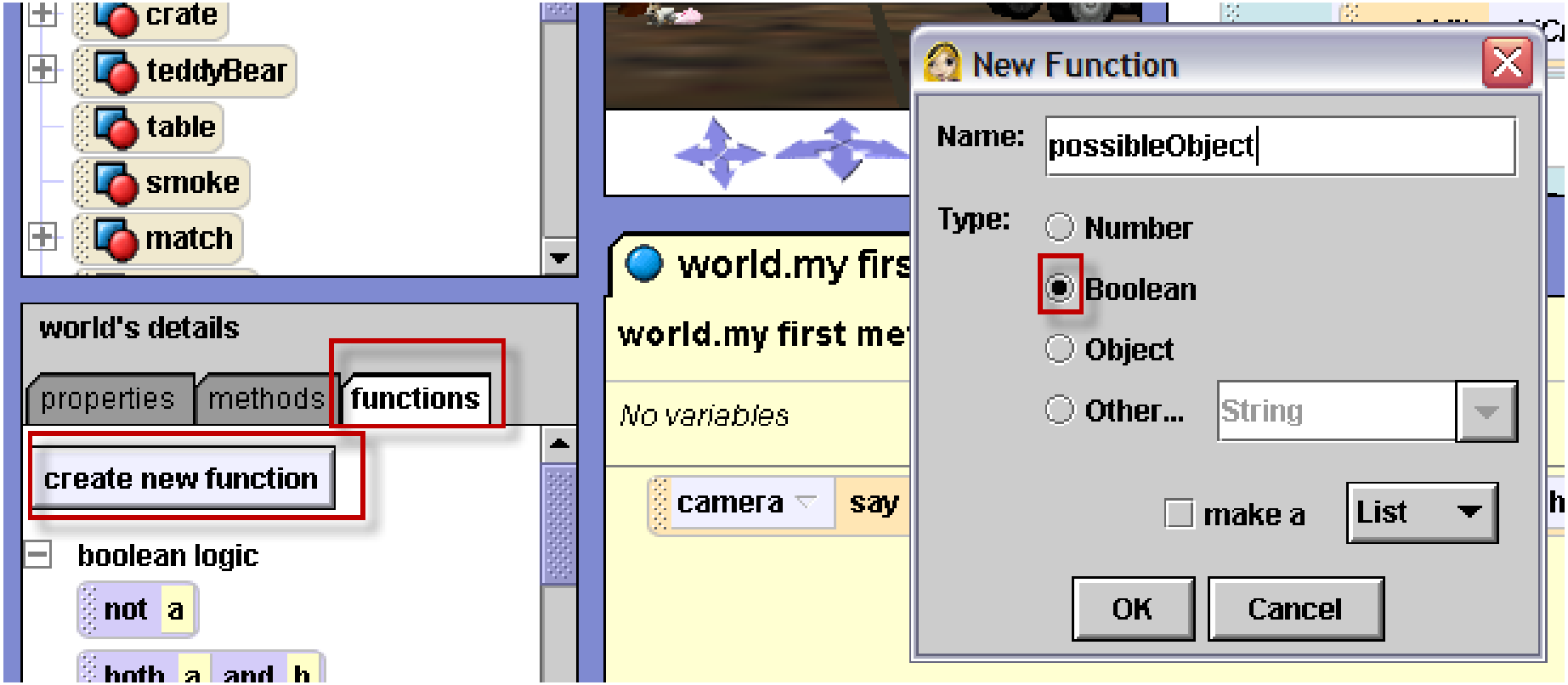
Step 1: What type of function to create

- We want our function to be type boolean. Boolean returns either true or false.
- This function will return true if the object clicked on is in the junkyard. It will return false otherwise

Step 2: How to create a function

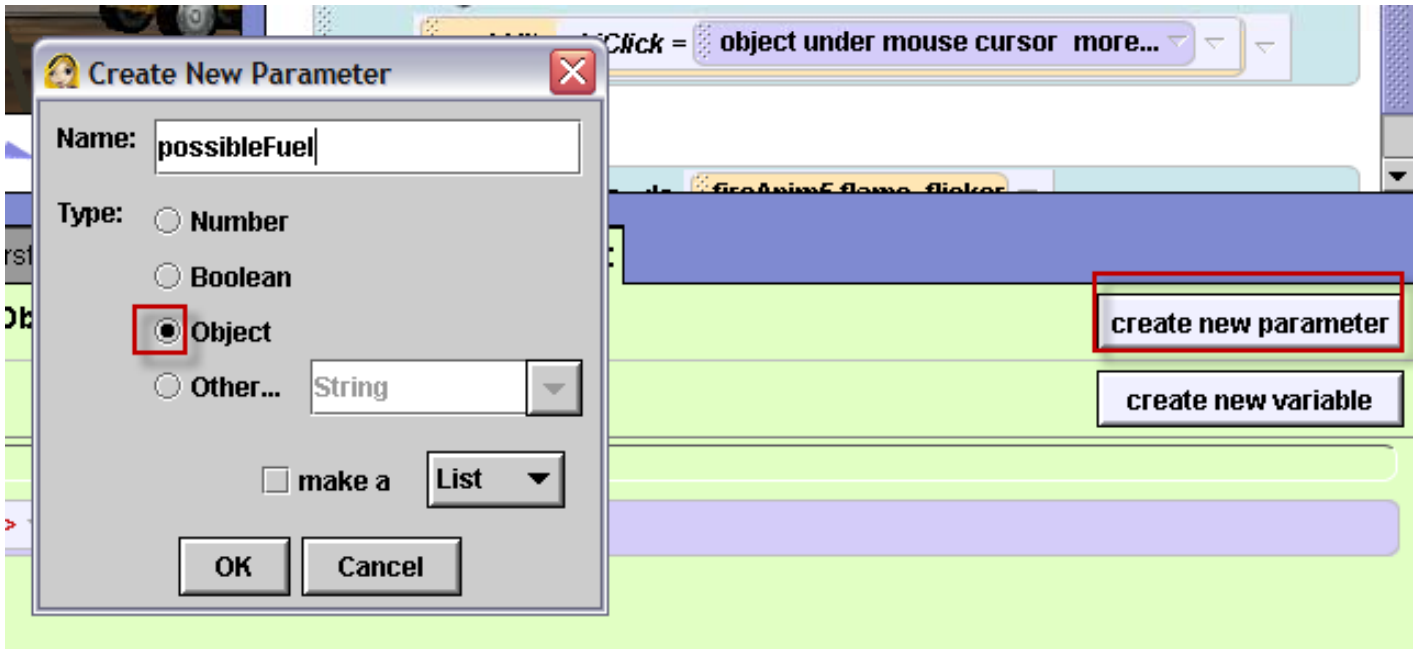
- Click on the function tab in the world detail's pane.
- Click on create new function
- Name it possibleObject and make sure it is type object
- See the screenshot on the next slide for an illustration

Creating a function



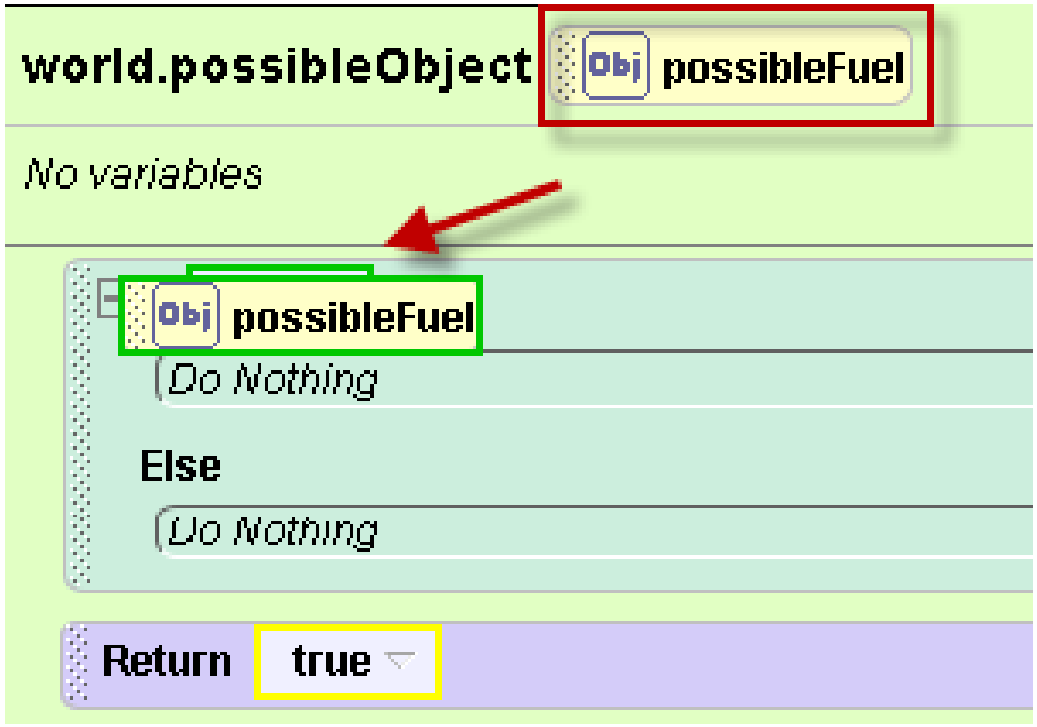
Part 3: Writing the function: Step 1: the parameter

