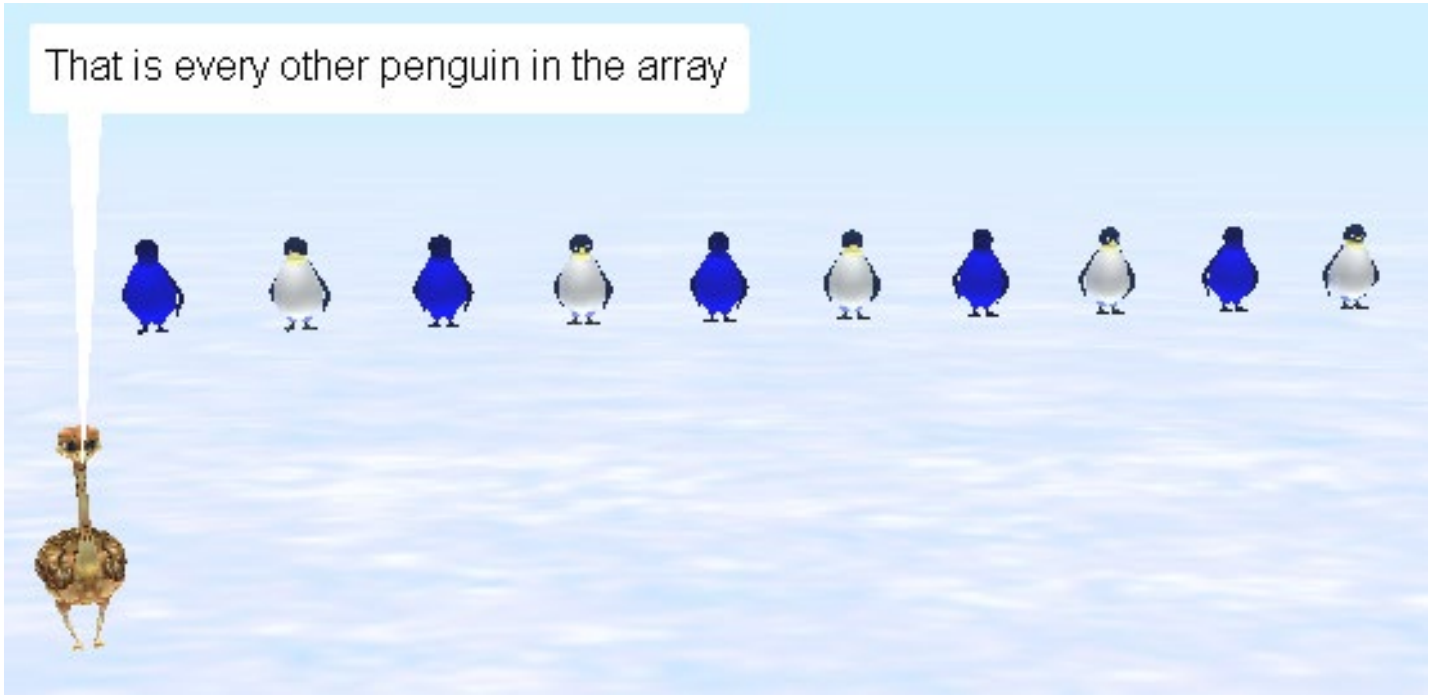
CompSci 94
Classwork: Array Indexing
Oct 29, 2024



Prof. Susan Rodger

Overview of the Story

- The penguins are scattered around. They will come together and then spread out into a line. Then every other one will turn blue, and the others then will turn red. Then the penguins are randomly colored one of four colors. The ostrich will tell you how many there are of each color, and then the first of each color will tell you they are the first.
- **Follow the steps that follow** to write this program.

