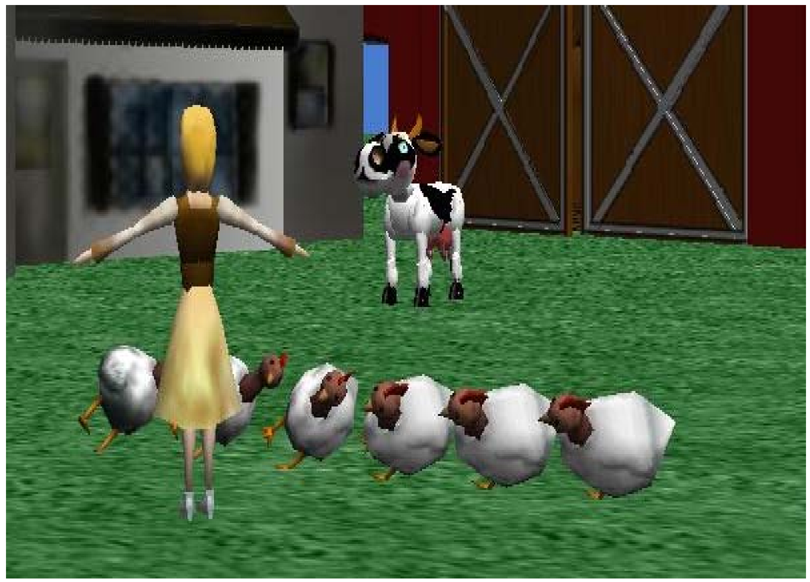
Tutorial for Arrays and Lists



By Ruthie Tucker

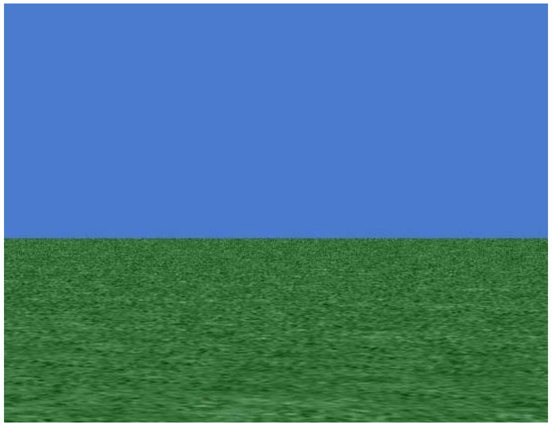
Under the direction of Professor Susan Rodger, Duke University 2008

www.cs.duke.edu/csed/alice/aliceInSchools

Description

- This presentation will cover the basics of using Arrays and Lists in an Alice world
- It uses a set of chickens on a farm
- Prerequisites
  - Everything

The Characters

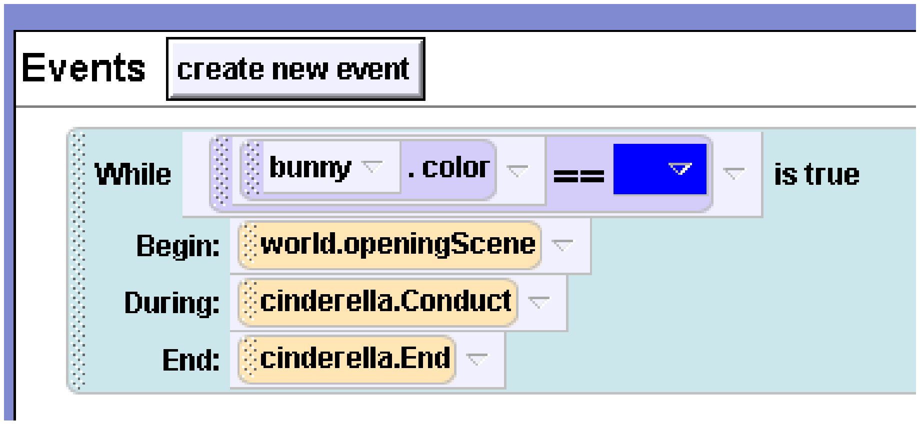





Setting up the background

- three methods have already been given to you.
- World.OpeningScene
- Cinderella.Conduct
- Cinderella.end
- These methods will allow Cinderella to conduct the chickens while they dance.

Setting up the Background

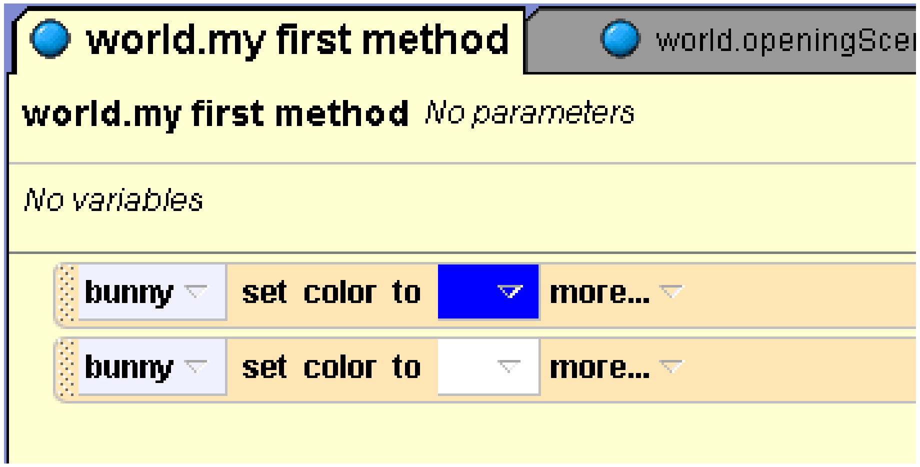


- This is how your event should look

- We are going to execute these methods in a BDE event in our events editor.
- We will put an invisible bunny in the background that changes colors to execute the rest of the BDE.

Setting up the Background

- This is how your world.my first method should look right now.

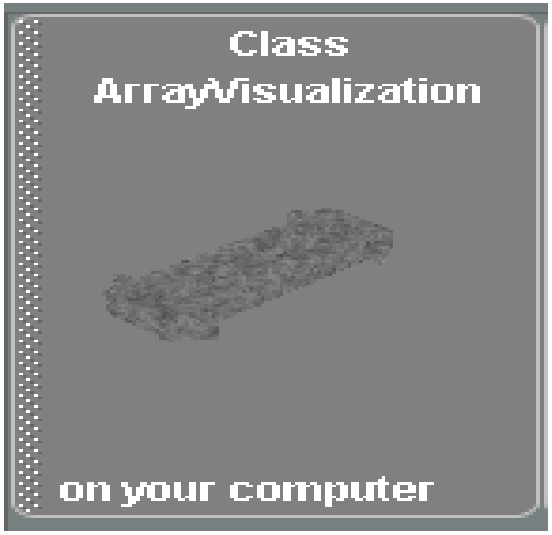


Animating the chickens

- Now we are going to animate our chickens
- To do this we will need to use two things
- A list
- And an Array

Getting Started

- What is a List?
  - A list is simply a way to organize information.
- What is an Array?
  - An Array is a structure for collecting and organizing objects or information of the same type into a group.
  - In Alice Arrays and classes are visual objects to be dropped into your world

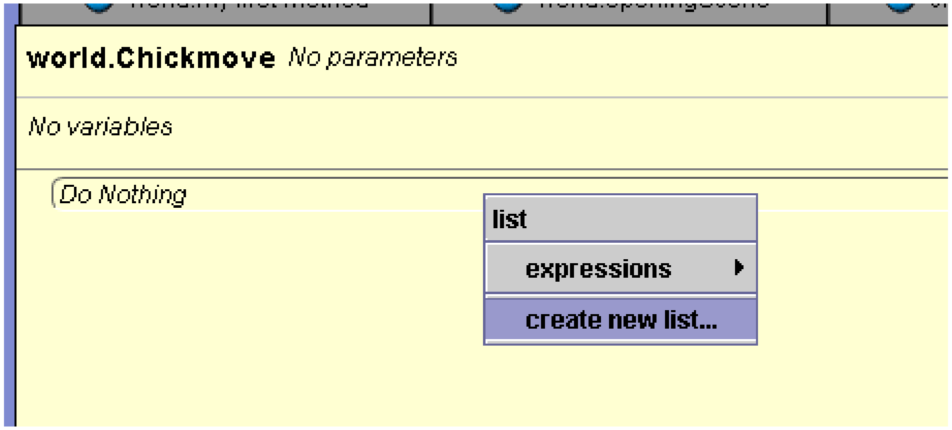


Side Note

- Glitch in Alice
  - Alice cannot compute your world if you put the same objects in a “List Visualization” as an “Array Visualization”.
  - I will demonstrate how to use the “List Visualization” later, but for now we are just going to make a regular list.
  - However, you can use a list without the actual Visualization, if you also want to use an “Array” in your world

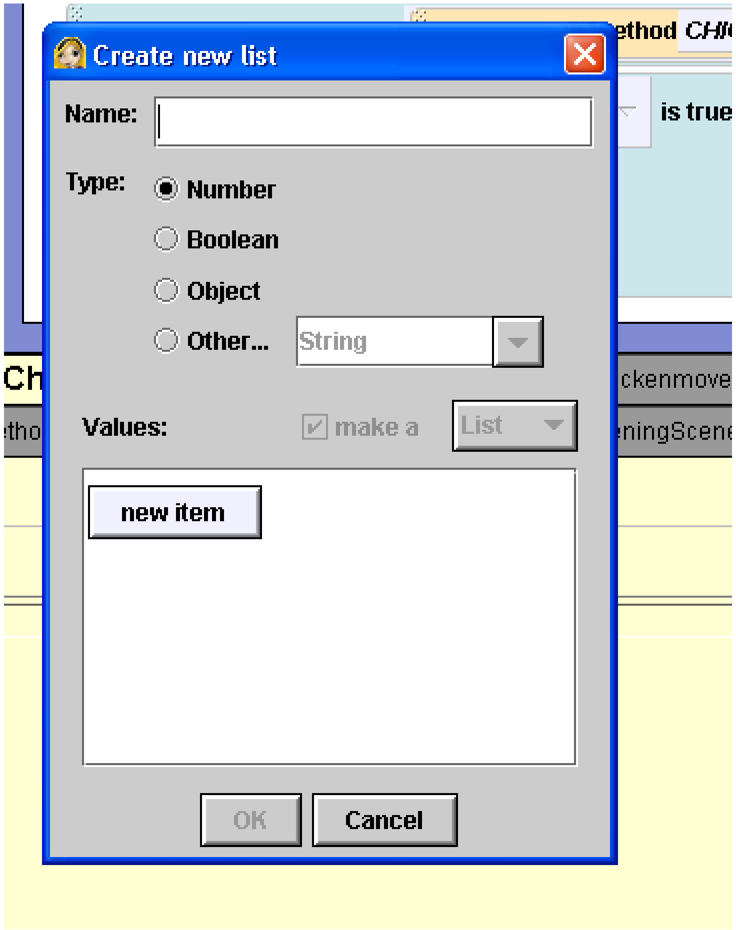
Making a List

- Create a world level method named "ChickMove."
- Drag the "For all Together" Button onto the method editor.
- Select "Create new list."



