Animating Objects in Groups: Using Arrays and Lists

By Ruthie Tucker
under the direction of Professor Susan Rodger
Summer 2008

Description

- This tutorial will cover the basics of using arrays and lists in an Alice world.
- Arrays and lists are very useful ways to put objects into groups.
- It uses a set of chickens on a farm to illustrate these concepts.

Starting Out

- To start, download the starting world for this tutorial, and save it in a directory that you can find again. Open up Alice, and then open the world. You will see several objects already in the world for you.

What is a List?
- A list is, you guessed it, a list of objects that you can have perform the same actions simultaneously, or one by one.

What is an Array?
- An Array is very similar to a list, except in an array, each item has a numbered spot in the array.
- In Alice, Arrays and Lists are also visual objects which can be dropped into your world.
Making a List

- We are going to create a list, and use it to make our chickens move in unison.
- Go to the world’s properties tab and click the create new variable button. Name the variable ChickensDanceTogether, and select Object as the type.
- Then, click the box next to the words make a List.

A white area will appear at the bottom of the gray box. Click the new item button, and an item 0 will appear. Set item0 to Chicken by clicking on the down arrow next to <None>, and selecting Chicken, and then the entire Chicken from the list. Add another item, and set it to Chicken2, and so on, until all of the chickens through Chicken6 are on the list. Then click OK.

Moving the chickens together

- Now we will create a method to move the chickens together using the list. Create a world-level method called chickensTogether.
- Find the For all together button at the bottom of the method editor, and drag it into chickensTogether.
- From the menu that pops up, select expressions, then world.Chickens DanceTogether.

World.chickensTogether

- Drag a Do in order into the For all together statement.
- Add the following code inside the Do in order. This code will make a chicken roll left, then right, then left. Use any chicken to write this code, it doesn’t matter which one, because we will be replacing it soon. For our example, we have used Chicken.
Lists

- Now drag the button labeled `item_from_ChickensDanceTogether` over the button labeled `Chicken`. This means that instead of just `Chicken` executing this method, now each chicken from the list will execute it together.
- Drag `item_from_ChickensDanceTogether` over each of the other two buttons labeled `Chicken` so that each chicken performs the whole sequence of methods.

Now we have programmed the list to have all the chickens roll left and then right at the same time. Drag `world.chickensTogether` into `world.my first method` and then play your world to make sure your code does this.

- Now 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 each kick up their left leg, one at a time.

Lists

- Find the `For all in order` button at the bottom of your method editor and drag and drop it into your `world.chickensTogether` method.
- Select `expressions`, and then `world. Chickens Dance Together`.

Lists

- Drag a `Do in order` into the `For all in order` statement.
- Add the following code inside the `Do in order`. We will modify this code so that it makes each chicken kick up their left leg. Use any chicken to write this code, it doesn’t matter which one, because we will be replacing it soon. For our example, we have used `Chicken`.

Lists

- Drag a `Do in order` into the `For all in order` statement.
- Add the following code inside the `Do in order`. We will modify this code so that it makes each chicken kick up their left leg. Use any chicken to write this code, it doesn’t matter which one, because we will be replacing it soon. For our example, we have used `Chicken`.

Go to the chicken’s functions tab. Scroll down until you see the function Chicken’s part named key. Drag this function into your method editor and drop it over Chicken on each of the lines of code in the For all in order. This function can be used to make only a part of an object in a list move, instead of the whole thing.

Click on the plus sign next to Chicken in the object tree so that you can see the parts of the chicken. Now, click on the down arrow next to the word named on the first Chicken’s part named statement. Select other.... Type in LeftLeg, exactly how it is spelled in the object tree, with the same capitalizations. Do the same thing to the other Chicken’s part named statements.

Now, click on the item_from_ChickensDanceTogether button at the top of the For all in order statement, and drag and drop it over the word Chicken in the first Chicken’s part named LeftLeg statement. This will make each object in the list’s left leg execute this line of code one at a time, instead of just Chicken. Do the same thing for the other two lines of code in the For all in order.

