Name: ______________________________

Intro to 'Random' - Variables Part 4

J. Wilcox

So far our work with variables, both variables that were already a part of Processing as well as those that we created ourselves, have been focused on whole numbers (in other words – px was always equal to a counting number…1, 2, 3…34, 35, 36…). What if we wanted the computer to just give us a random value instead? Processing has a built in random number generator!

random (200);

If we enter just one argument (one value) in the parenthesis, that will be our maximum value. Processing will choose a number between 0 and 200 in this example.

random (-1, 3);

If we enter two arguments (two values), the first number will be the minimum and the second number will be the maximum. Processing will choose a number between -1 and 3 in this example.

One thing that is important for us to know is that Processing will choose a decimal value when using 'random'. For right now, when we name our variable, instead of 'int' we will use 'float'. One other change we will make is to set the variable's starting value in our first block of code instead of right away. So our code will begin looking like this –

```
float bluex;

void setup () {
        size (400, 400);
        bluex = random (400);
        }

void draw () {
        background (50);
        fill(0, 0, 255);
        ellipse (bluex, 100, 50, 50);
        }
```

Try running the program several times. What did you notice?

____

Does the circle's position change vertically, horizontally or both? ________________

How could you change the code to have the circle randomly show up anywhere on the screen (so that its x position and its y position will both change)?

____

____

We can also use the random function when having the ellipse move across the screen. Instead of constantly adding the same value, we can have Processing choose a value for us.

```
float bluex;

void setup () {
        size (600, 400);
        bluex = 0;
        }

void draw () {
        background (50);
        fill (0, 0, 255);
        ellipse (bluex, 200, 50, 50);
        bluex = bluex + random (-1, 3);
        }
```

The width of the canvas is being changed to 600.

Processing will continually choose a number between -1 and 3 to add to the ellipse's x value (the position that the ellipse is drawn on the screen from left to right).

Run that program a couple of times. How would you describe what you see?

____

____

____

We could use the random function to create a race! We'll need another variable as well as another circle. Let's add those new lines to our code…

```
float bluex;
float yellowx;
```

We need another variable. Let's call it yellowx

```
void setup () {
        size (600, 400);
        yellowx = 0;
        bluex = 0;
        }
```

yellowx will start at 0 also…we don't want to give one of the circles a head start!

```
void draw() {
        background (50);
        fill (0, 0, 255);
        ellipse (bluex, 100, 50, 50);
        fill (255, 255, 0);
        ellipse (yellowx, 300, 50, 50);
        bluex = bluex + random (-1, 3);
        yellowx = yellowx + random (-1, 3);
        }
```

Add another ellipse in our final block of code.

Run the program a few times. What happened?

____

____

____

Can you figure out how to include a third circle in the race? Hints: you will need another variable (maybe you could call it greenx), and green is *fill (0,255,0);*

What if you made the line for green x in draw – *greenx = greenx + random (6);* instead of *greenx = greenx + random (-1,3);*?

____

____

____

Another way to use ‘random’ would be to create a work of pixel art. We could have the computer draw circles randomly on the screen. We will need to make sure that the background command is put in the setup block so that our circles aren’t covered up over and over again.

Try this…

```
float px;
float py;

void setup () {
          size (400, 400);
          background (50);
          }

void draw () {
          px = random (width);
          py = random (height);
          ellipse (px, py, 25, 25);
}
```

We could add a line that would allow us to control when the circles start and stop. This could be done by adding an ‘if’ statement. If the mouse is pressed, we want the computer to draw circles. If the mouse is not pressed, we want the computer to stop. Add the line – if (mousePressed) just after the void draw line. That block of code will now look like this (notice the extra set of braces {} that are needed):

```
void draw () {
if (mousePressed) {
   px = random (width);
    py = random (height);
   ellipse (px, py, 25, 25);
}
}
```

Can you figure out how to have the color of each circle that is created change? You could start with just using gray scale (only one argument for fill…)

Other thoughts – at the beginning (one circle), bring in idea of multiple circles showing up randomly maybe include random sizes…different shapes?

Have two circles that start in upper left corner and move randomly down towards the bottom right corner

A circle that travels from the left side of the screen to the right side of the screen at an even pace (it should not move randomly but instead travel the same amount each time). Have the circle grow and shrink randomly as it travels (making sure that the shape still remains a circle the whole time…just that it is a smaller or larger circle).