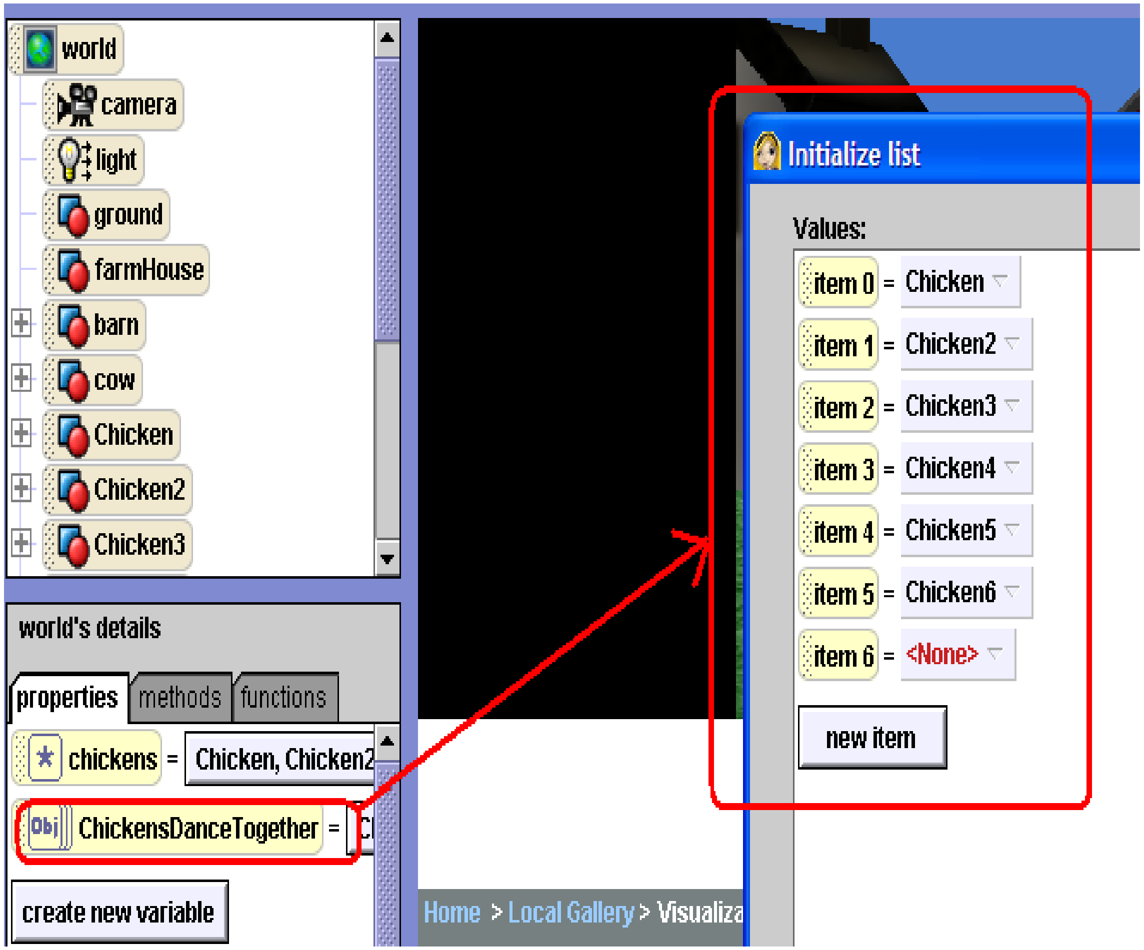
Making a List

- A box will pop up for you to enter in the items on your list.
- Name it “ChickensDance Together.”
- Enter in each chicken, keeping in mind that the entries start at slot ‘0’.



Lists

This is how your box should look.

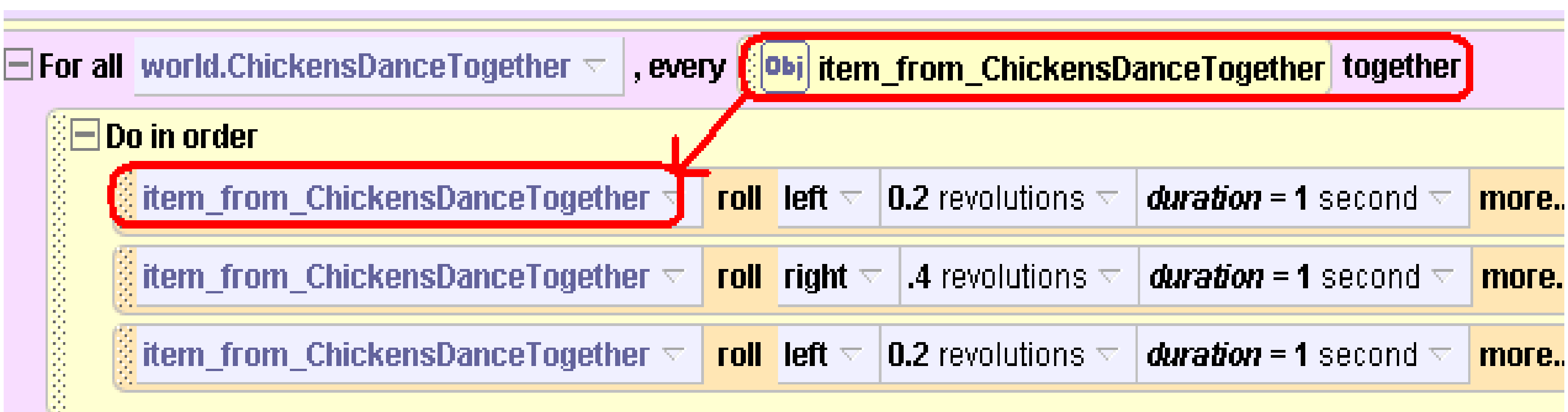


Lists

- We want the chickens to roll left and right together.
- Above is the code to have one chicken roll left and then right
- I selected Chicken 2 but any chicken will work
- Put all of this code in your World.Chickmove method

Lists

- Now drag the tab labeled “item_from_ChickensDanceTogether” over the box that was previously labeled “Chicken 2.”
- Repeat this for each command to make the chicken roll



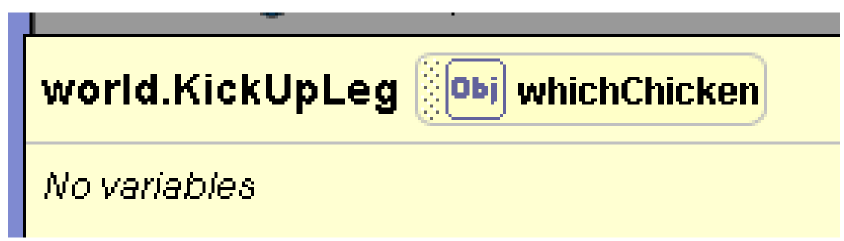
Lists

- Now we have programmed the List to have all the chickens roll left and then right at the same time.
- But what if we want them to do something, one at a time?
- This is where ‘For all in order’ comes in.
- We will now program the chickens to kick up their left leg, one at a time.

Lists

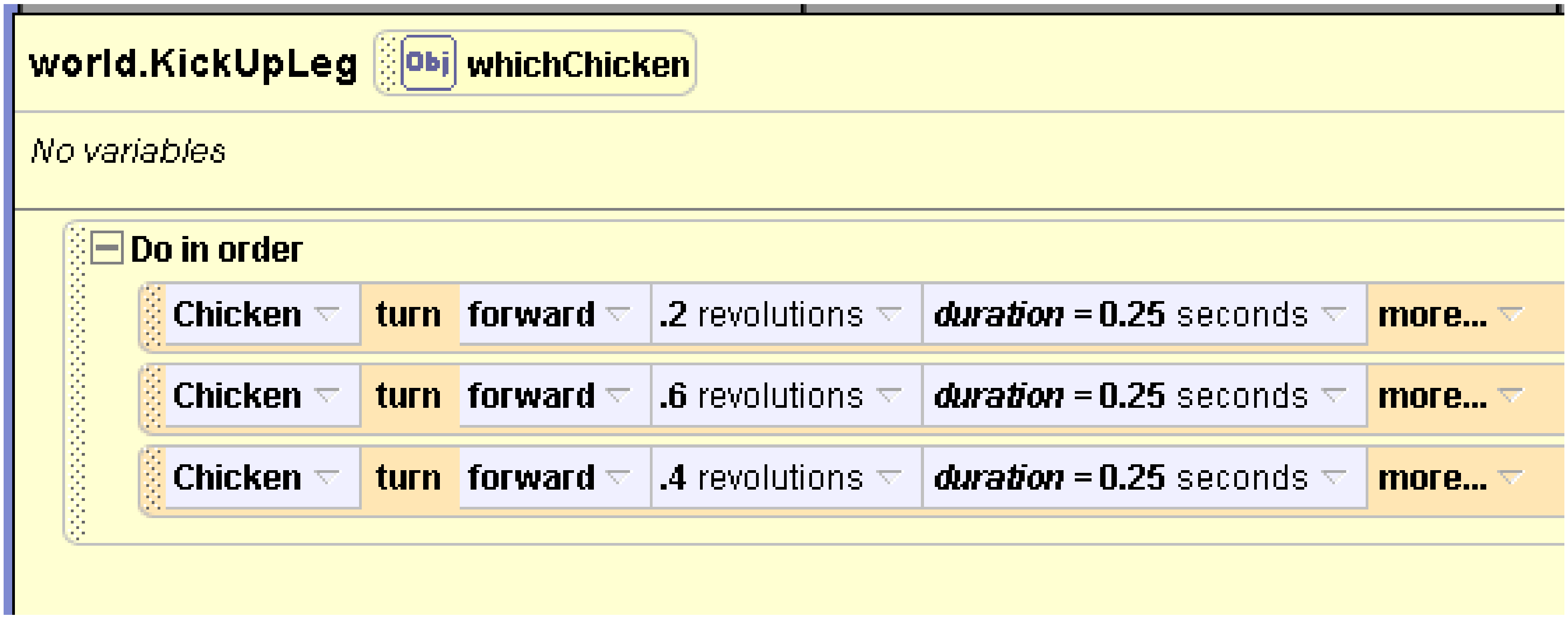
- Open a New method named “KickUpLeg” and create a new parameter called “WhichChicken.”
- Click on the “For all in order” tab at the bottom of your screen and drag it into your method

- Side Note: Always make sure it is an “object” parameter

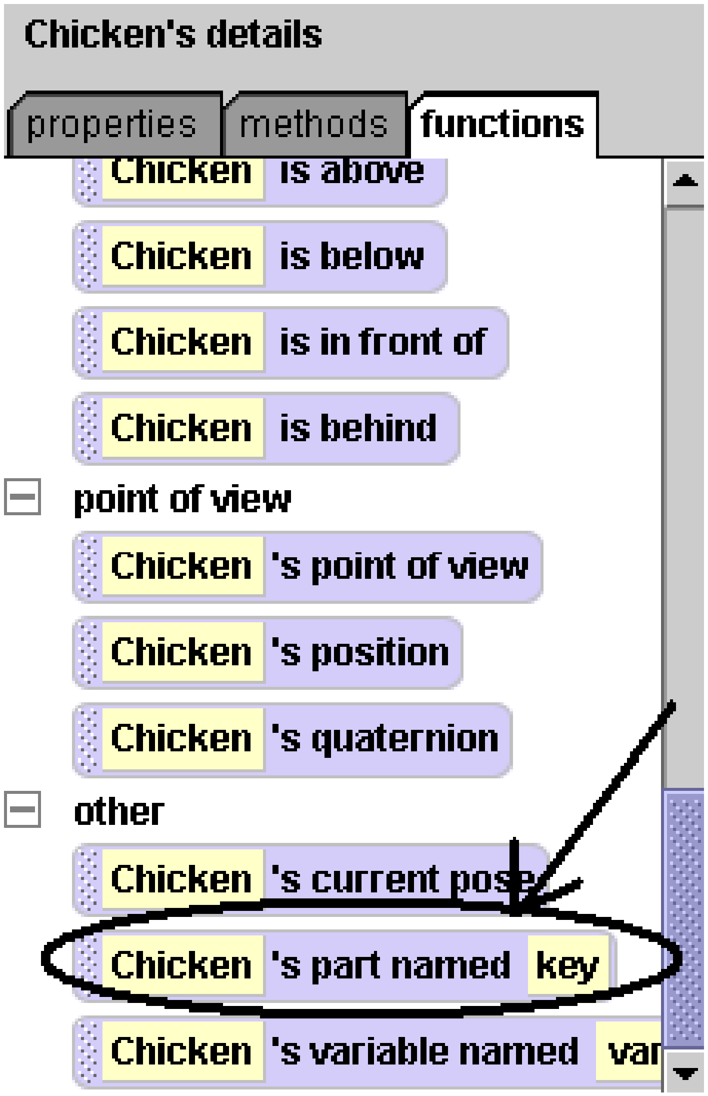


Lists

- First write the method for the chicken's leg to kick, using the entire chicken as the place holder.



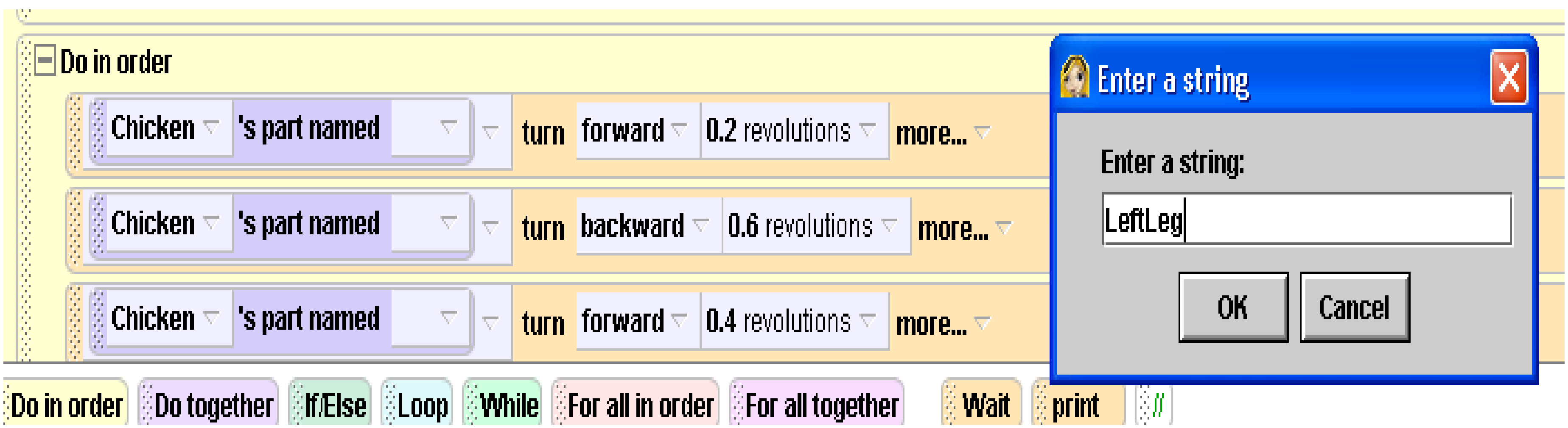
Part named Key



- Click on world level functions and select a function labeled "Parts named key."
- Replace each "chicken" with this function in your method.