- The object that was clicked on is going to be used in this function. In the function, create new parameter
- Name it possibleFuel and it is type object



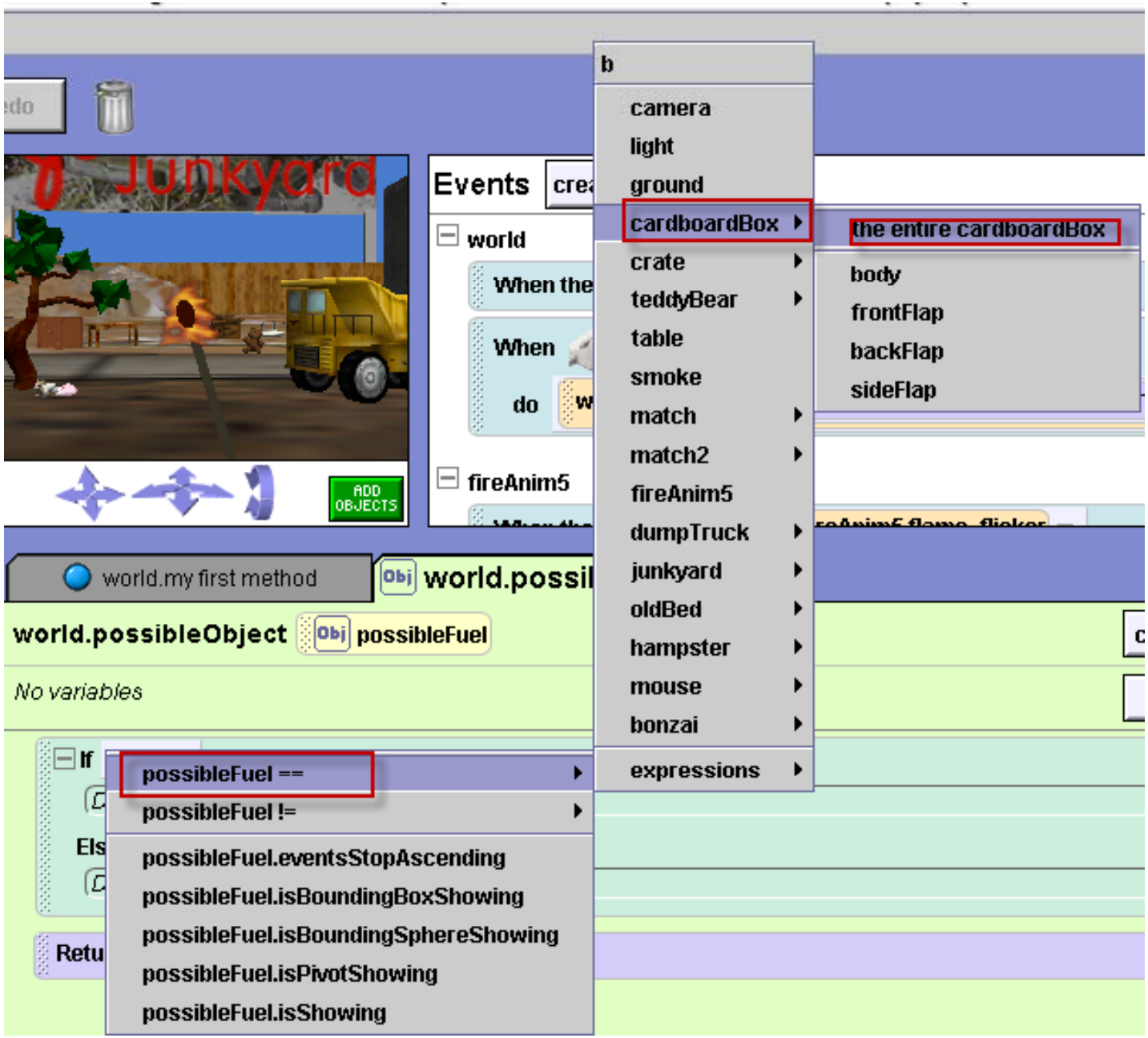
Step 2: Writing the function: if/else

- Drag an if/else into your function and select true in the drop down menu
- Drag the parameter possibleFuel on top of the true



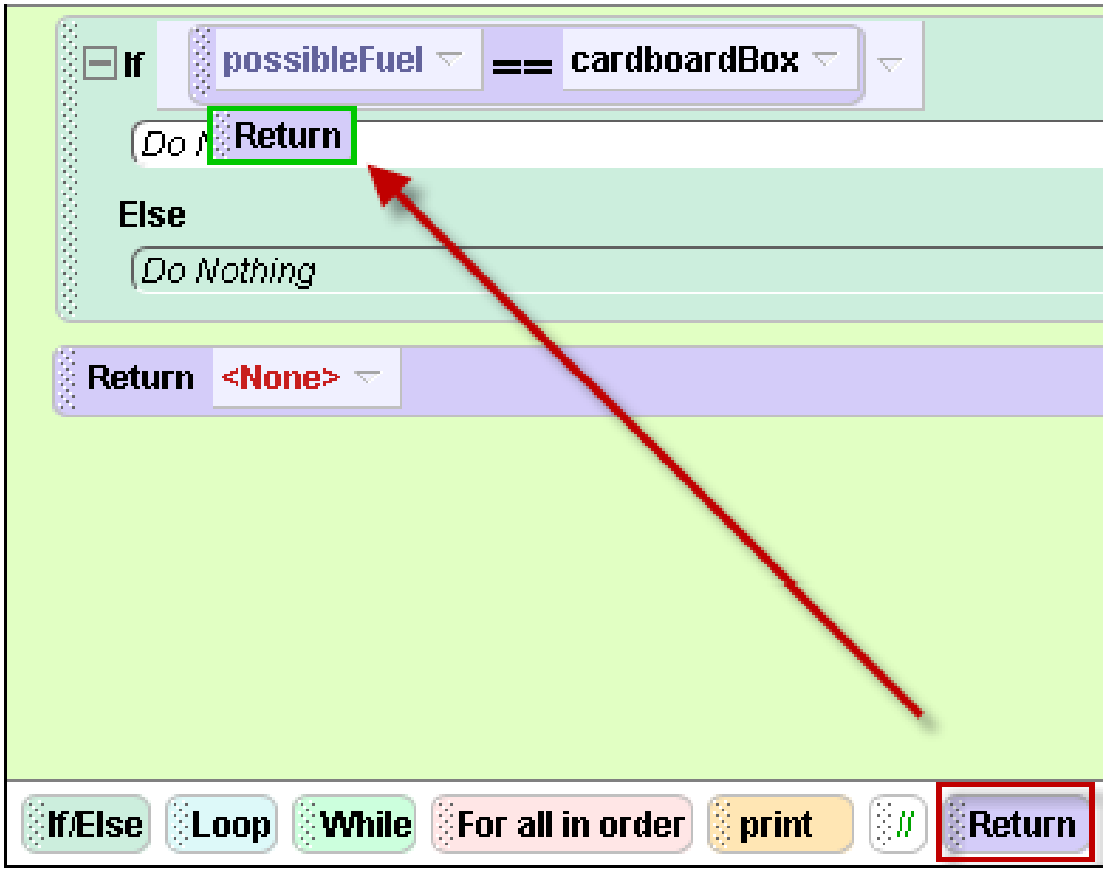
Step 3: The conditional

- When you release the parameter, select possibleFuel ==;
- then select cardboardBox,
- the entire cardboardBox
- See the screenshot on the next slide for an illustration