1) Setting up the scene

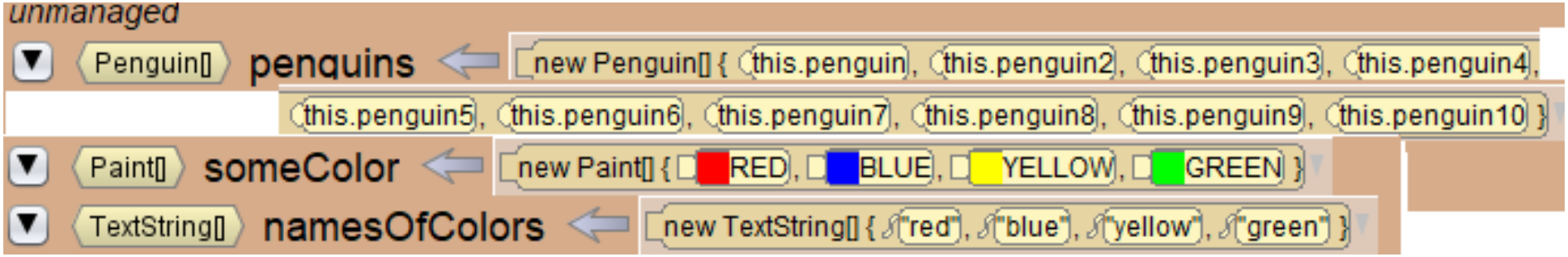
- Start with the starter program that is called **classworkOct29starter.a3p**, then skip to step 2)
- If you want to build the program, then follow these steps:
  - Add in any ground cover, maybe snow?
  - Drag in these objects so they are in positions similar to the picture on the next page
    - Flyer: ten penguins, and one baby ostrich

1) Setup (continued)

- Put the ostrichBaby in the front left.
- Put penguin4 in the back left. Scatter the rest of the penguins