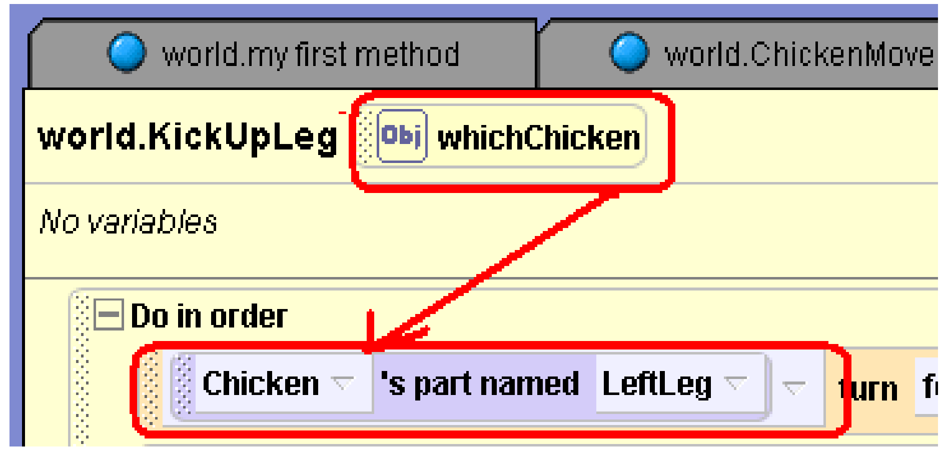
Part Named Key

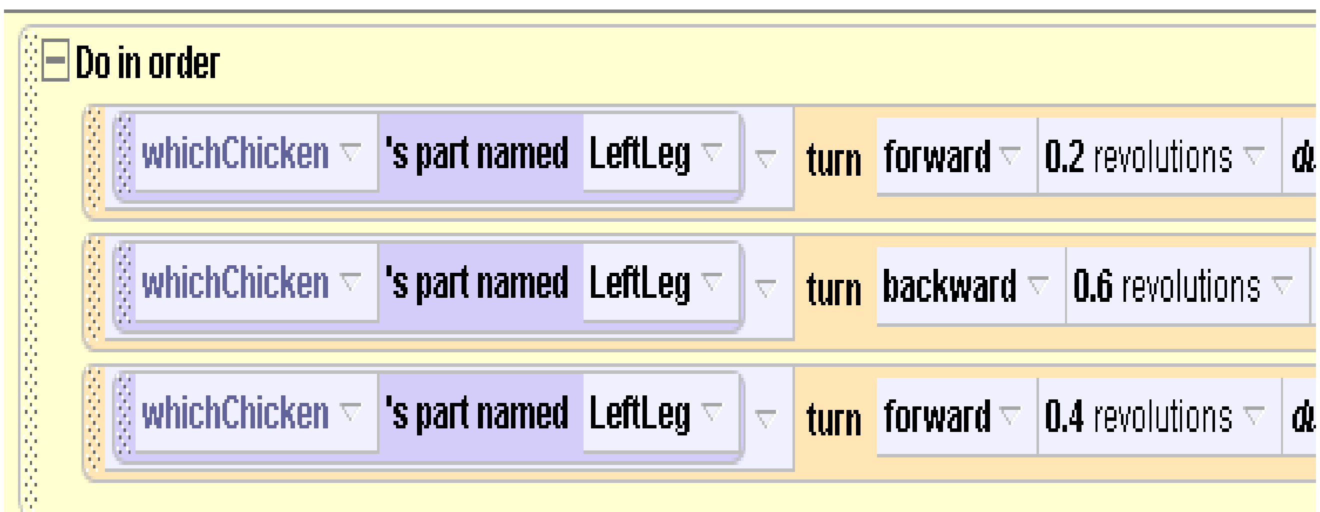
- Then, drag the parameter ‘whichChicken’on top of ‘Chicken’.
- For each white box select the arrow down key and hit “other”
- A box will appear for you to type.
- You must get the syntax exactly correct.
- Type in “LeftLeg”



Part named key conclusion

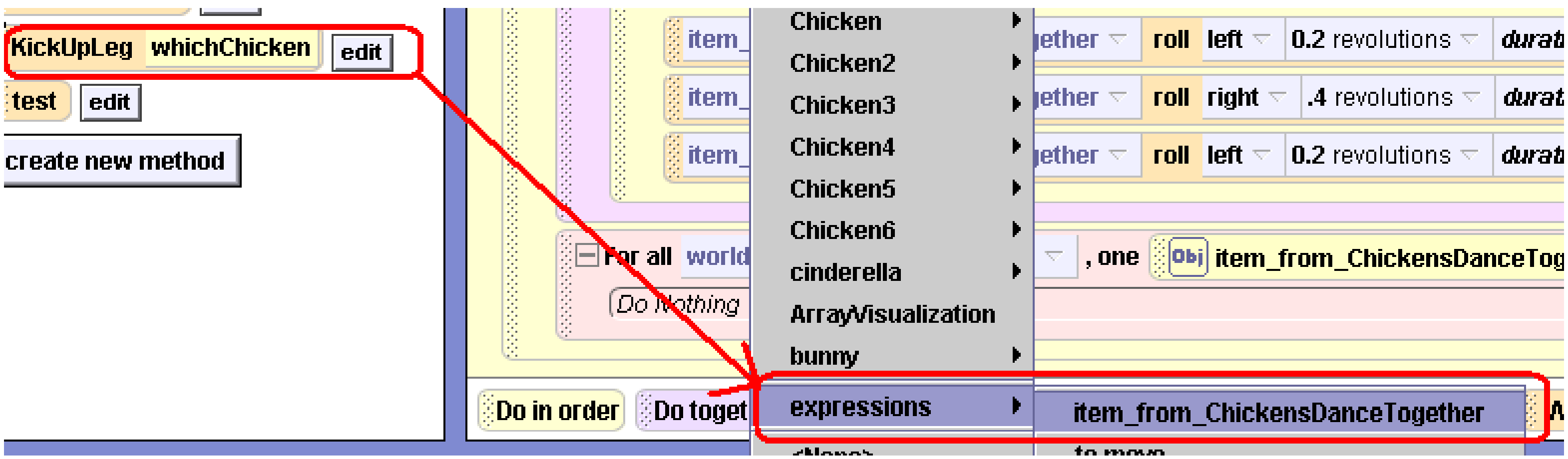
- Finally take your object parameter and drop it back in where it says chicken
- Your code should look like the picture below when you are done





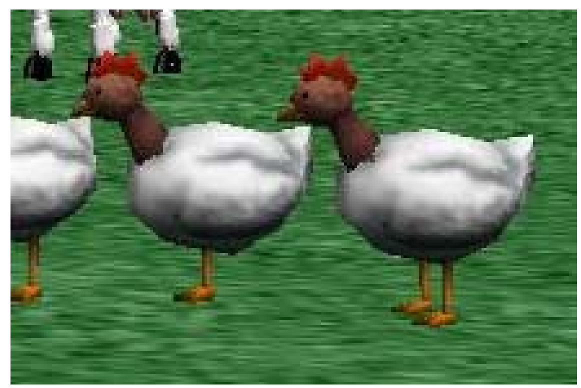
Lists

- Now go back into your “chickmove” method and drag your “KickUpLeg” method into the “For all in Order” space.
- Select Expressions and then Item_from_ChickensDanceTogether



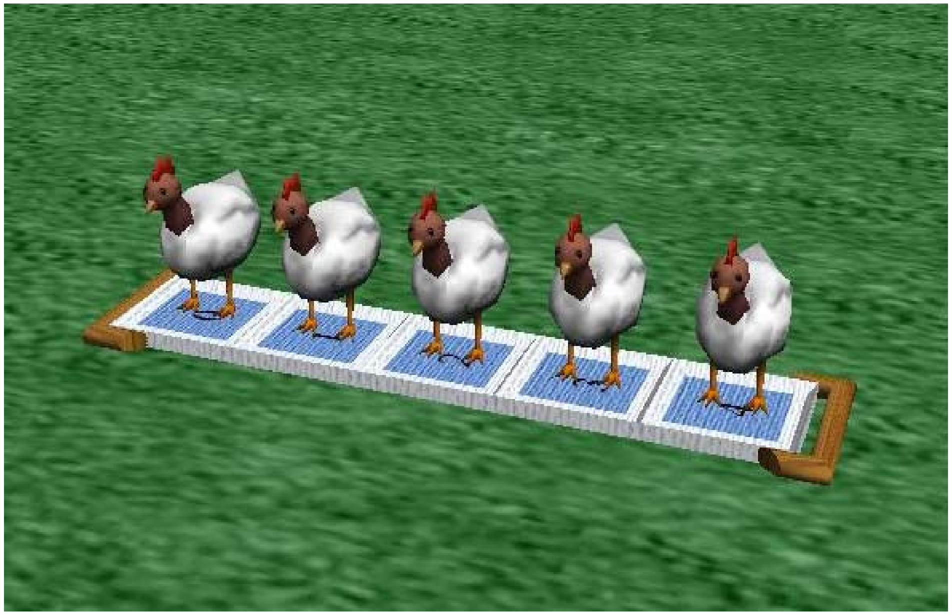
Lists

- Now that this portion is programmed all the chickens will roll right and then left at the same time for the “Together Functions” and kick up their legs one at a time for the “For all in order Functions”
- This saves lots of time and lines of code
- Imagine if we had to program each chicken to do this individually!



Side note on List Visualizations

- If you are not also using an Array and would like to visualize your list you can select the list visualization class from the Visualizations folder
- DO NOT DO THIS RIGHT NOW OR IT WILL MESS UP THE LIST YOU JUST CREATED
- After entering in all of the Chickens they should be aligned in a row that looks like this.
- If you “Set is Showing to False” the List Visualization will disappear, but it is still there.



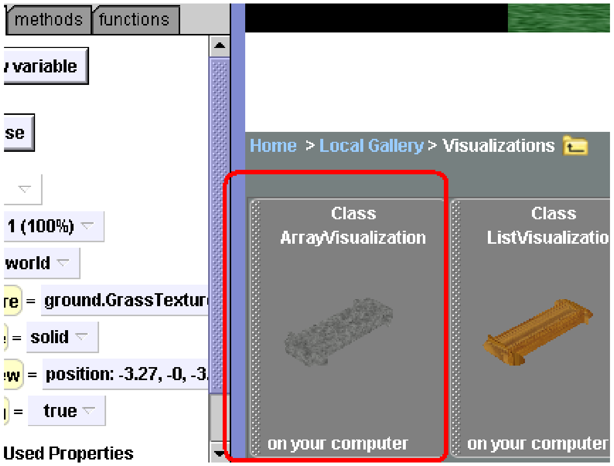
Arrays

- Lists are great for coding a group of objects to all do something.
- But what if we want to code only some of the objects in our group to do something.
- This is where Arrays come in.
- The following slide will show the place that you need to go to find the Array. It is called “Array Visualization”.