Junkyard
ADD OBJECTS
Events
world
When the
When
do
fireAnim5
world.my first method
world.possil
world.possibleObject
possibleFuel
No variables
If
possibleFuel ==
possibleFuel !=
possibleFuel.eventsStopAscending
possibleFuel.isBoundingBoxShowing
possibleFuel.isBoundingSphereShowing
possibleFuel.isPivotShowing
possibleFuel.isShowing
b
camera
light
ground
cardboardBox
crate
teddyBear
table
smoke
match
match2
fireAnim5
dumpTruck
junkyard
oldBed
hampster
mouse
bonzai
expressions
the entire cardboardBox
body
frontFlap
backFlap
sideFlap

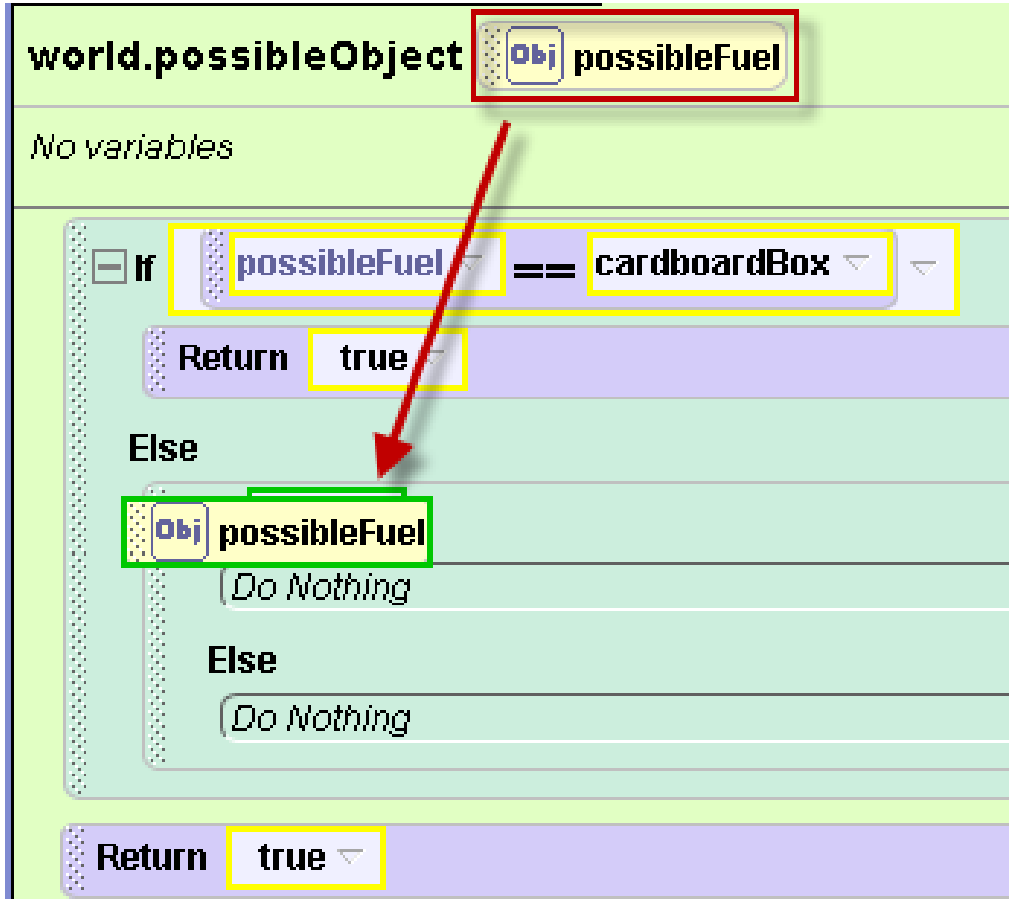
Step 4: Return

- Drag return on top of the Do Nothing
- Select true



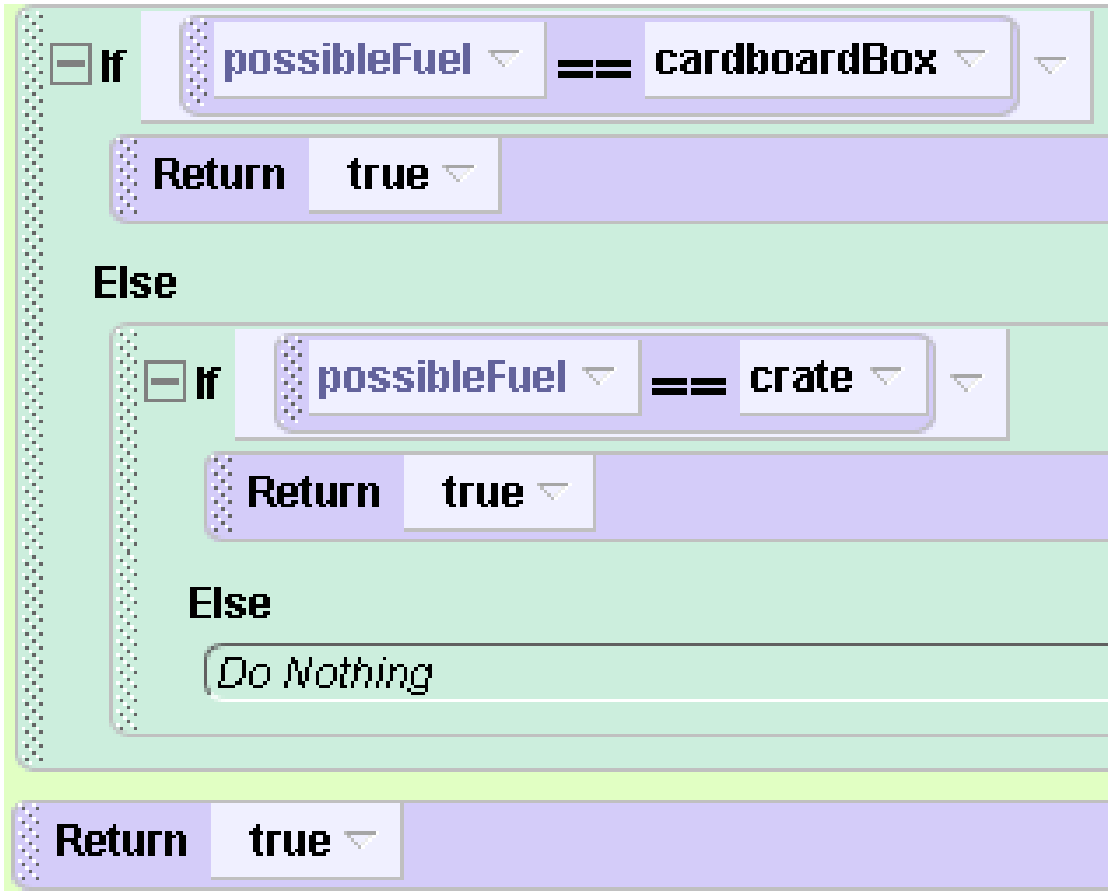
Step 5: Nested if statements

- Drag if/else on top of the Do Nothing underneath Else
- Then drag possibleFuel on top of the true
- When you release it, select possibleFuel==;
- Select crate
- Select the entire crate