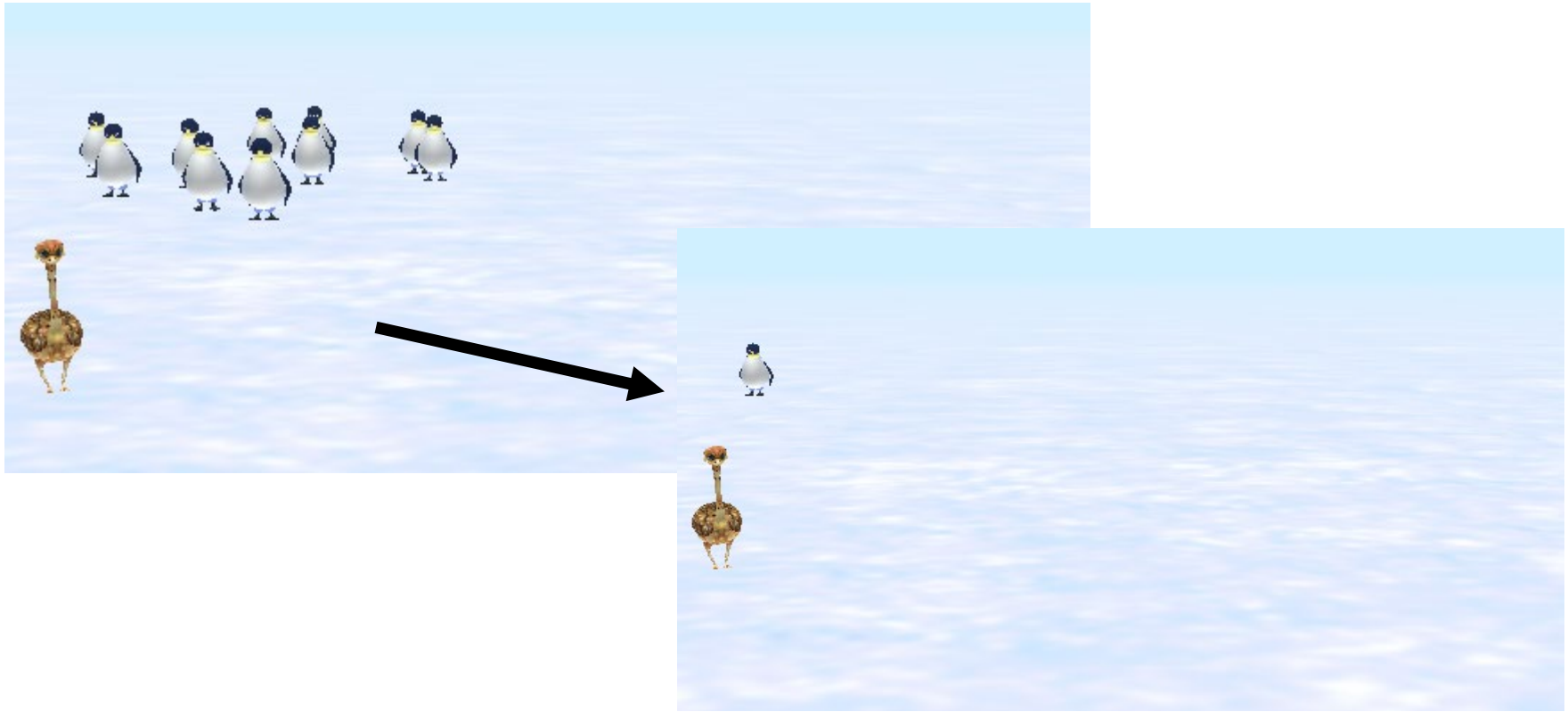
1) Setup continued

- Create three Scene arrays
  - An array of penguins named “penguins”. Put them in in order: penguin, penguin2, etc.
  - An array paints of any four colors
  - An array of TextString of the names of the four colors, in the same order



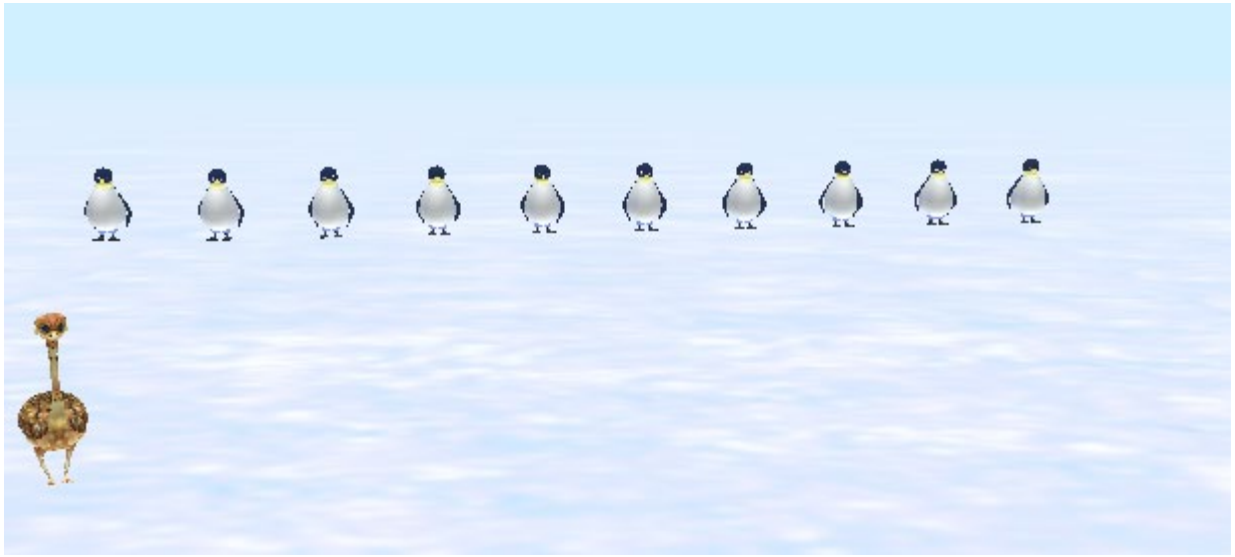
2) Write the **Scene** procedure named **setup**

- This procedure should do the following to setup the penguins.
- First have all the penguins at the same time move and orient to penguin4 (which is to the back left).



2) Setup Procedure continued

- Then spread the penguins out in a line by having them move left specific amounts: penguin move 0, penguin2 move 1, penguin3 move 2, penguin4 move 3, penguin5 move 4, etc.
- Notice each penguin moves 1 more than the previous penguin
- Hint on the next page – use an array index loop!



2) Setup Procedure continued

- To spread the penguins out in a line, use a count loop with a variable.

Set an index variable to 0

Repeat for each penguin

    move the penguin at the index position left

    update the index variable

3) In myFirstMethod

- Drag in a do In Order
- Call the setup procedure
- Run your program!