Starting the Array

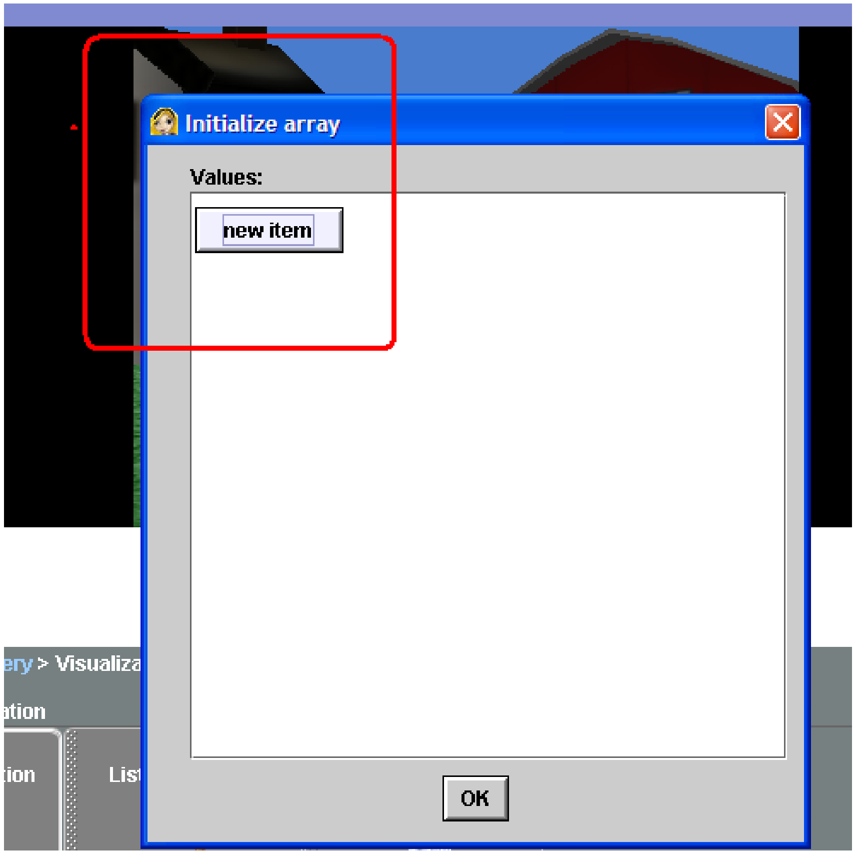
- Go to add objects
- Select the folder at the end of the list entitled “Visualizations”.

- The first object on the left should be a class called “Array Visualizations”.
- It should look like a strip of concrete.
- Select it



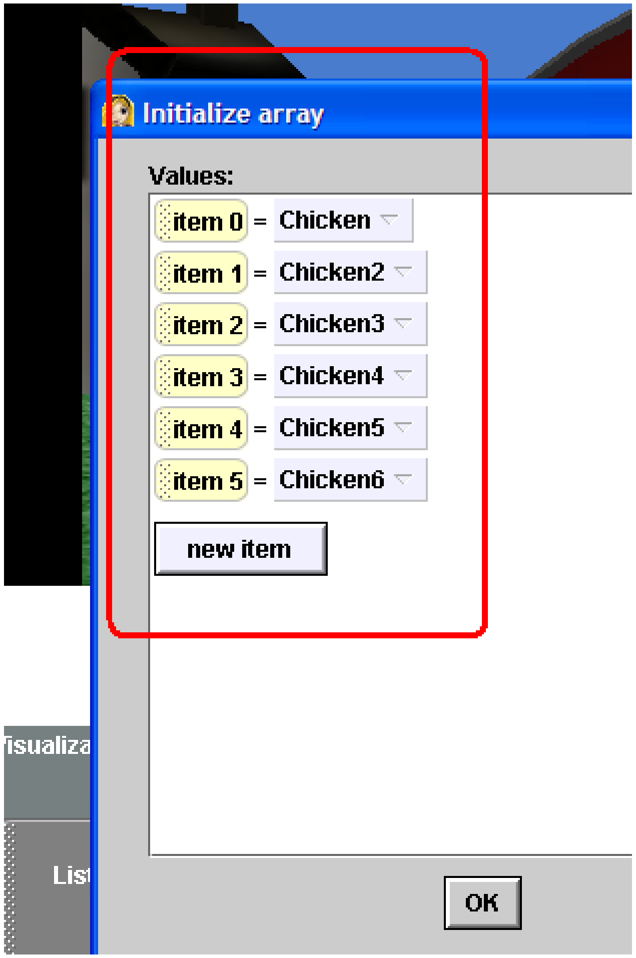
Starting the Array

- We are going to use our Array for the chickens
- When you add the Array to your world this box called “Initializing Array” will appear
- Count the number of chickens you have and select the button “New Item” for that many.
- Remember it will start at 0



Array continued

- Once you have entered all of your chickens into the Array your screen should look like this.
- If it does, click ok. If not, hit “undo” at the top left and try again
- Once you hit ok your chickens should be neatly lined up on the Array.

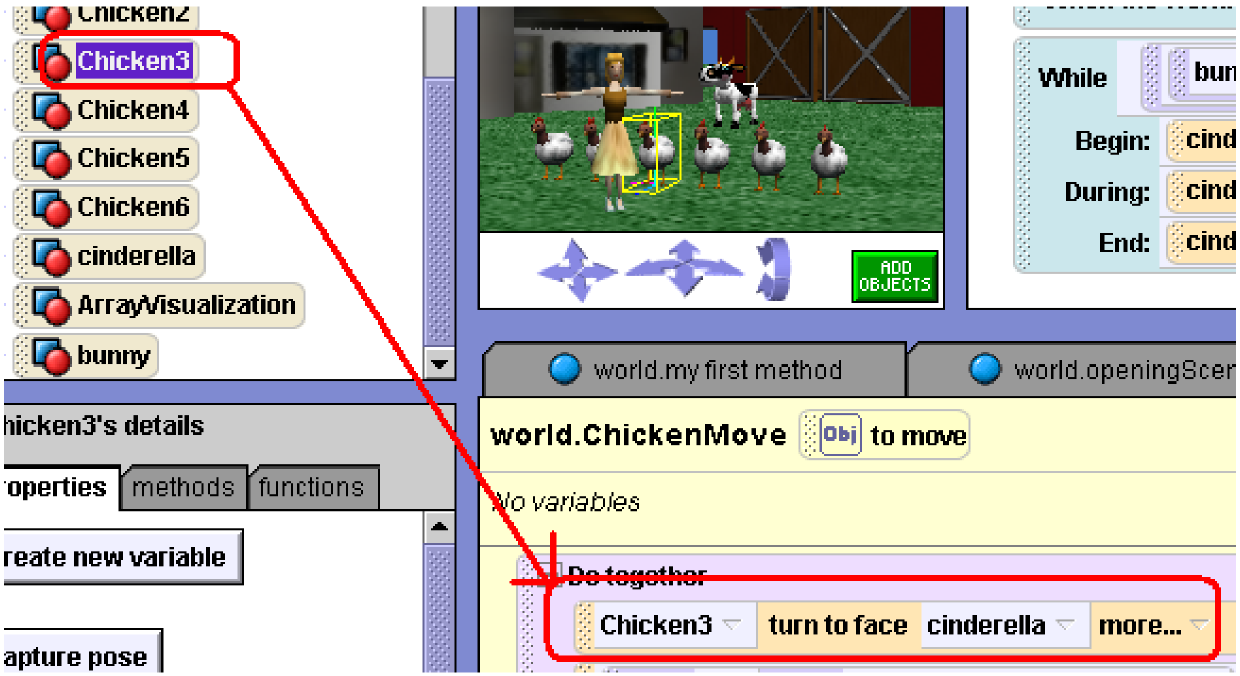


Arrays

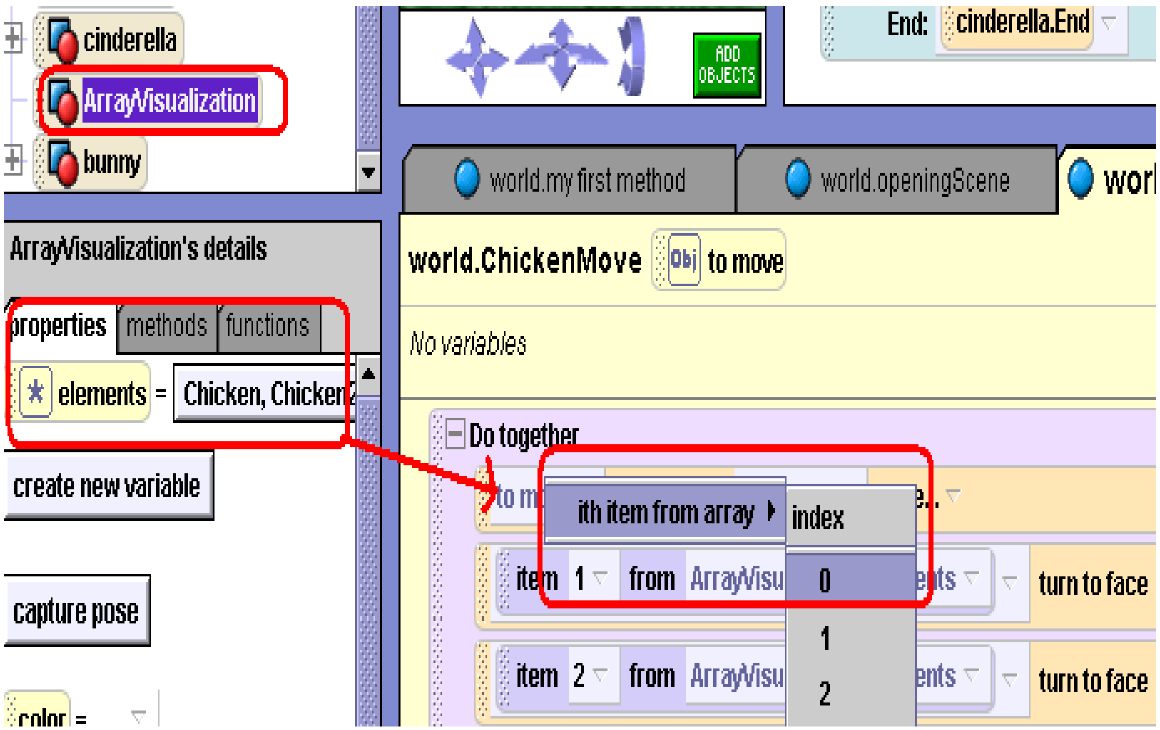
- Side Note: As with the list you can set “Is showing to False” to hide your Array Visualization so that the chickens do not look like they are standing on anything.
- However, the Array is still there and functional

Arrays Continued

- We want all of our chickens to turn to face Alice
- Code the simple method first



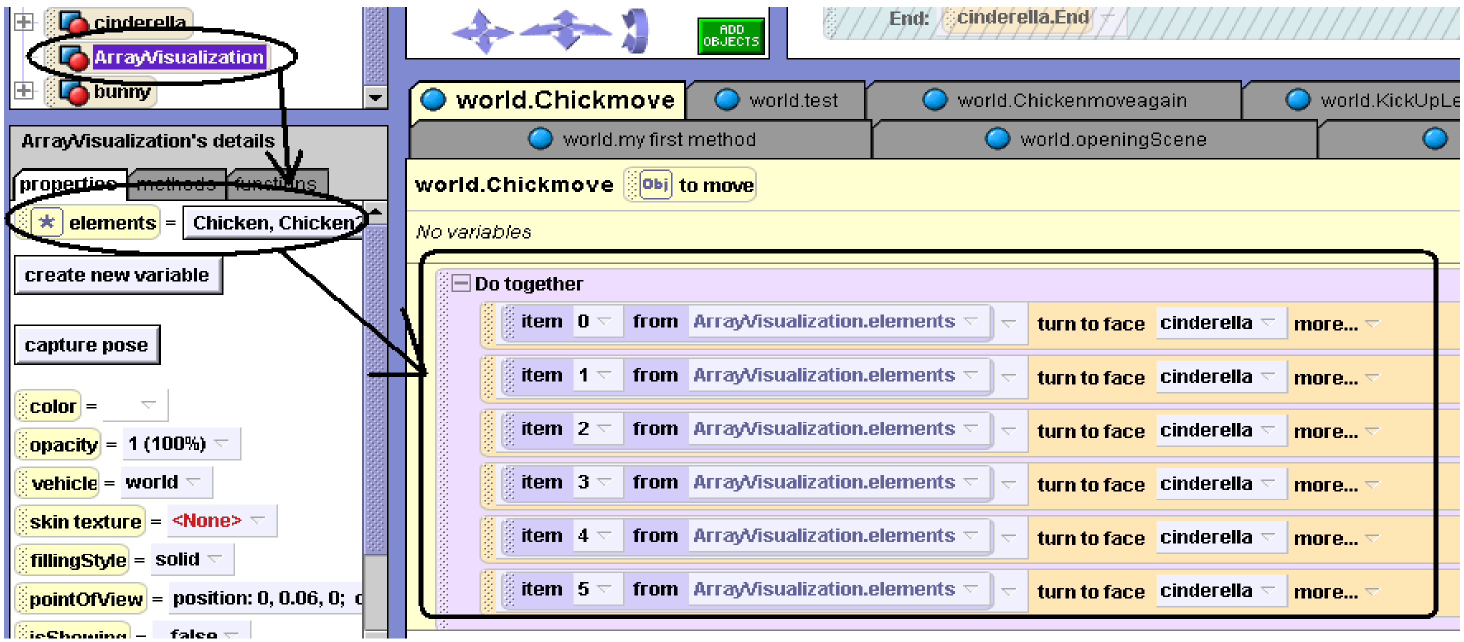
Arrays Continued



- Now go to Array Visualizations, properties, and drag the “elements” tab down to your parameter.
- Select 0, repeat this step for each chicken in your Array.