Nested if/else statements

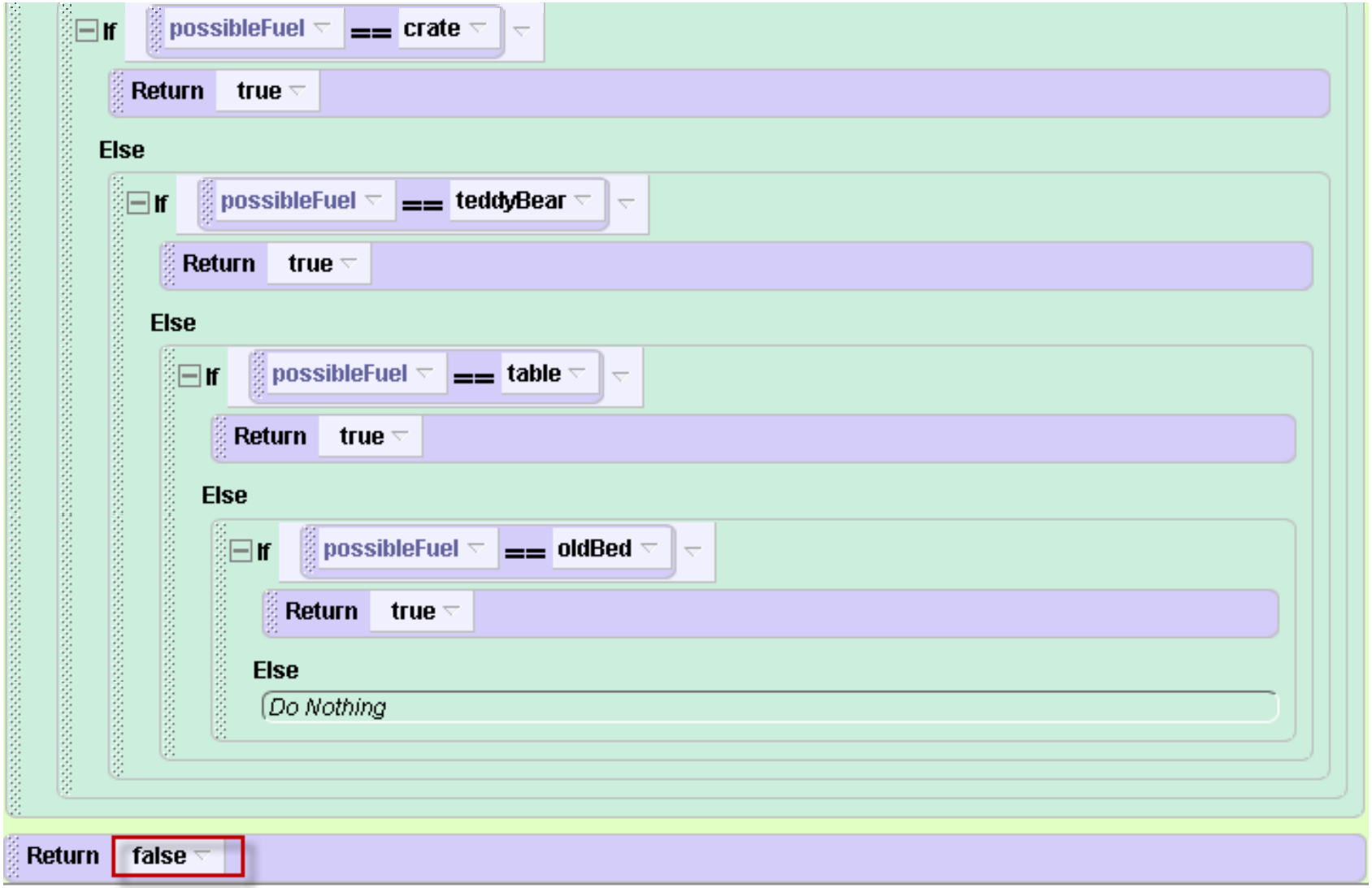
- Then drag return true onto the Do Nothing. Your function should look like this so far:



Step 6: Finish the function

- Drag a new if/else onto the Do Nothing underneath Else
- Repeat the steps for these objects:
  - the teddyBear
  - the table
  - The oldBed
- Then we want to return false for any object not in the junkyard
- See the screenshot on the next slide for the rest of the function

The Rest of the function



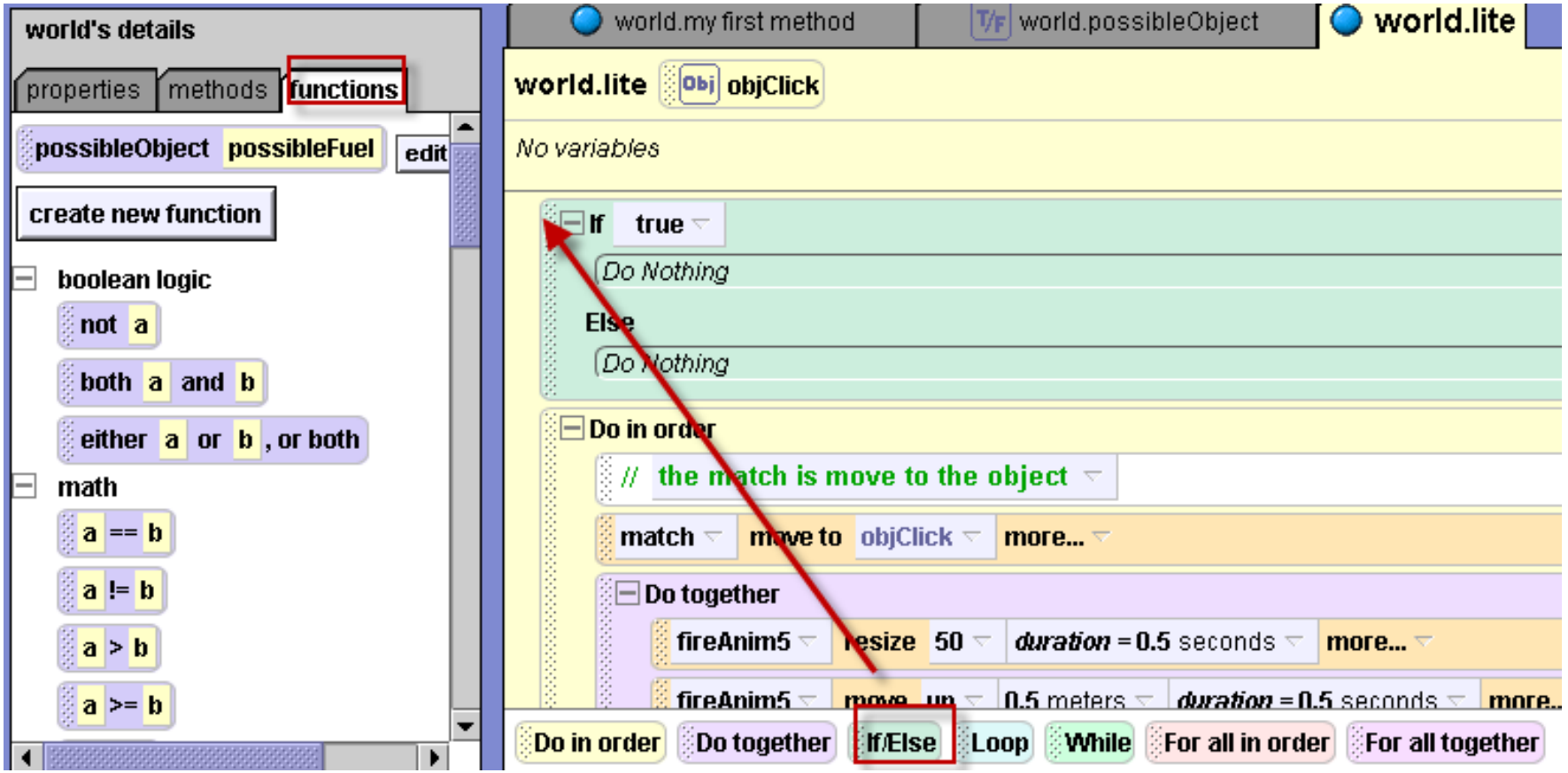
- Don’t forget to set the final Return to false

Part 4: Calling the function

- Click on the methods tab in the world details pane
- Click edit beside the method lite so that you can see the code
- Click on the functions tab in the world details pane