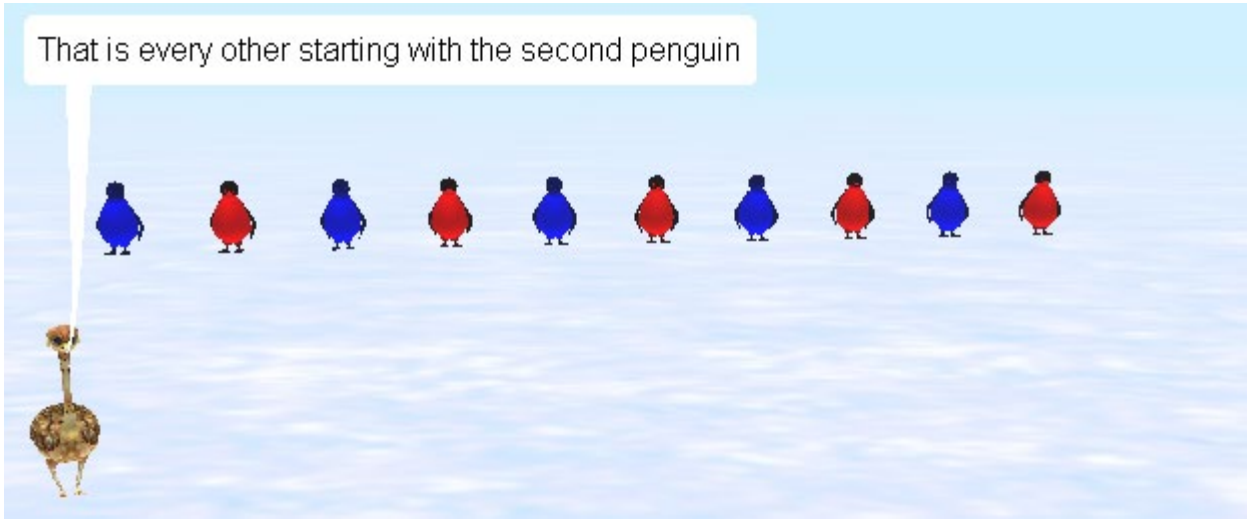
4) Add code in myFirstMethod to do to every other penguin: move up/down and paint blue

- *Create an index variable and set it to 0*
- *While index is smaller than the length of penguin array*
  - *Move the penguin at the index position up and then down*
  - *Paint the penguin at the index position blue*
  - *Update the index*
- *OstrichBaby should say "that is every other penguin in the array"*

4) Picture of every other penguin painted Blue

5) Now go through the array again and do to every other penguin starting with the second penguin: move up/down and paint red

- Still in myFirstMethod. Start with the second penguin and move it up and down and then paint it red. Then do the same to the 4th penguin, and then the 6th penguin, etc.
- Then have ostrichBaby say "That is every other starting with the second penguin"



6) Write the **penguin procedure** named **randomlyColor**

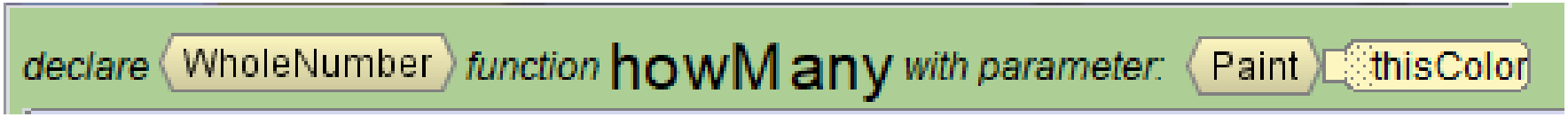
- It has no parameters.
- This procedure randomly colors the penguin one of four colors (use the same colors from the colors in the array someColor, but you won't be able to use the array here) Each color should be equally likely.

7) Add code in myFirstMethod

- Have each penguin at the same time randomly paint itself (one of the four colors).
- Which loop do you use?