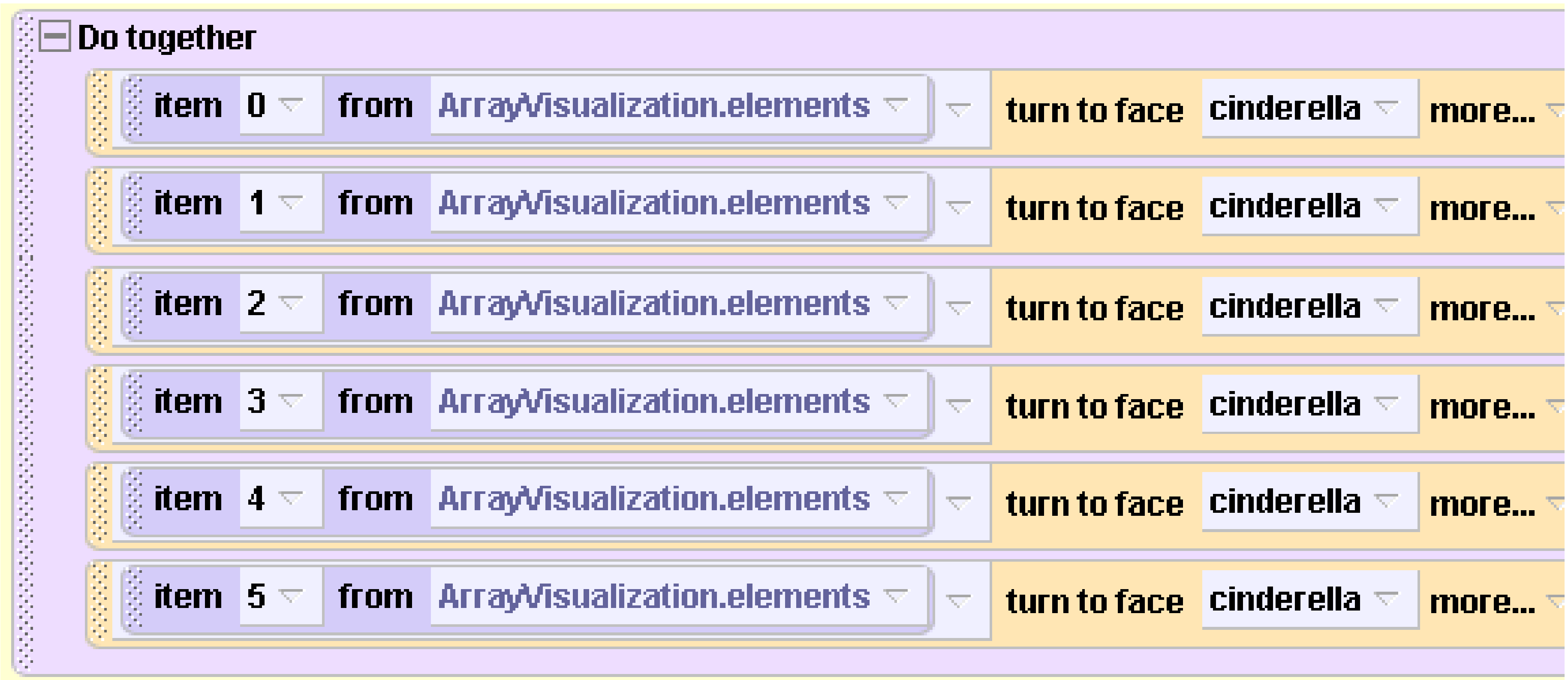
Array Continued

- Note the small circle on the left labeled "elements."
- This is how your array will appear in your tool bar after you have entered it in



- Select each number down the line, till you have all of your chickens in the code.

Complete Turn to Face Code

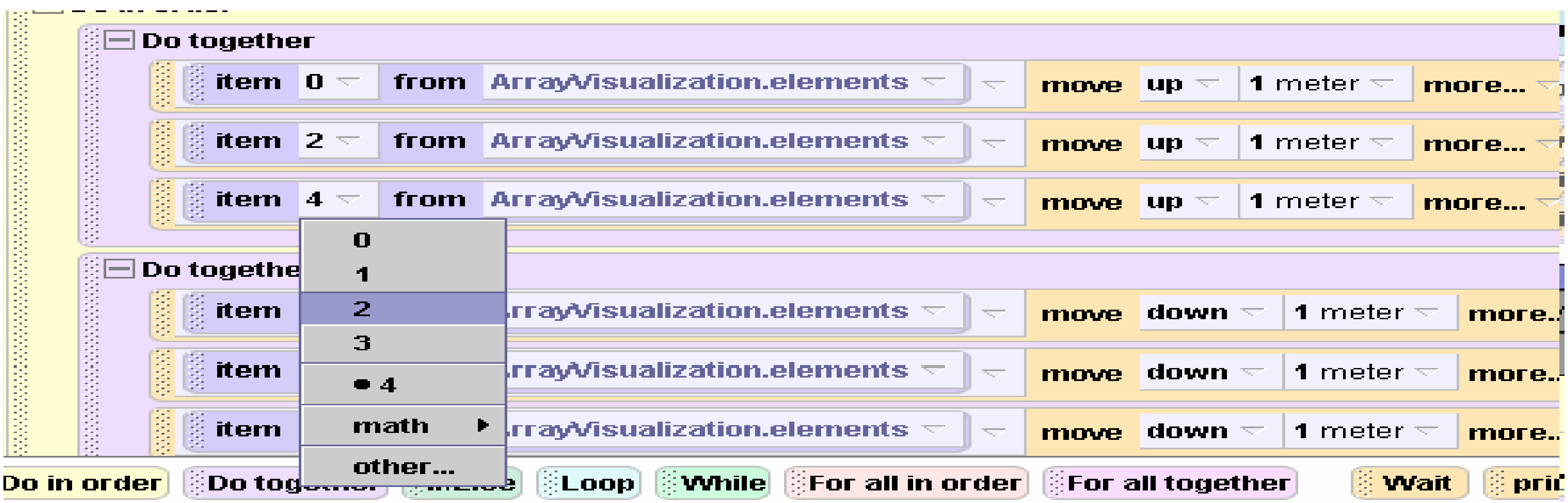


More with Arrays

- Play your animation, the chickens should all turn to face Cinderella.
- Now, lets try separating out the commands
- We want every other chicken to move up and every other chicken to move down. The first step is to code the simple process of moving up and down with any one of the chickens.

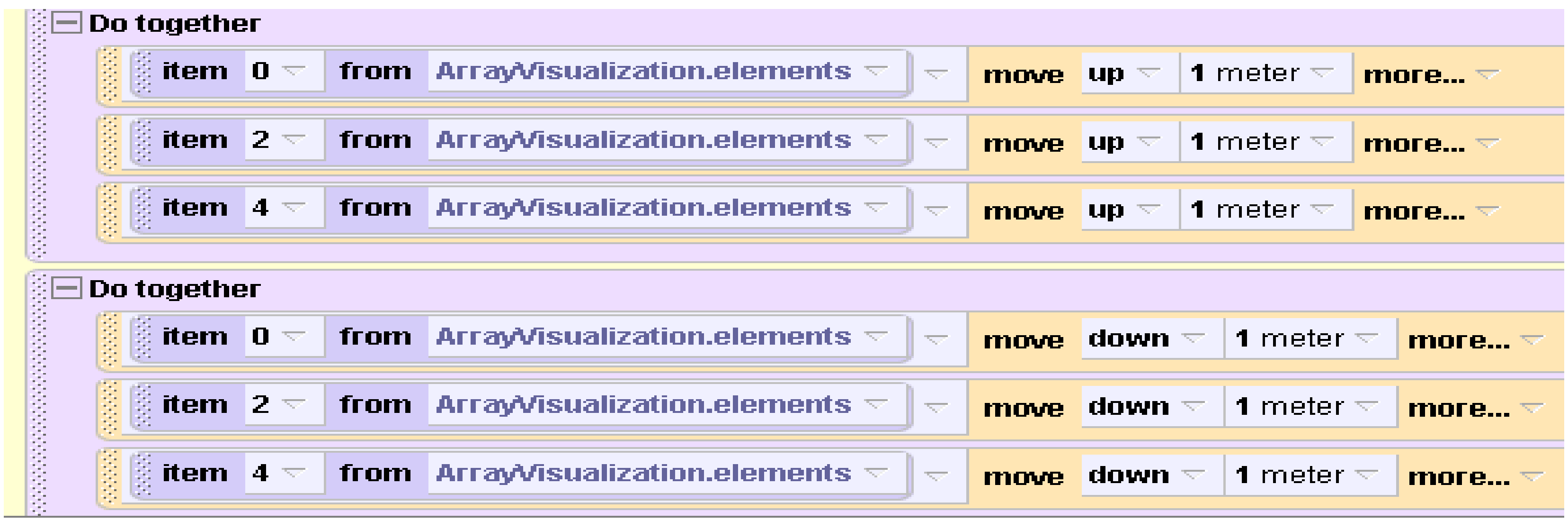
Getting Chickens to do Different Things

- when you add the Array visualization, simply choose certain numbers to do a task rather than all of them.
- Then you add another task and set the rest of the items in the Array to that task



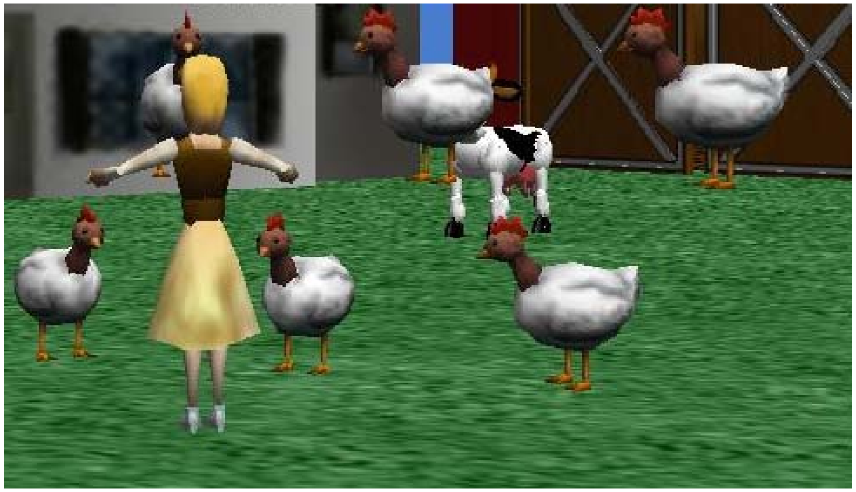
Arrays Continued

- Once you have finished plugging in the Array your code should look like this
- Note that only chickens 0, 2, and 4 are going to move.
- This is the beauty of the Array
- It allows you to only move selected objects in your Array
- Whereas with the List you must move everything

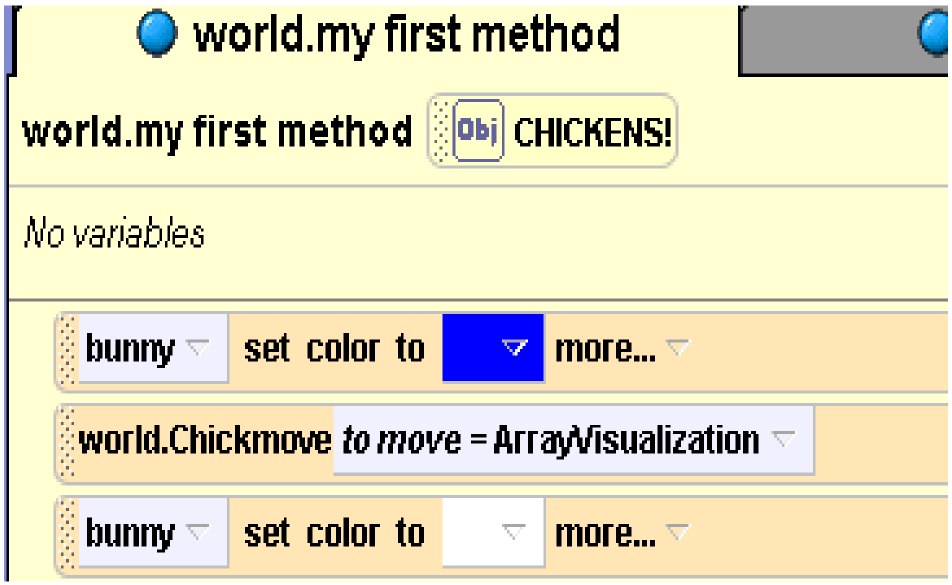


Arrays

- Now play it.
- Note how Chickens 0, 2 and 4 are moving down while chickens 1, 3, and 5 are moving up.
- You can program each chicken individually or in any number that you want.