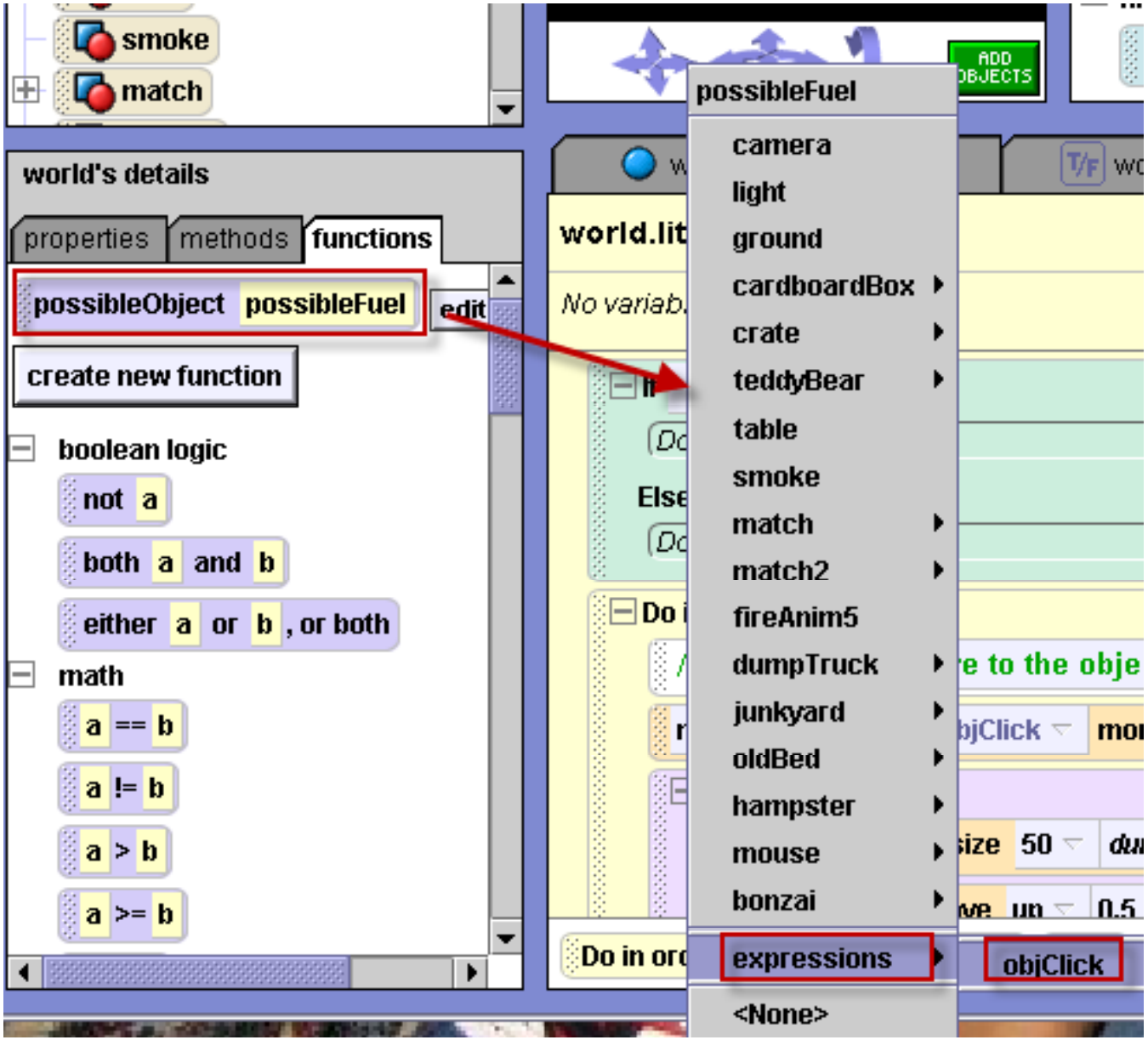
Calling the function (cont 1)

- Drag if/else to the top of the lite method



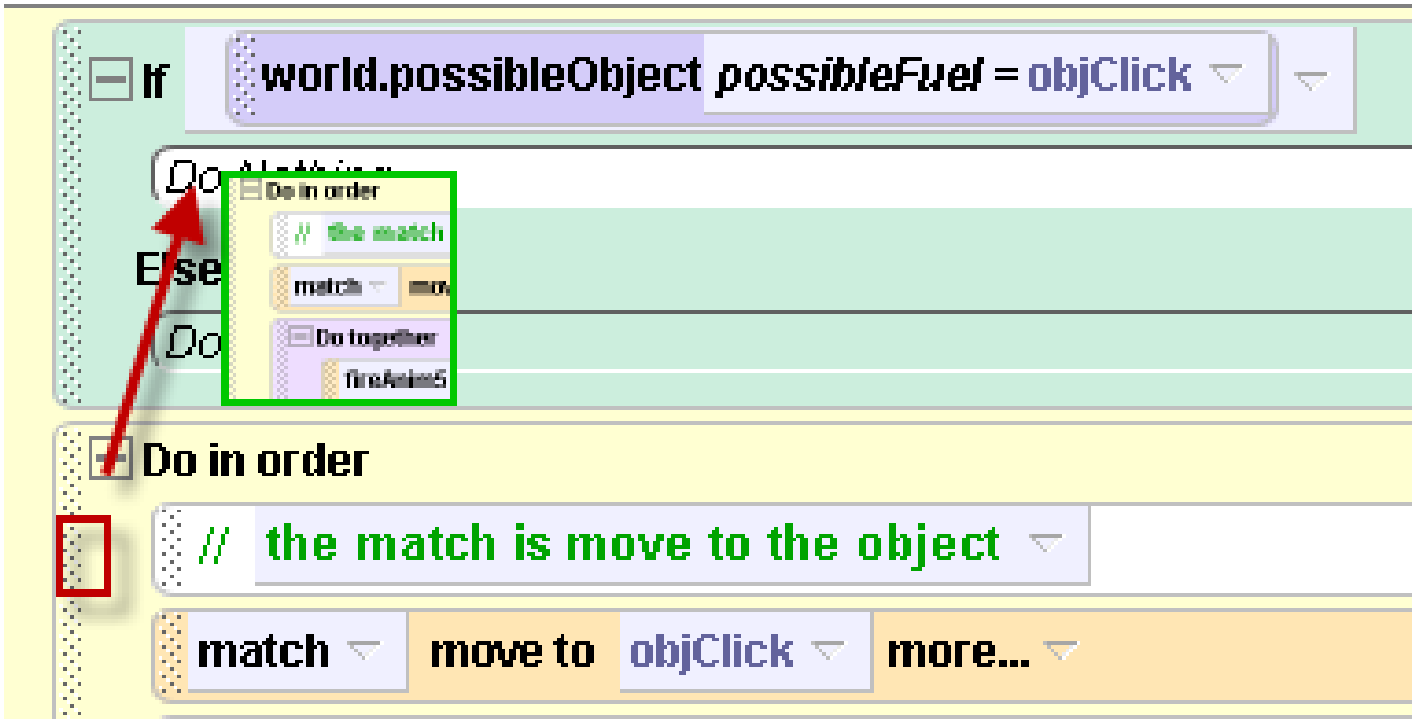
Calling the function (cont 2)

- Now drag the possibleObject from the function tab on top of the true
- Select expressions
- Select objClick
- See the screenshot on the next slide for an illustration


smoke
match
world's details
properties
methods
functions
possibleObject
possibleFuel
edit
create new function
boolean logic
not a
both a and b
either a or b , or both
math
a == b
a != b
a > b
a >= b
possibleFuel
camera
light
ground
cardboardBox
crate
teddyBear
table
smoke
match
match2
fireAnim5
dumpTruck
junkyard
oldBed
hampster
mouse
bonzai
expressions
<None>
objClick
ADD OBJECTS
world.lit
No variab
Else
Do in orc

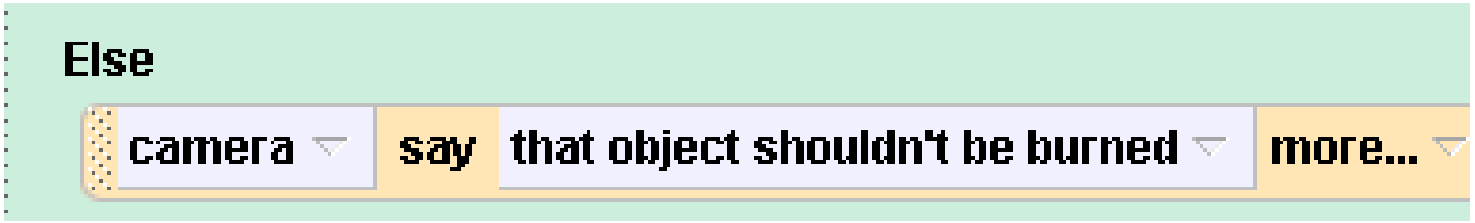
The if

- Click on the already written Do in Order
- Drag it into the if/else statement on top of the Do Nothing
- Play your world