Play your world to see the results.

Now that this portion is programmed, all the chickens will roll right and then left at the same time for the For all together code, and kick up their legs one at a time for the For all in order code.

This saves lots of time and lines of code.

Imagine if we had to program each chicken to do this individually!
List Visualizations

- For future reference, there is an object called ListVisualization that is a visual representation of the list that your objects are in that you can drop into your world.
- Do not do this right now, because it will cause problems for the array that we will be creating shortly.
- If you were to enter all of the chickens into a ListVisualization, they would be aligned in a row that looks like this.
- If you wanted your chickens to stay in their neat line, but did not want to see the ListVisualization, you could set its isShowing to False.

Arrays

- Lists are great for coding a group of objects to all do the same thing (either together or one at a time).
- 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 ArrayVisualization.

Starting the Array

- Click on Add Objects.
- Scroll across until you see a folder called Visualizations.
- The first object on the left should be a class called ArrayVisualization.
- It should look like a strip of concrete.
- Add one of these to your world.

Starting the Array

- We are going to use our array for the chickens.
- When you add the ArrayVisualization to your world, this box labeled Initialize Array will appear.
- Count the number of chickens you have and click the new item button that many times.
- The array will start counting at 0, so the number on your last item button will be one less than how many chickens you have.
Arrays

- Enter your chickens into the array in the same way that you did for the list. Once you have done so, your box should look like this.
- If it does, click OK. If not, use the Undo button at the top left of your screen and try again.
- Once you click OK, your chickens should be neatly lined up on the array.

Arrays

- You may have to re-position the chickens a bit so that they are all showing on the screen.
- Click on your ArrayVisualization and then go to its properties tab. Set its isShowing property to false to hide your ArrayVisualization so that the chickens do not look like they are standing on anything.
- Even though it is invisible, the array is still there and functional.

Arrays

- Drag a Do Together into your chickensTogether method.
- Then, drag the following code into the Do Together. Use any chicken when writing this code, we will be replacing it shortly. You can do it by writing the line of code once, and then right-clicking on it and copying it until there are 6.

Arrays Continued

- Now click on ArrayVisualization in the object tree, and go to its properties tab.
- Click on the button at the top that says elements, and drag and drop it on top of Chicken in the first line of code in your Do together.
- Select the number 0 from the list of numbers. This represents the object that is in the 0 slot of the array.
Arrays Continued

- Do this for each of the lines of code in the Do together, selecting the number in order through number 5. Your code will look like this when you’ve finished.

```
Array Visualization elements — turn to face center left
Array Visualization elements — turn to face center left
Array Visualization elements — turn to face center left
Array Visualization elements — turn to face center left
Array Visualization elements — turn to face center left
```

Getting Chickens to do different things

- Right now, each chicken in the array is doing the same thing at the same time. Let’s use the array to make only half of the chickens do something.
- Add two more Do Togethers to your chickens Together method, one right below the other.

```
world.ChickensTogether[0].parameters
```

Arrays

- Enter the code seen below into the two Do Togethers, using the same method as for the previous code.
- Note that only chickens in the 0, 2, and 4 slots 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.

```
Do Together
Item 0 Item 1 Item 2 Item 3 Item 4 Item 5
Item 0 Item 1 Item 2 Item 3 Item 4 Item 5
Item 0 Item 1 Item 2 Item 3 Item 4 Item 5
Item 0 Item 1 Item 2 Item 3 Item 4 Item 5
Item 0 Item 1 Item 2 Item 3 Item 4 Item 5
```

Arrays

- Play your world. The latest addition should look something like the picture below.
- Only the chickens in the even numbered slots are moving.

```
Do Together
Do Nothing
Do Nothing
```
Recap

- After finishing this tutorial, you can see how lists and arrays can be used interchangeably to animate your Alice world.
- The **For all together** command for lists allows you to animate a group of objects to do something together.
- The **For all in order** command for 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.