Arrays

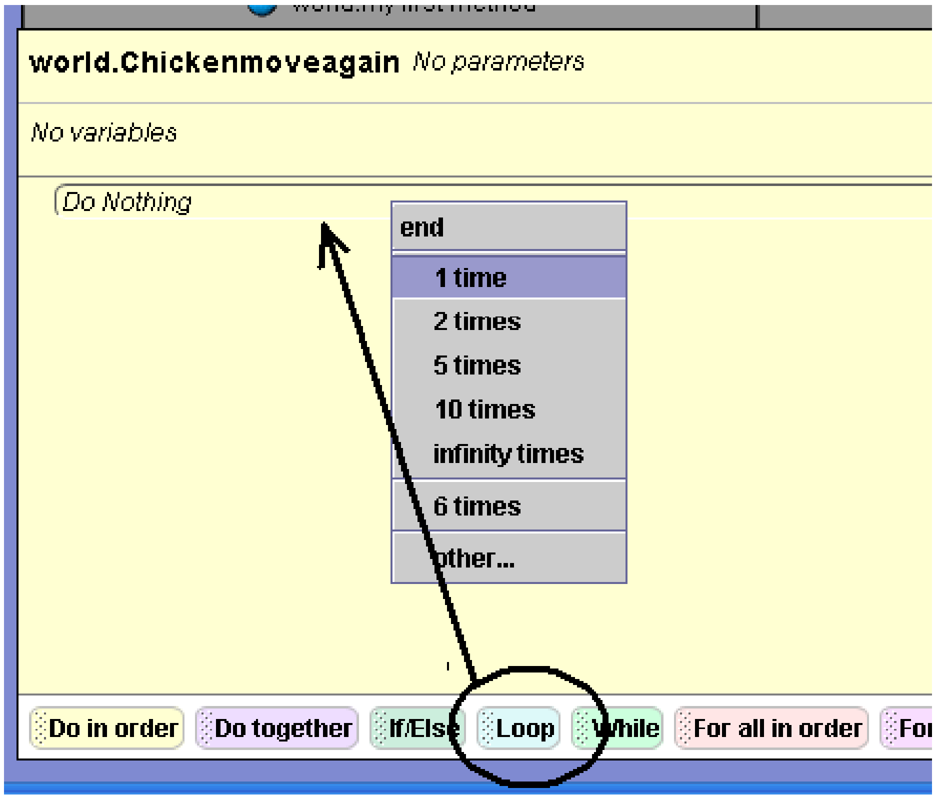


- Finally, lets drop our "chickmove" method into our world.myfirstmethod. So that it will play in our animation.
- For the parameter, select "ArrayVisualization."

The Complicated Loop

- Now we are going to learn a faster way to make Array's work.
- Lets write a new method called "Chicken move again"

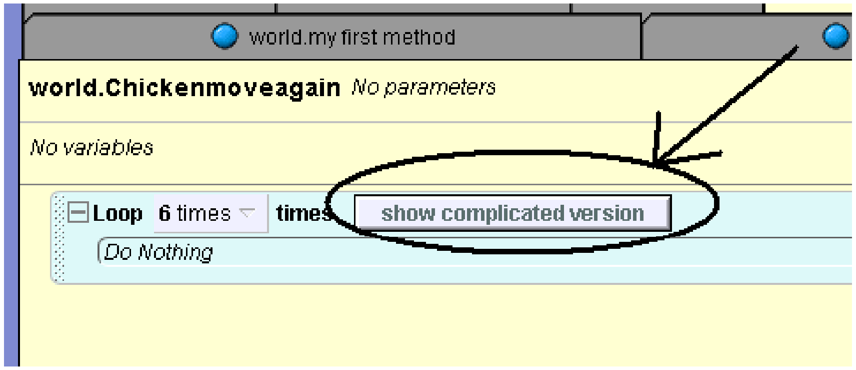
The Complicated Loop

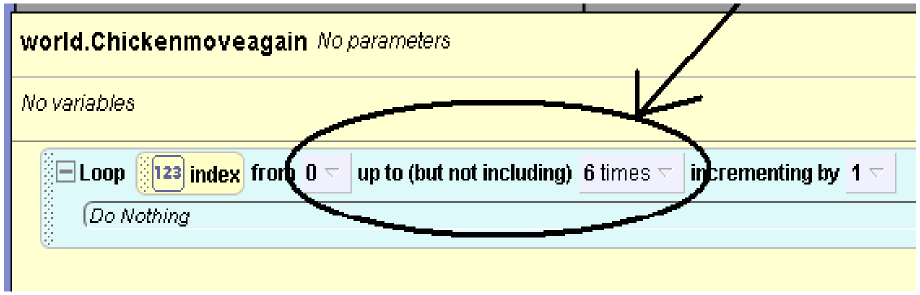


- First drag in a regular loop and select 6 times

The Complicated Loop

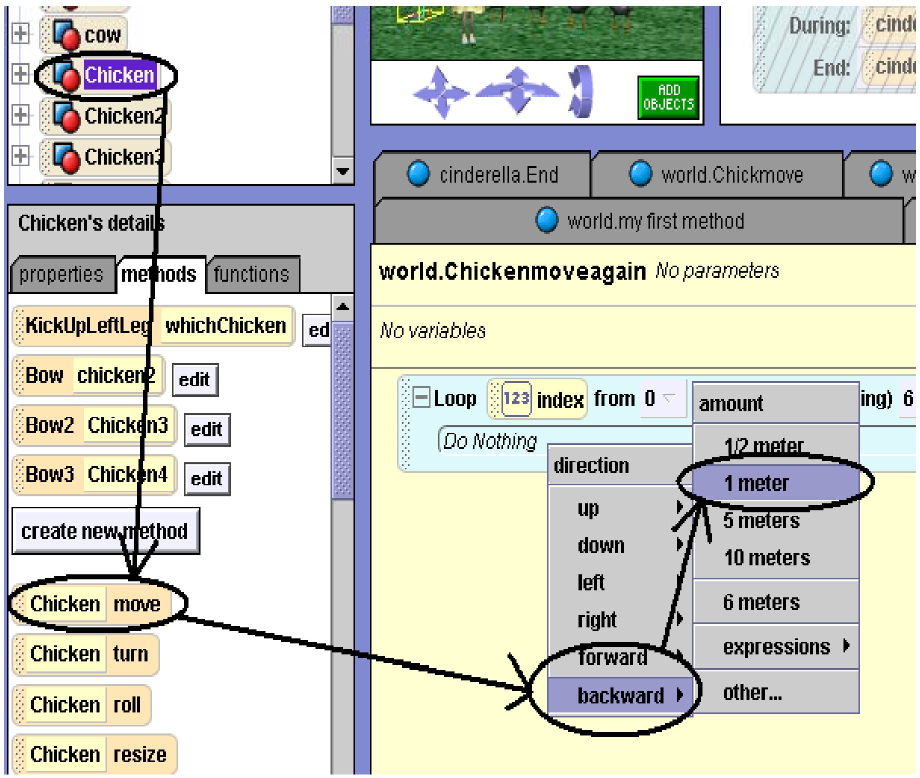
- Now click on the button that says, “Show complicated version.”





- Your loop should now look like this.
- This means that it will loop through chickens in slots 0 – 5.

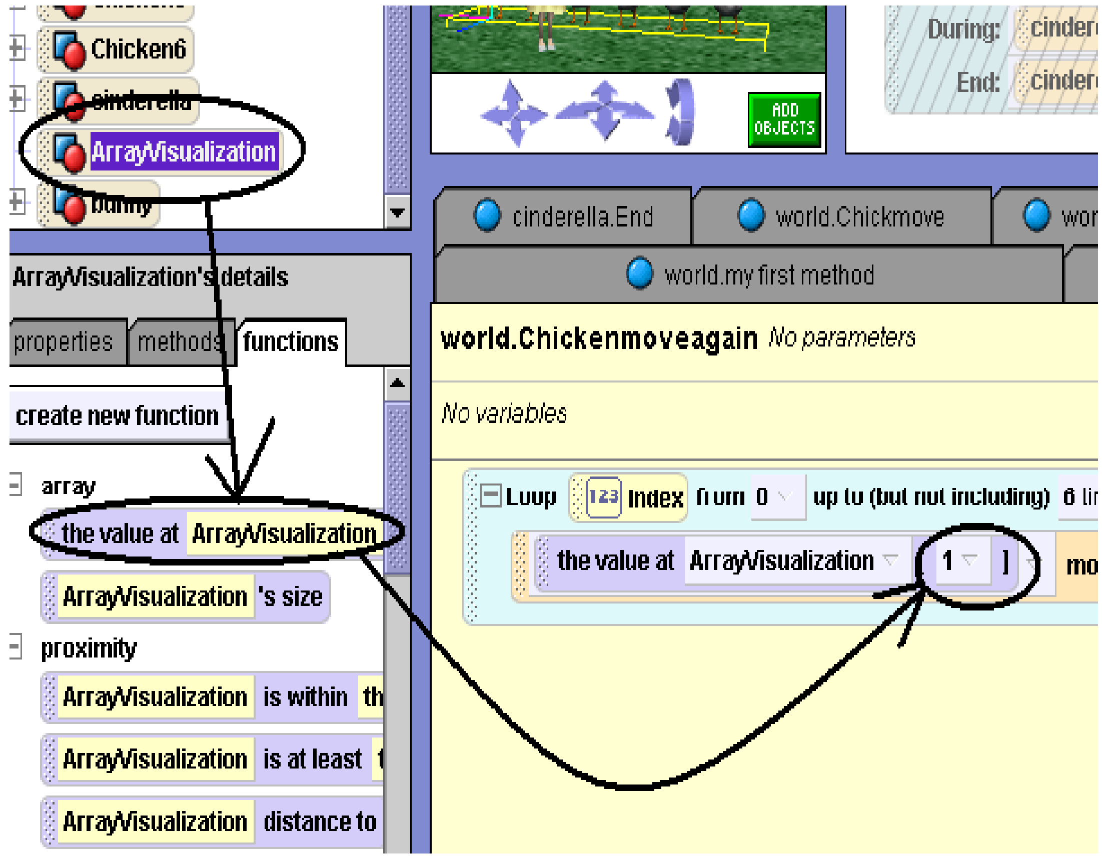
The Complicated Loop



- Now we will put a simple method in our world that tells “chicken” to move back 1 meter.

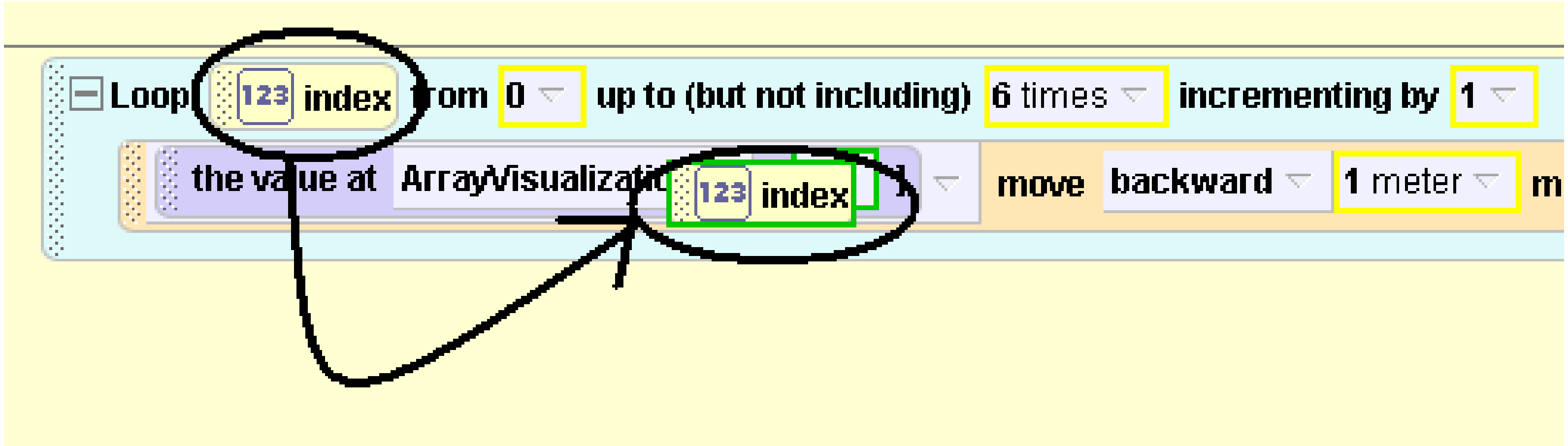
The Complicated Loop

- Go to the functions tab of your Array visualization.
- Drag the function labeled “the value at array visualization” over and drop it on top of “chicken.”
- The number you pick doesn’t matter, it’s just a place holder.