8) Write the **Scene function** named **howMany**

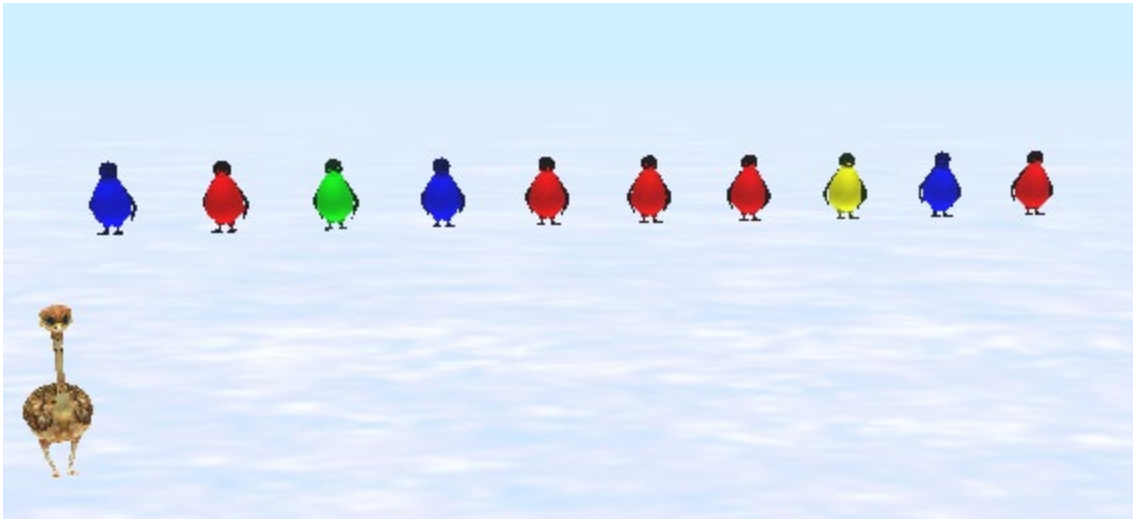
- This function **returns** a **WholeNumber**.
- This function has **one parameter**, a paint named **thisColor**
- This function calculates and returns how many penguins are the color thisColor.



- Hints on next slides

8) Scene function howMany (cont)

- For example, for this random coloring, howMany red would return 5, whereas howMany yellow would return 1.



- Hints next…

8) Scene function howMany (cont)

- You will need to loop through the penguin array and count how many there are of a specific color.
- You will need a variable that is initially 0 before the loop and is updated whenever you find a penguin of that color.

9) Add code in myFirstMethod

- Add code so for each color, you say how many penguins there are of that color.
- You will need to use a count loop with an index variable so you can get the color and also the string for the name of the color.
- Note you are looping over the someColor array and for each color calling the howMany function
- Hint: outline of code on next slide

Examples

- Hints follow…

9) Outline of code…

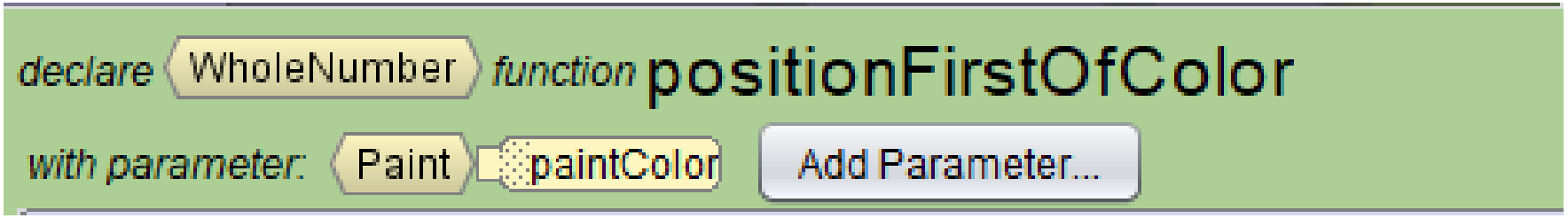
- *Set your index variable to 0*
- *Loop through color array*
  - *For each color, call the **howMany** function on that color (save this value in a variable)*
  - *Then have the ostrichBaby say "there are 3 penguins that are blue" (if there are 3 that are blue)*
  - *Update your index variable*

9) continued

- Did you consider the difference between how to say:
  - There is 1 penguin …
  - There are 3 penguins ….