The else

- Scroll down to the Else and drag the camera say method and type in the string “that object shouldn’t be burned”:

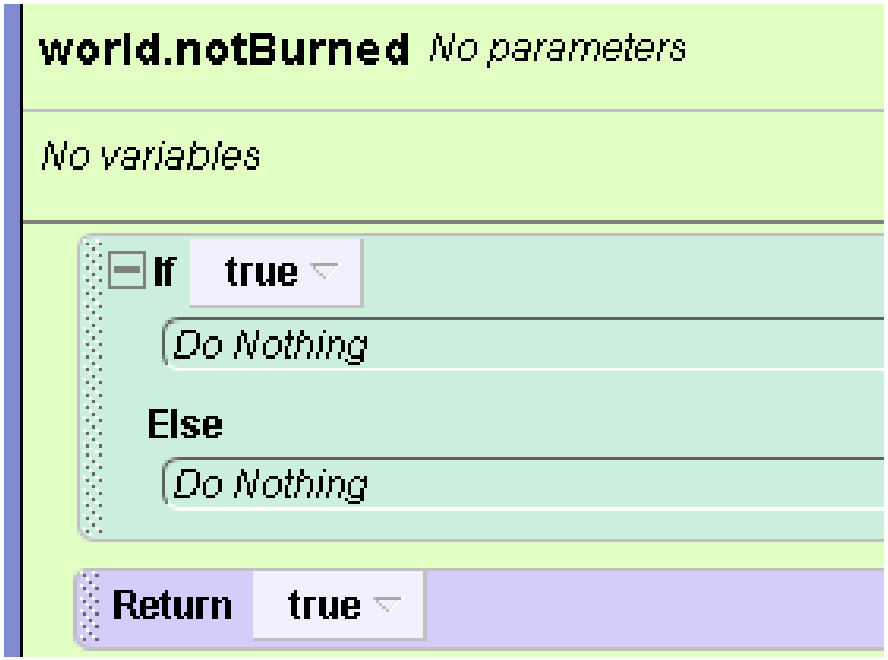


Part 5: One more error

- Now when you play your world, only the objects that are in the junkyard are burned.
- However, when you burn one of the objects, you can still burn it again
- Let's nest another if statement in the lite method to check whether or not the object has been burned
  - "Burned" means its' color has been changed to black

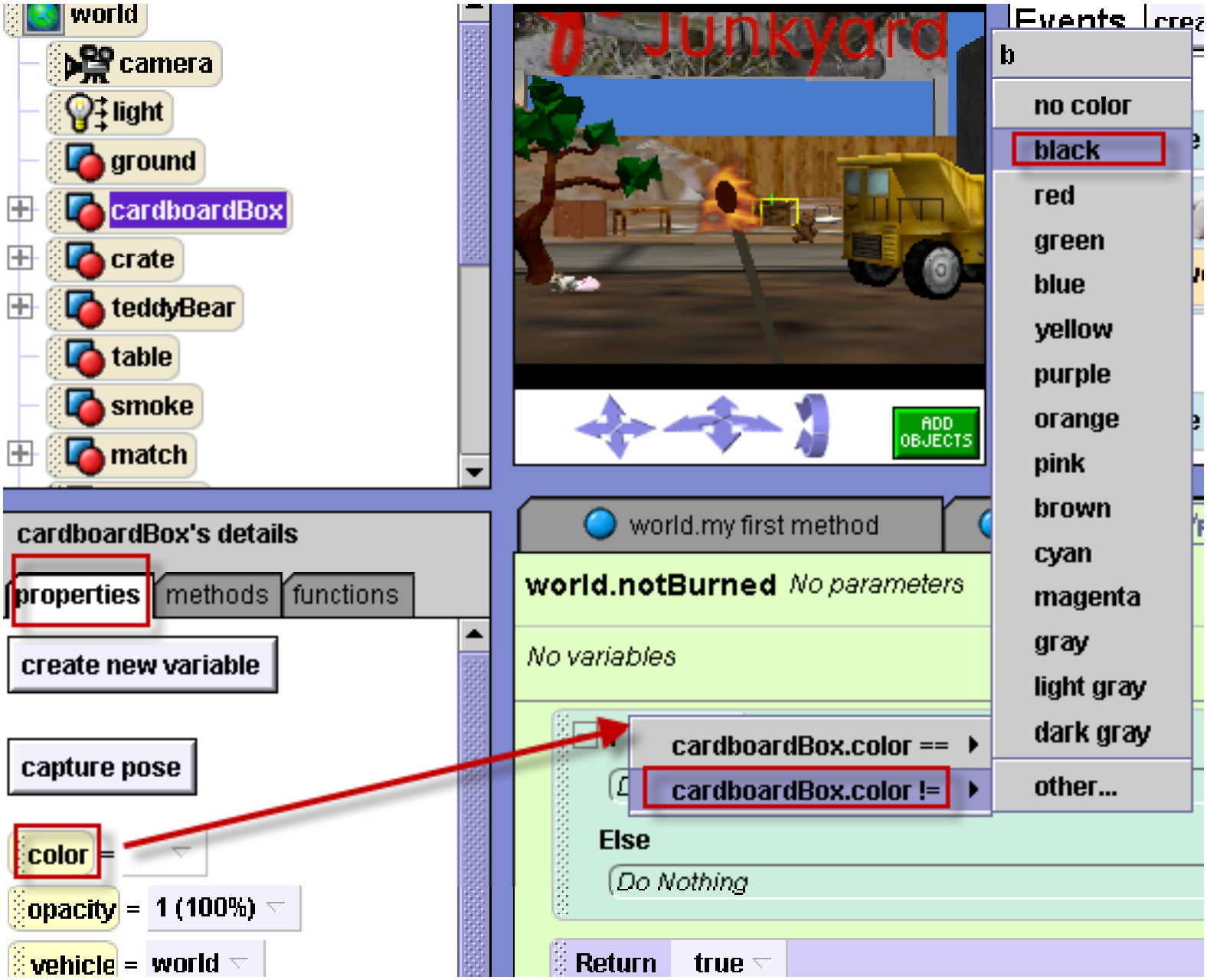
Step 1: Writing a new function: is it burned?

- Now, click on the function tab in the world details pane. Create a new function named notBurned. It should also be type Boolean
- Drag an if/else into the function



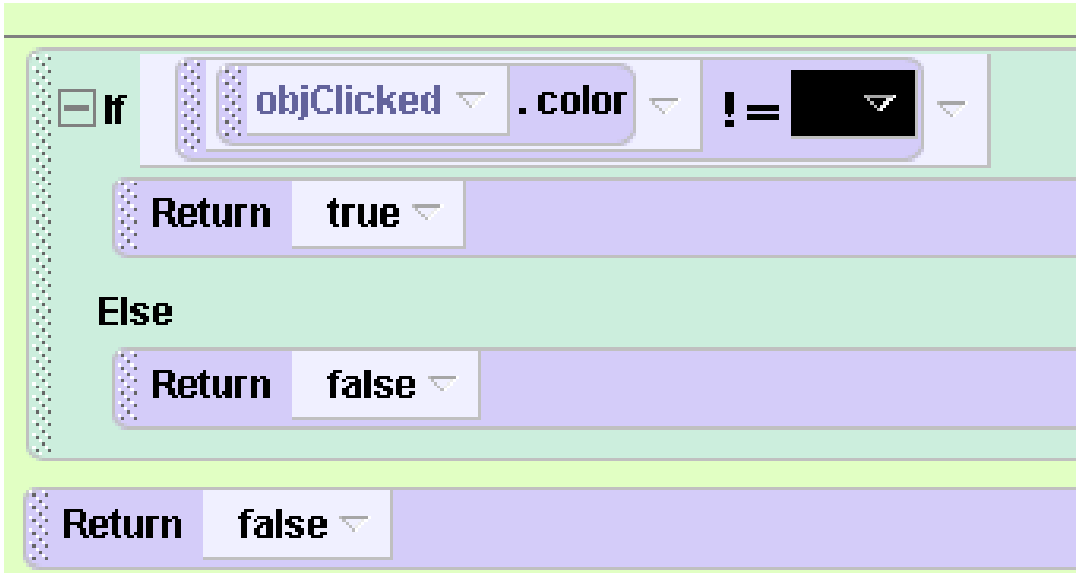
Is it burned?