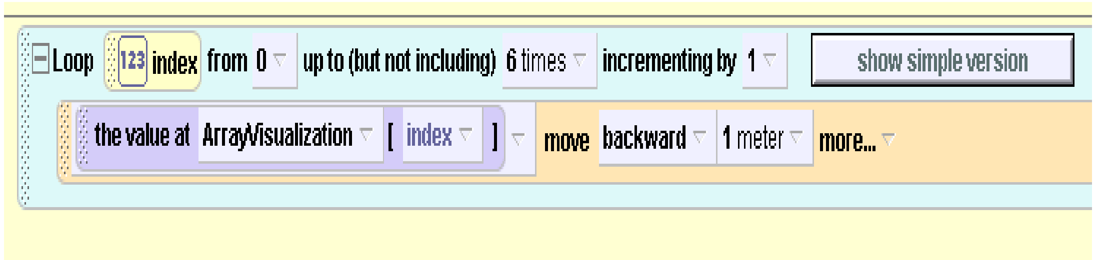
The Complicated Loop

- Now take the “123 Index tab” and drop it over the number that was your place holder



The Complicated Loop

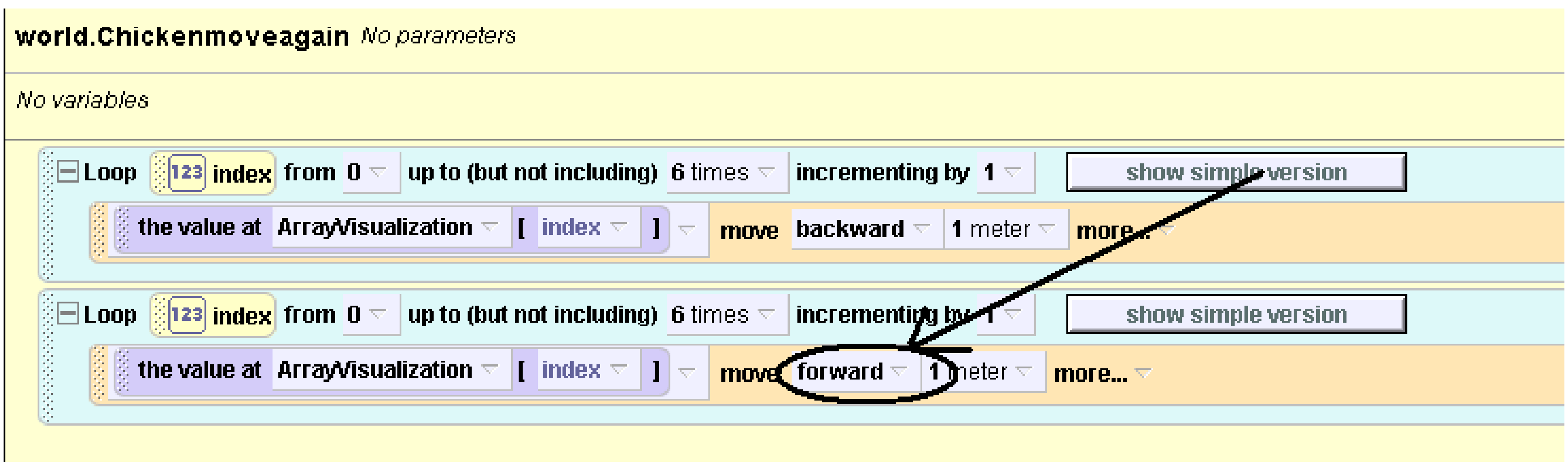
- Once you have done that try playing your “chickenmoveagain” method.



- Your chickens should move back one at a time.
- As you can see this greatly shortens the amount of code that you need to write methods with arrays.

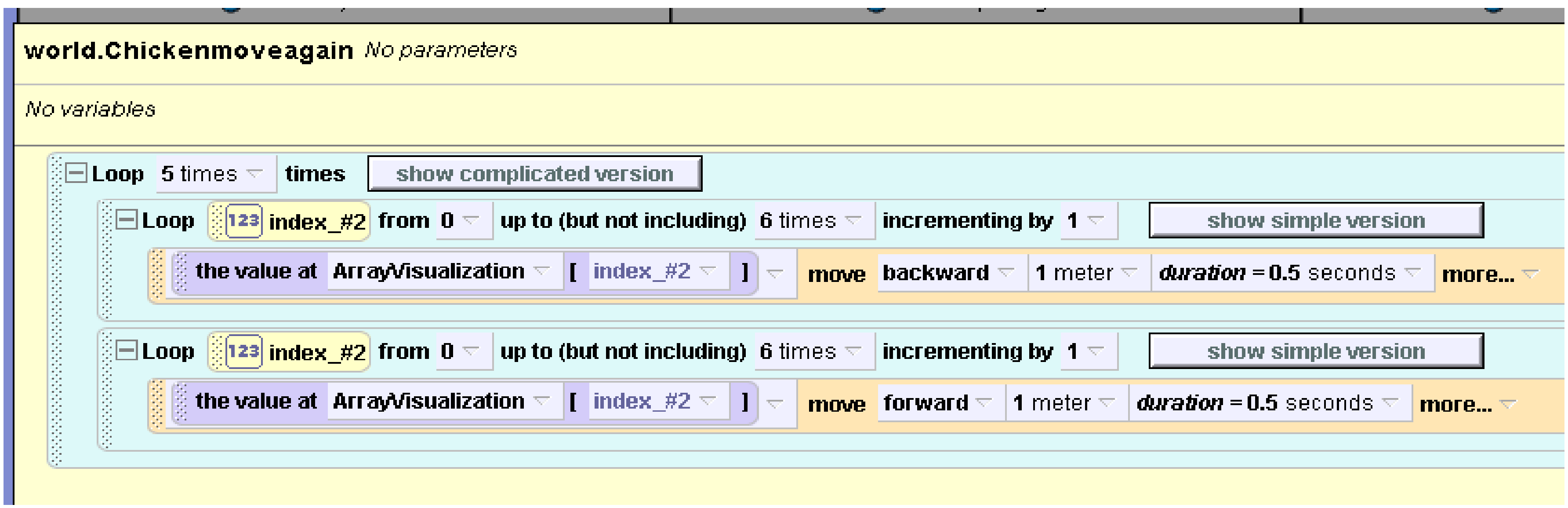
The Complicated Loop

- Make a copy of your loop and reverse the direction.
- This time they should go forward.
- Now try playing your world again



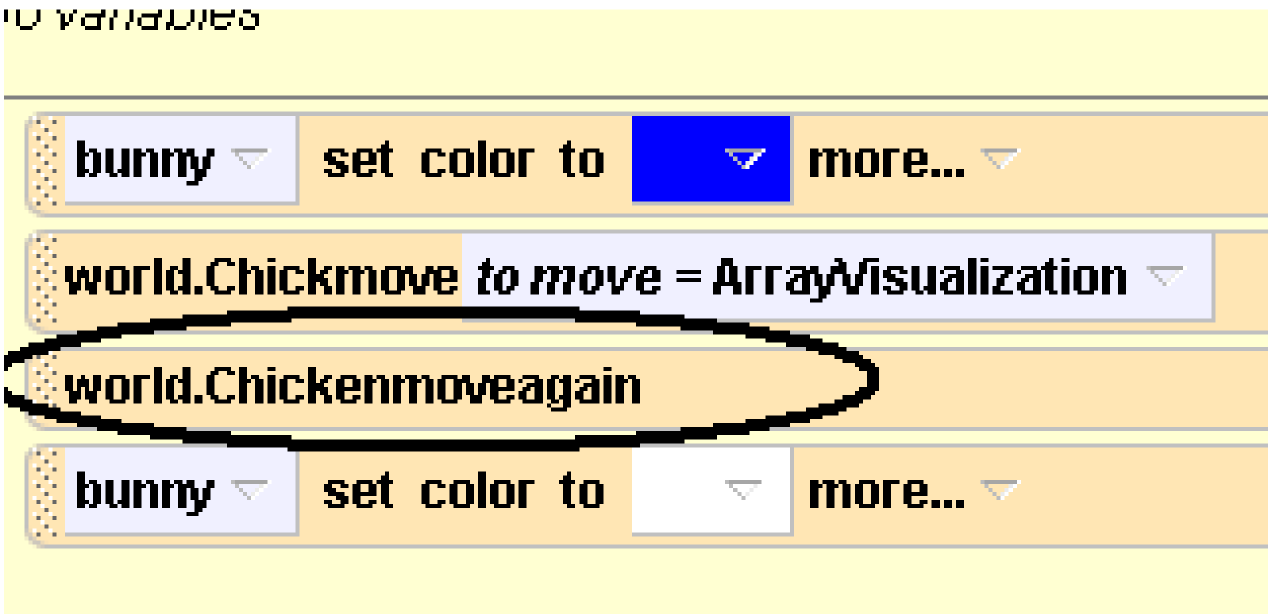
The Complicated Loop

- Play your complete method when you are done.
- Now lets make the animation a little bit more fun by speeding it up and putting a loop around the entire thing.



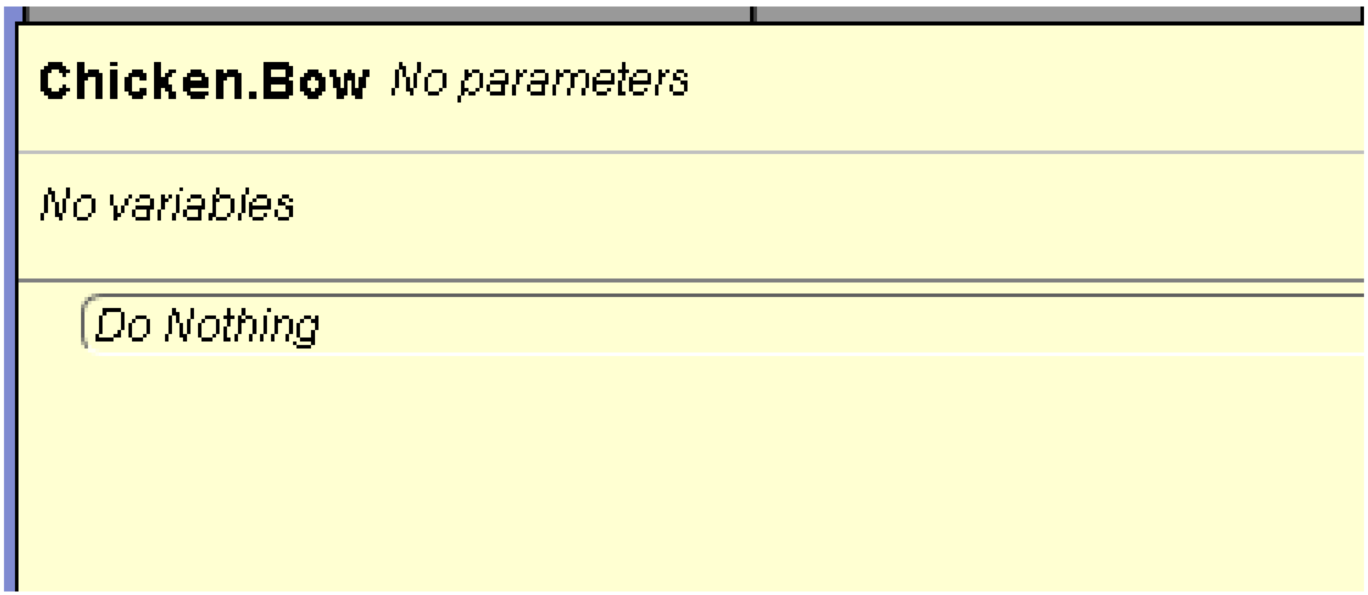
The Complicated Loop

- Now lets put this new method inside of world.myfirstmethod.
- Watch your chickens dance.