- Hint: You will need an if for that!
- See pictures two slides back

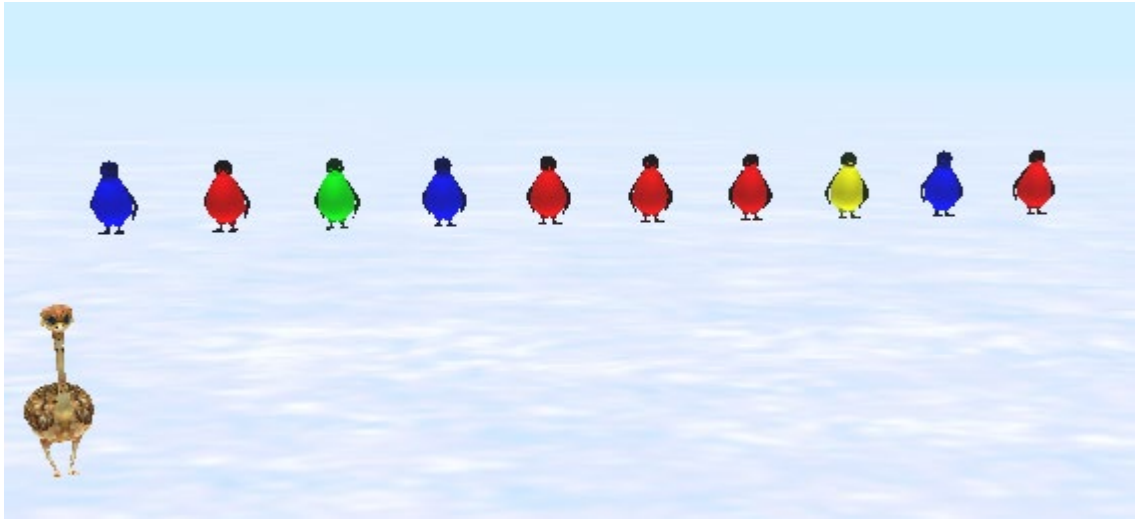
10) Write the **Scene** function named **positionFirstOfColor**



- This function has one parameter of type Paint named paintColor
- This function returns a wholeNumber
- This function returns the position of the first penguin of that color. If there is no penguin of that color then it returns -1.
- See examples and hints

10) Scene function positionFirstOfColor

- Here is an example. The first red penguin is the second penguin from the left (in position 1 since array positions start at 0). The first blue penguin is position 0, the first green penguin is the third penguin (at position 2). The first yellow penguin is position 7.
- Hints next



10) Scene function positionFirstOfColor

- You will need to loop through the penguins with an array index loop. That is, create a variable set to 0 initially and update it by 1 each time through the loop.
- As soon as you find the first penguin of that color, you are done, just return its index position.
- If you don’t find it after looking at all the penguins, return -1.