- In this function, we want to return true if the object has not been burned
- We know an object has been burned if it's black
- Click on cardboardBox in the object tree
- Click on it's properties tab
- Drag color on top of the true in this function
- Select != black.
- See the screenshot on the next slide for an illustration


world
camera
light
ground
cardboardBox
crate
teddyBear
table
smoke
match
Junkyard
ADD OBJECTS
b
no color
black
red
green
blue
yellow
purple
orange
pink
brown
cyan
magenta
gray
light gray
dark gray
other...
cardboardBox's details
properties
methods
functions
create new variable
capture pose
color =
opacity = 1 (100%)
vehicle = world
world.my first method
world.notBurned No parameters
No variables
cardboardBox.color ==
cardboardBox.color !=
Else
Do Nothing
Return true

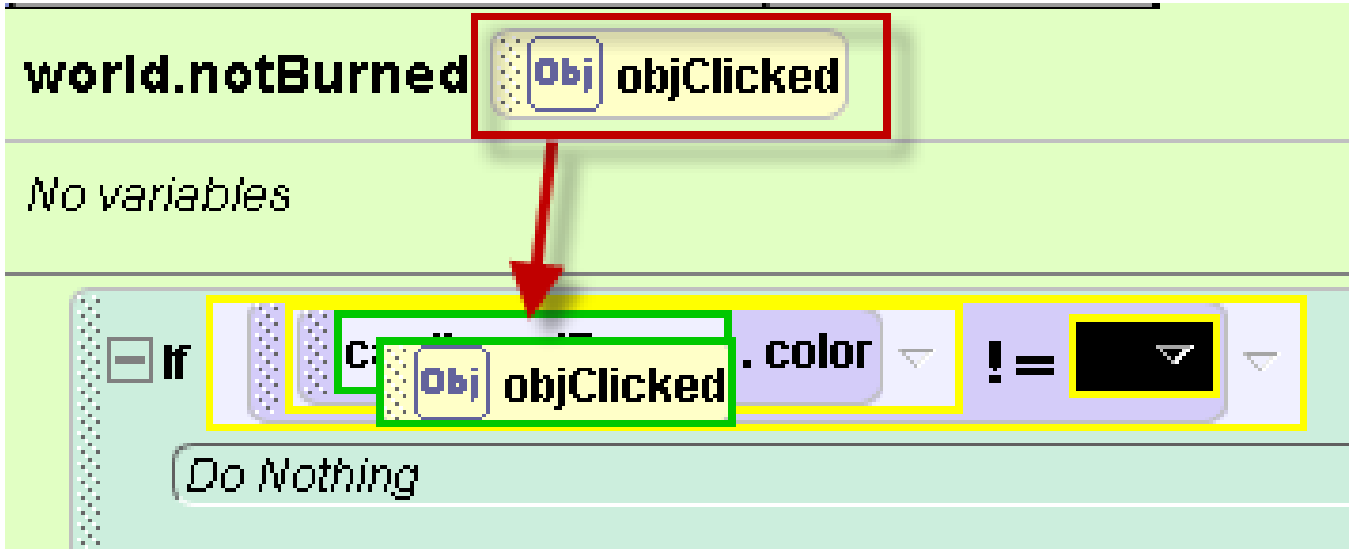
Step 2: Return statements

- Now drag return on top of the first Do Nothing. Select true
- Drag return on top of the second Do Nothing. Select false. Select false for the last return
- Complete method:



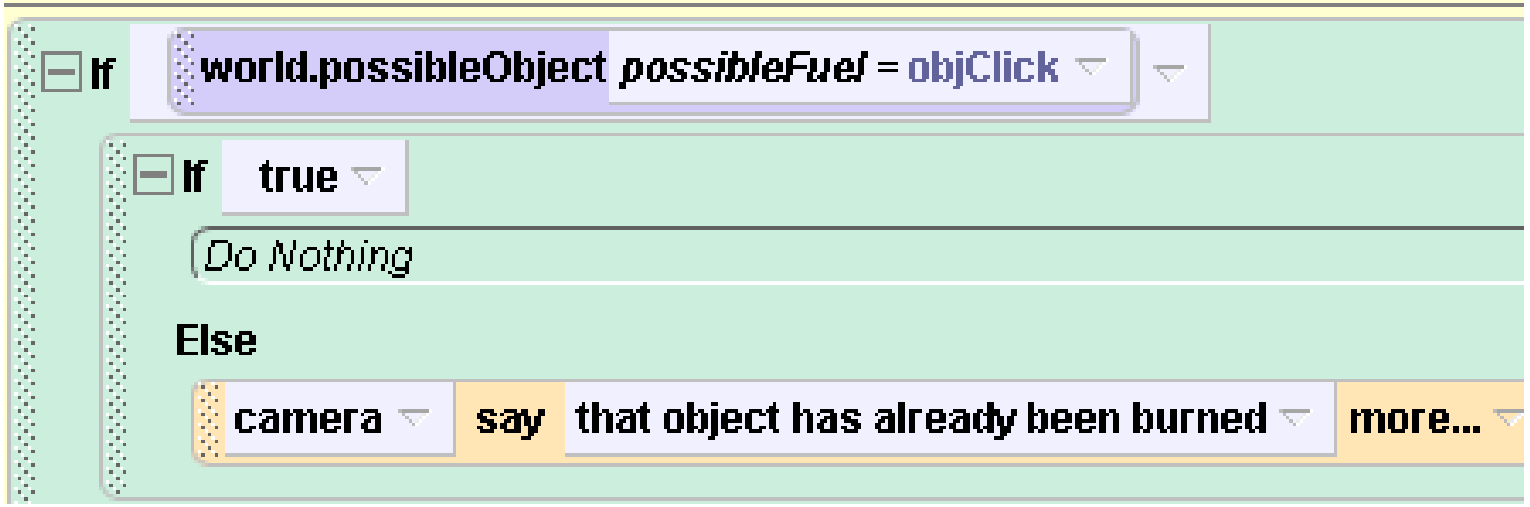
Step 3: Create new parameter

- Create a new parameter in the function
- Name it objClicked and it is type object
- Drag objClicked on top of cardboardBox in the function's if statement