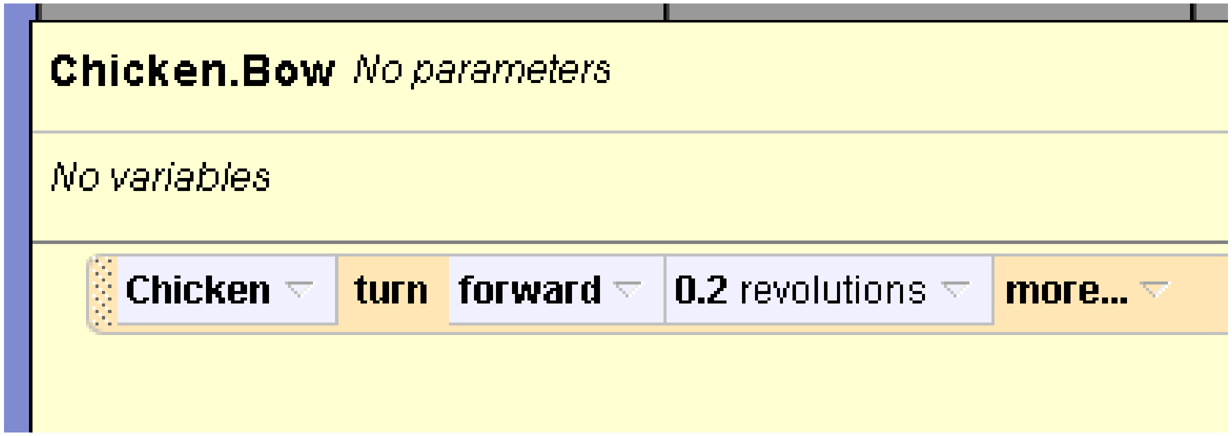
Finishing touches

- To finish up we are going to review lists and have the chickens bow.
- Create a new method under chicken and name it “Chicken.Bow”



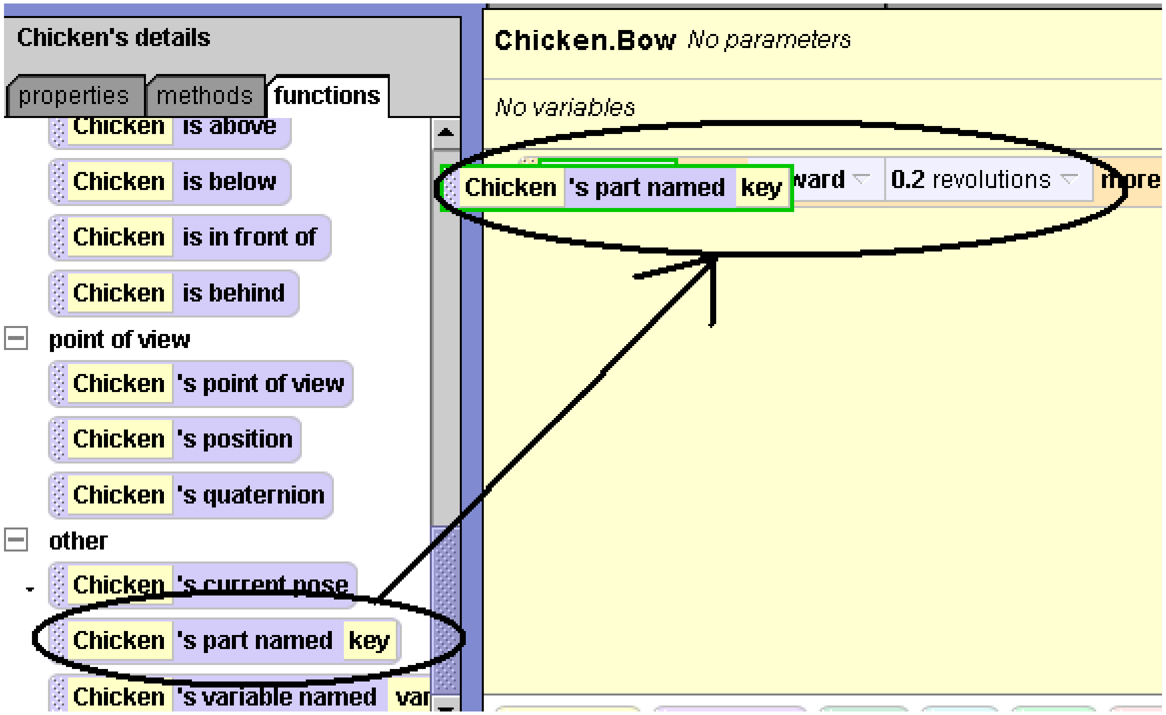
Finishing Touches

- We will now code the chicken to turn forward 0.2 revolutions.



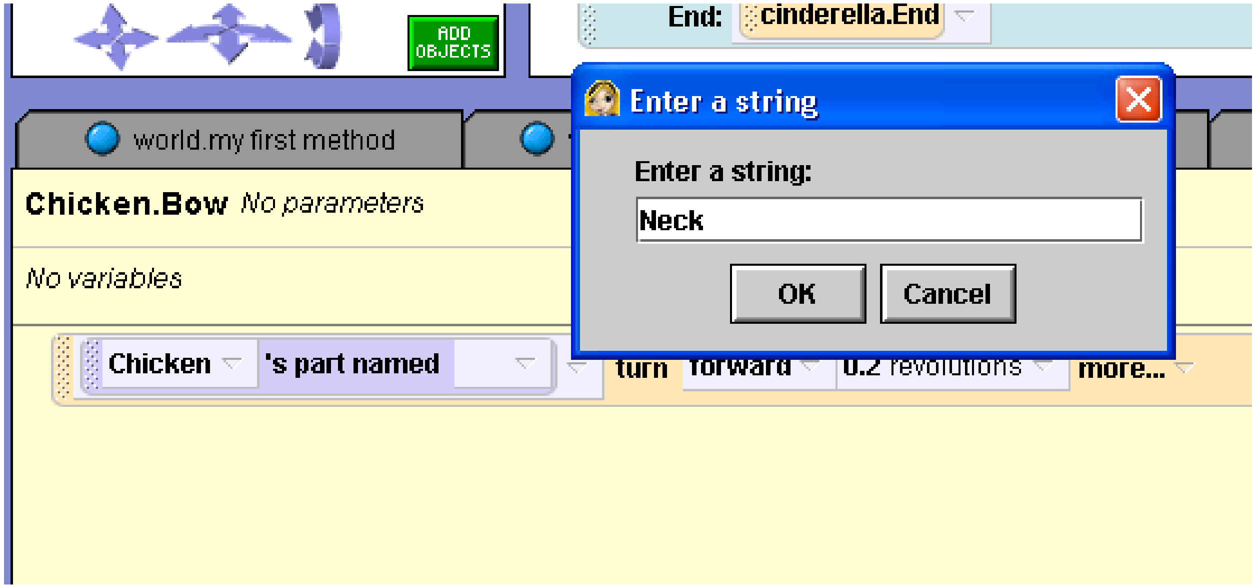
Finishing Touches

- Go to functions and find “parts named key.”
- Drag it on top of chicken



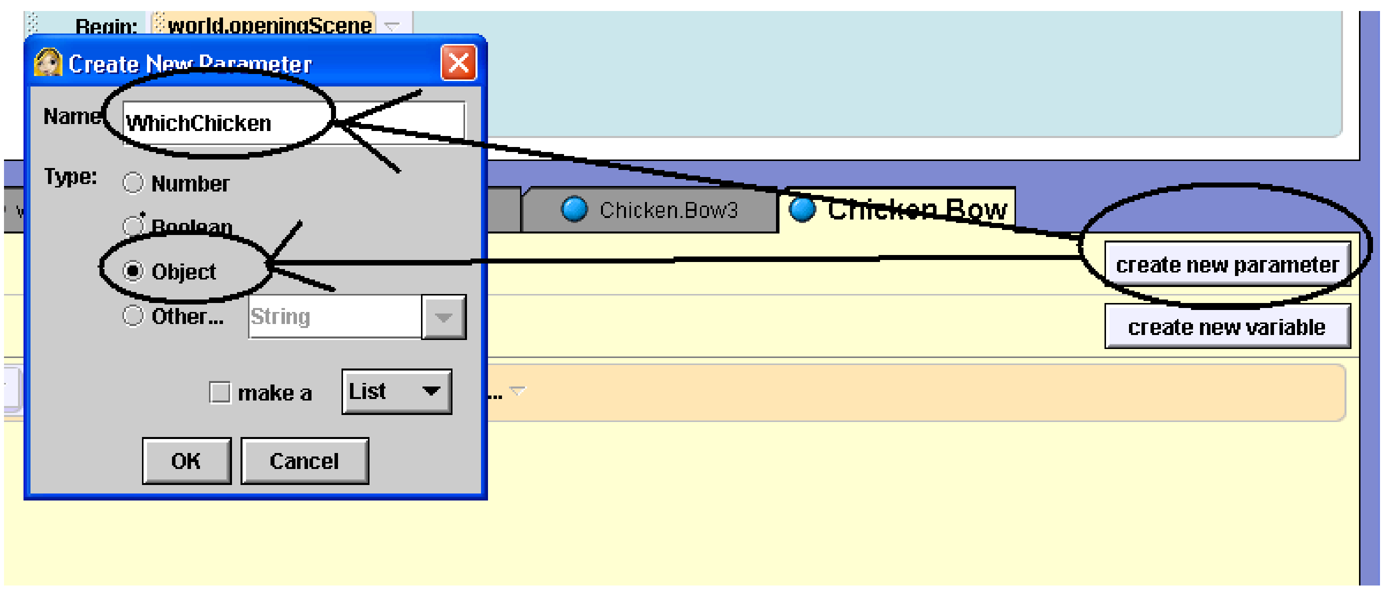
Finishing Touches

- Name the part “Neck”
- Make sure the syntax is correct.



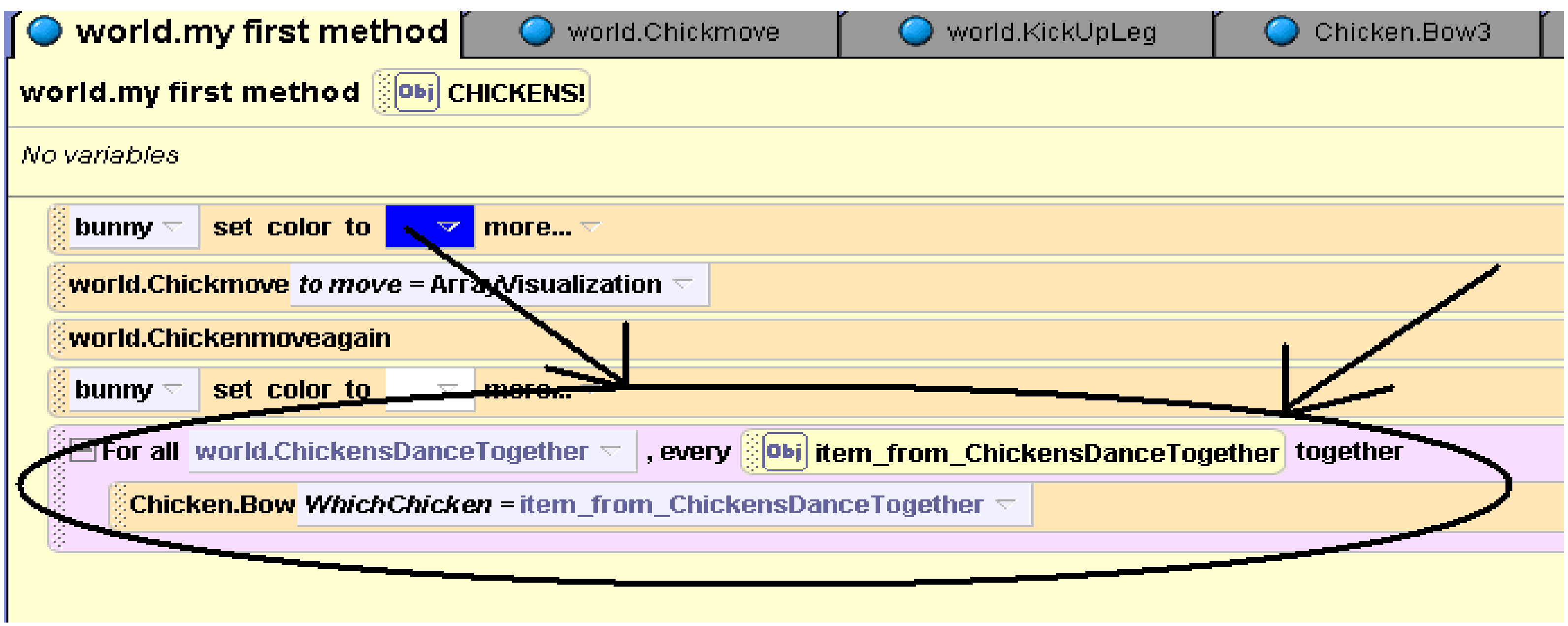
Finishing Touches

- Create a parameter and name it “WhichChicken”
- Make sure its an object parameter



Finishing Touches

- Finally, call your bow method at the end of "world.myfirstmethod."
- Play your world
- *That's all folks!!!!!!!!!*



Recap

- When we put all of these elements together you can see how Lists and Arrays can be used interchangeably to animate your Alice world.
- The ‘For all together’ command in Lists allows you to animate a group of objects to do something together.
- The ‘For all in order’ command in Lists allows you to animate the whole group of objects to do the same thing, but one at a time.
- The Array allows you to have a group of objects do different things.