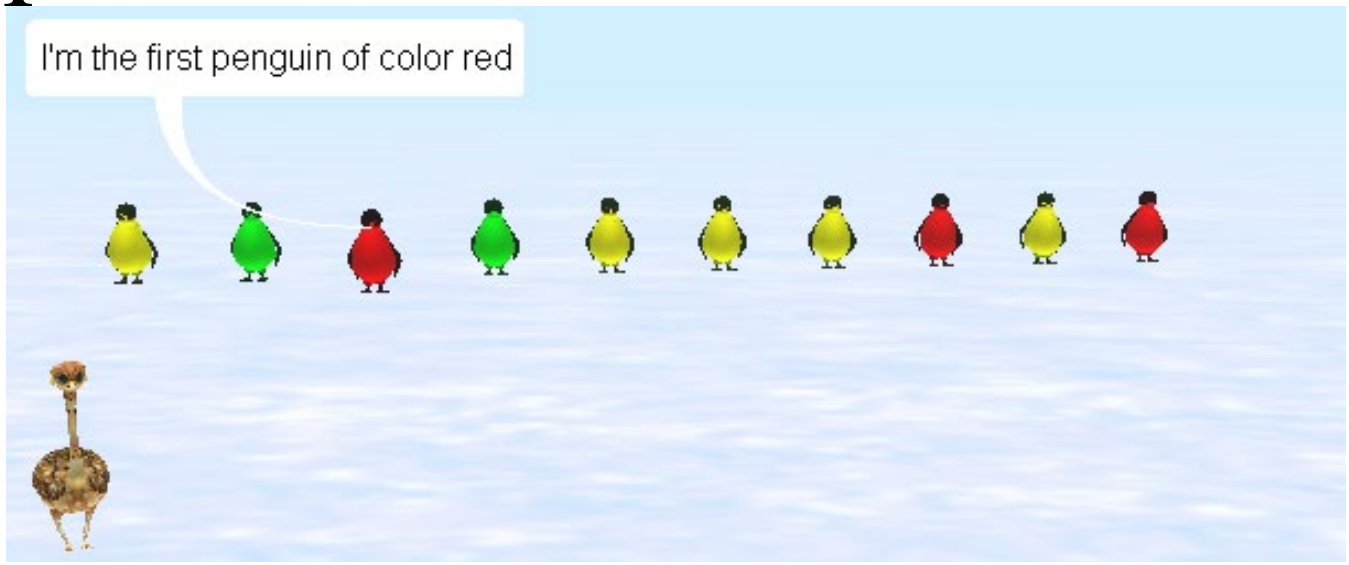
11) Add the last code to myFirstMethod

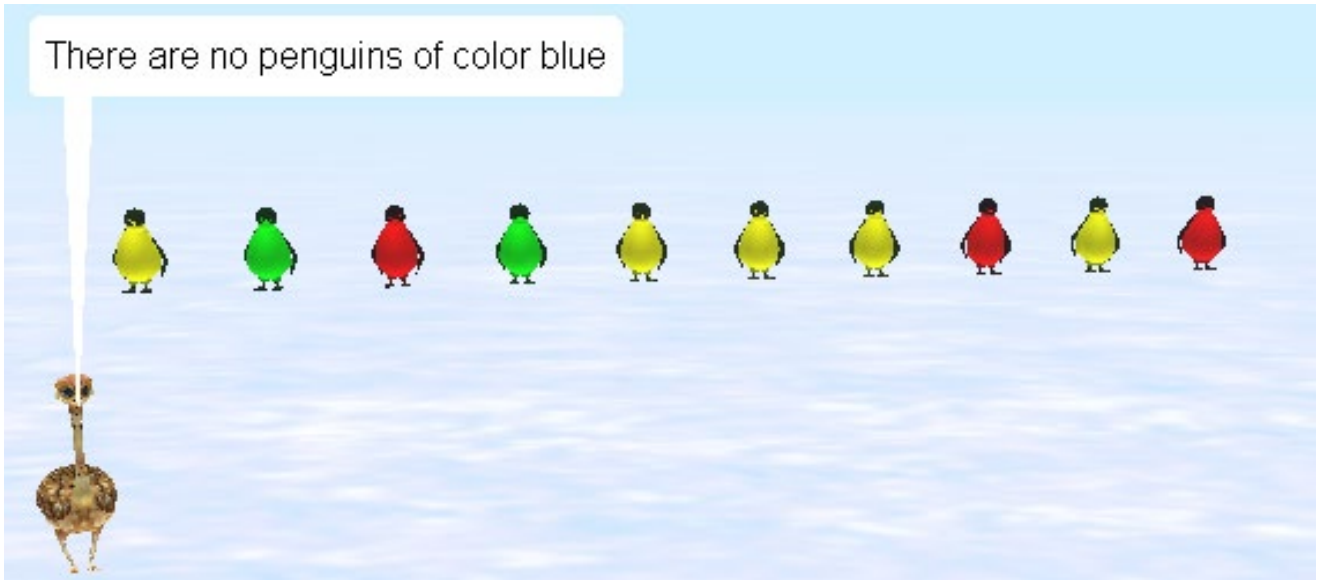
- Loop over the color array. For each color call positionFirstOfColor and save the index position returned in a variable.
- Then if the position is a valid position, have that penguin move forward, say they are the first penguin in the array that color (and say the color), then they move back.
- If the position is NOT a valid position, have them say there are no penguins of that color (say the color)
- Hints next…

11) Code in myFirstMethod continued

- *Set index to 0*
- *Loop through color array*
  - *Calculate positionFirstOfColor for current color and store in a variable*
  - *Check position and if valid*
    - *Move penguin in that position forward*
    - *Have penguin in that position say they are the first of that color (and say the color)*
    - *Move the penguin in that position back*
  - *If position not valid, say no penguins of that color*
  - *Update the index*

11) Examples


I'm the first penguin of color red


There are no penguins of color blue