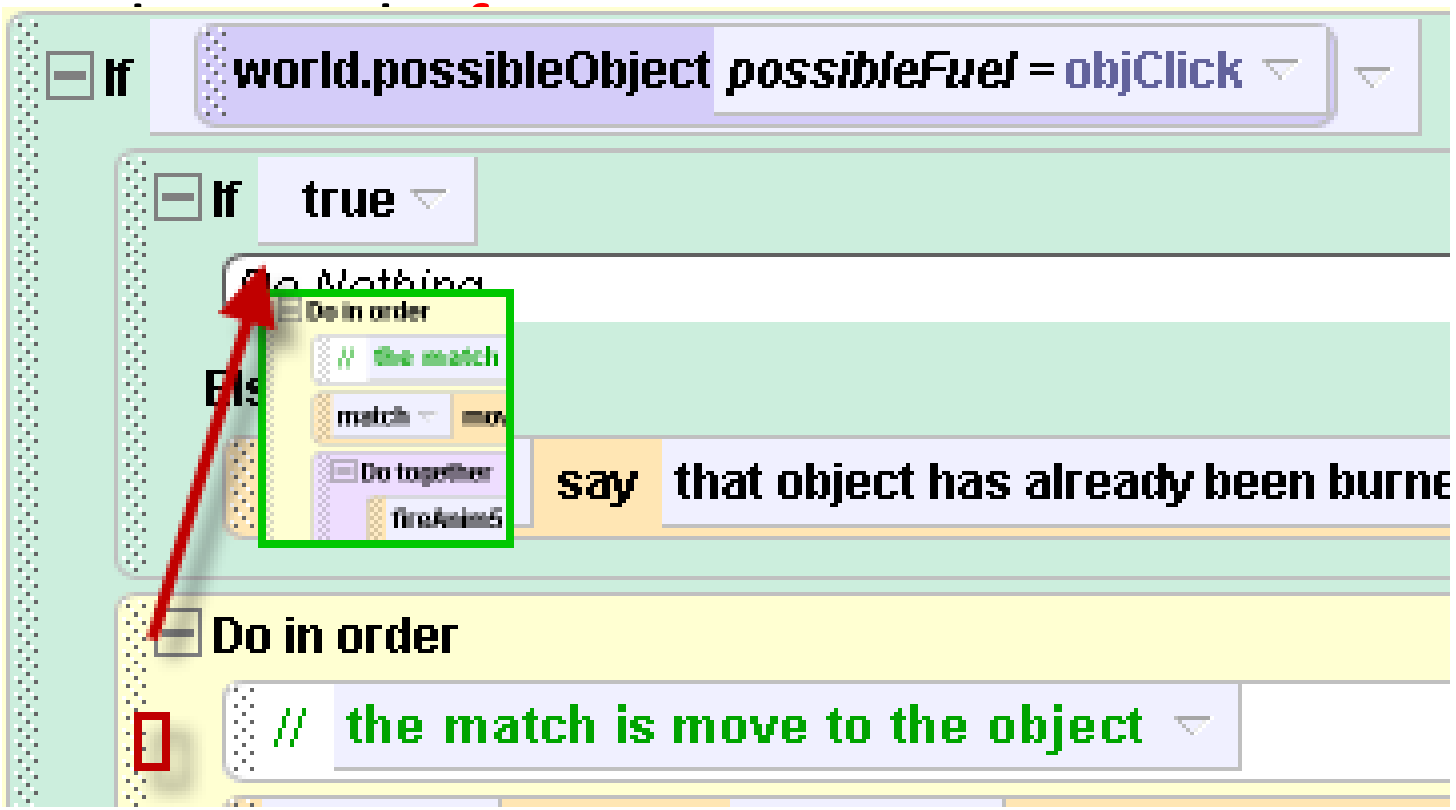
Step 4: Nest an if/else

- Click on the world.lite tab to see the code
- Drag in a new if statement
- For the Else of that statement, drag in the camera say method and type in: “that object has already been burned”
- So far, your method should look like this:



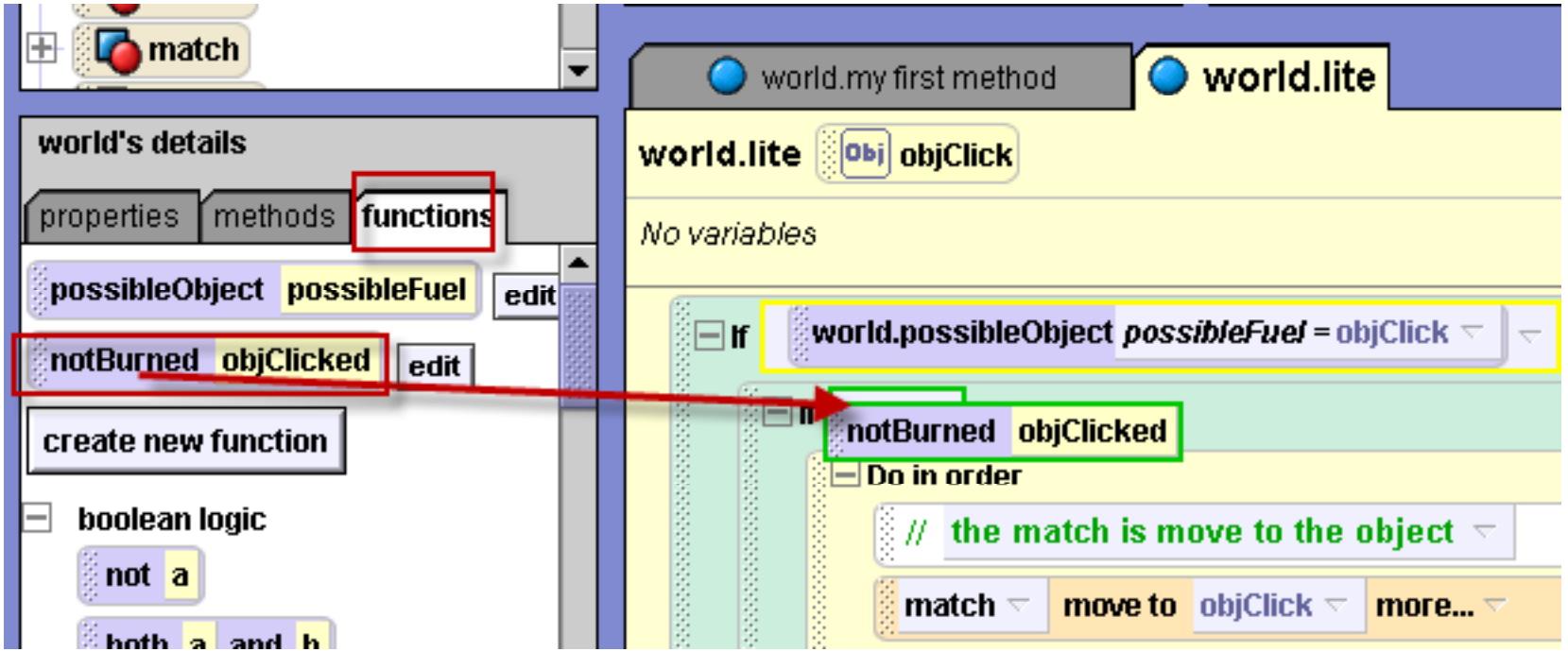
Nested if/else (cont)

- Now click on the already written Do in order and drag it on top of the Do Nothing



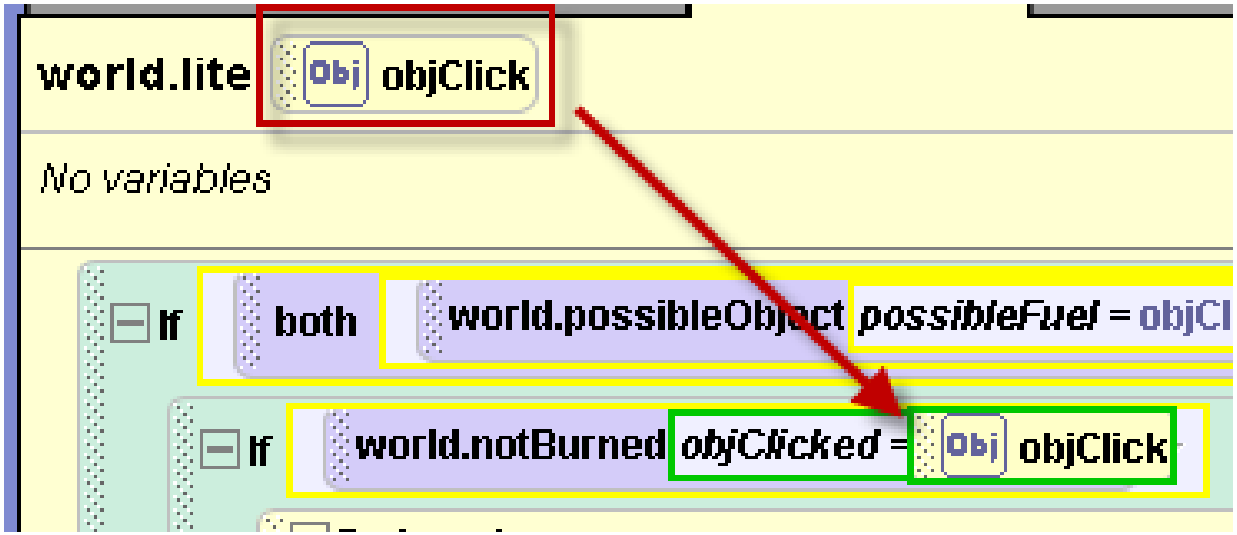
Part 6: Calling the function notBurned

- Click on world in the object tree. Click on the functions tab. Drag the function notBurned on top of the true in your if statement



Calling the function notBurned (cont 1)

- Then drag the parameter objClick onto the <None>



- Play your world.

Recap

- A function is not a behavior or action
- It is used to return information about the state of the world or the characters in the world
- There are many different types of functions. For example, functions can return objects, numbers, booleans, etc
- Write your own functions based on questions you need to answer about your world