Homework Exercise 11: Arrays And Elli

Goal: To learn about arrays and use them to build a seven segment worm, which we name Elli because it's made from ellipses.

Arrays – Variables with numbered elements

An array is a variable made up of multiple items that we refer by number. Because it's a variable it has a datatype. For example, suppose the x-coordinate for Elli's seven segments are:

0, 20, 40, 60, 80, 100, 120

then we can store these values in an array, elliX. We declare the array

```
int[ ] elliX = new int[7];   // declare an array for x-values of segments
```

This declaration is consistent with the other declarations we have already learned in class. The int[ ] is the *datatype*, which we say as *integer array* because of the brackets; elliX is the variable being named; the = means that the variable is being assigned an initial value, and new int[7] means the array will have seven elements that are integers. These will be numbered 0 through 6 – *notice the counting from 0 rule we always use* – as in

```
elliX[0] = 0;        // set last segment of elli
elliX[1] = 20;
elliX[2] = 40;
elliX[3] = 60;
elliX[4] = 80;
elliX[5] = 100;
elliX[6] = 120;      // set first (head) segment of elli
```

Notice that the square brackets are used in the *array references*, i.e. the elliX[0]. Square brackets are characteristic of arrays in almost all programming languages

Initial Elli

Create the initial version of Elli, by (1) creating a new dynamic Processing program – meaning it has setup( ) and draw( ) – and (2) create a new function called elli( ) in which to give the instructions for drawing the worm by (3) drawing the seven segments with 30 pixel diameters, aligned at 100 pixels down from the top and 15 pixels in from the side.

Things to attend to: Declare the array at the top of the program, make the canvas largish and squarish, lower the frameRate(8), or lower if needed, load the elements of the array elliX[ ] with the given x-coordinate values in the setup( ) using a loop, and call the elli( ) function in draw( ), its only instruction besides background(255).

To draw the final segment (stored in array element 0), which must be drawn first because of the way the circles overlap, write

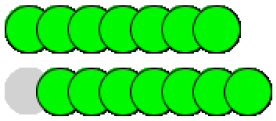
```
ellipse(15+elliX[0], 100, 30, 30);  // draw last segment
```

The other segments are similar, of course, so a loop might be best; if the loop variable is i, then you would reference the elements as elliX[i]. At this point, Elli displays, but that's all. Add eyes and other decoration as you wish.

Moving Elli

As the worm moves to the right, the segments' positions will shift left. That is, the x-coordinate of the i+1st segment, which is NOW in elliX[i+1], will become the x-coordinate in elliX[i] for the NEXT draw. The elliX[0] coordinate will be lost – replaced by elliX[1] – because the worm moves forward, and a new coordinate position must be created for the head, elliX[6]. Thus if the initial positions of the seven segments are

0, 20, 40, 60, 80, 100, 120



they become

20, 40, 60, 80, 100, 120, 140

to take the first move to the right.

So, to make Elli move, all we need to do is shift all of the array elements left one position, and compute a new value for elliX[6], which can be accomplished with

```
for(int i=0; i<6; i++) {
   elliX[i] = elliX[i+1];          //move each segment's x closer to the end
}
elliX[6] = elliX[6]+20;            //create a new x position for the head
```

Notice that this loop only advances six of the seven segments, because there is nothing to shift into head segment. So, after shifting all of the body, we create a new position for the head. Add this code after drawing Elli's body to be ready for the next draw. Watch the worm move across the screen.

The Same Idea for Y

So far, we have used an array, elliX[ ], to keep track of the x-coordinates of the worm. We can do this for the y-coordinates, too. Presently, the y-coordinate is always 100, but it can work just like the x-coordinate.

To keep track of the y-coordinate, declare another array, called elliY of seven elements, at the top of the program. Initialize each element to 100 in the setup( ); again, use a

loop. Advance each segment of the y-coordinate, as we just did for the x-coordinate. Create a new y position for the head (elliY[6]=100;) right after creating its new x-position. That's no more than 8 lines of code that exactly match what you've done with the x-coordinate.

Run the program … nothing has visibly changed, but now we have two arrays that fully describe the worm's location. So, after two "steps" the description of the worm is:

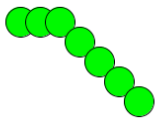
```
elliX[ ] =  60,  80, 100, 120, 140, 160, 180
elliY[ ] = 100, 100, 100, 100, 100, 100, 100
```

and we know that the ith segment will be an ellipse centered at elliX[i],elliY[i].

Going Up and Down

Now prove that Elli can move around flexibly by replacing the definition of the head to be

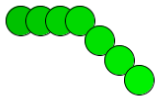
```
elliY[6]=elliY[6]+20;   // move Elli down screen
```



The worm should now travel diagonally down the screen.

Glam Worm

We have used an array to keep track of the x-coordinates of Elli, and another to keep track of the y-coordinates. Now, declare a third array to keep track of the color of each segment. In declaring this array, by analogy to the earlier definitions, remember that int is a datatype that elements can have, and so is color. Modify your program so that each segment has a different (but unchanging) color, and you can use any colors you wish.



Wrap-Up You have learned about arrays, variables composed of several elements that are referenced by numbers, called *indexes* or *indices*. The least index is always 0; the greatest index is the number of elements minus 1; that is, we said the arrays would have 7 elements but 6 was the largest legal index because of counting from 0. When we declare arrays, we say how many elements we want. Array elements are referenced one at a time by giving the number of the desired element in brackets.

Turn-In After completing all steps, including giving Elli some facial features, submit your .pde file to the class dropbox.