Homework 7: Creativity and Processing

Goal: The purpose of this exercise is to give you a chance to create, and to get experience writing simple Processing Programs to give practice with the language features.

In the assignment you will work in group of two using pair programming and write four active Processing programs to display something of interest to you. These are not to be complex programs, but rather simple programs that do something cute or clever or interesting or have some other property that would interest a viewer. Below are four examples; notice that these are short programs that do something worth watching for a moment. That’s what you’re supposed to do.

Examples
Here are four simple Processing programs like the kind expected in this assignment. You can use anything from these programs, but keep in mind that small changes from one of these programs is NOT creative … it’s derivative! Grading is based on creativity, good use of the language and your comments.

Random Lines
The random number generator, “random(low, high);”, generates a float whose value is between “low” and “high”. Here is a program that draws batches of 10 random lines of either white or red, chosen at random. Notice that I can use the random function directly as an argument in the line( ) function.

```
/* Valentines Lines */
int i;

void setup() {
  size(600, 500);  // set canvas
  background(0);    // pick black for the
  translate(5);     // more fun to go slow
  strokeWeight(2);  // wide lines better
}

void draw() {
  for (i = 0; i < 10; i++) {  // Do a batch of 10
    line(random(1, 499), random(1, 499), random(1, 499), random(1, 499));
  }
}
```

Seattle Rain
Randomization is used in many places in computing ... for example, to simulate the physical phenomena. Let’s “simulate” the Seattle rain. This program generates droplets of a random size and one of two different (but very similar) random colors. The slow frame rate makes the action develop in kind of a hypnotic way, like Seattle rain.

```cpp
int i;
float x;

void setup() {
  size(500, 500);
  background(0);
  frameRate(5);
  fill(0);
}

void draw() {
  if (random(0, 1) < 0.5) { // dark or light drop?
    stroke(180,100,255); // dark this time
  } else {
    stroke(200,203, 255); // it's lighter
  }
  for (i = 0; i < 10; i++) {
    x = random(10, 20); // how big should drop be
    ellipse(random(1,499), random(1, 499), x, x); // rain!
  }
}
```

Programming A Heart Beat
Processing comes with various ways to make shapes, and so it is very flexible. Here a heart has been drawn with a few basic shapes, and colored red. The program then turns down the red some, and then turns it back up again, making it look like it is beating.

```java
/* The Heart Beat */
float dir=-1; // direction of change
float tinto=255; // start out red

void setup(){
  size(500, 500); // set canvas
  background(50); // pick gray for the app
  frameRate(15); // more fun to go slow
  noStroke(); // show not lines
  fill(255,0,0); // start out pure red
}

void draw(){
  ellipse(200,250,100, 100); //piece together
  ellipse(300,250,100, 100); //a heart shape
  triangle([153, 270, 348, 270, 250, 400]); //from basic shapes
  rect(240, 250, 20, 20); //
  tinto=tinto+dir*5; //set a new color
  if (tinto < 100 ) {
    dir=1; //is it too dark?
    //yes, start to lighten
  }
  if (tinto > 255) {
    dir=-1; //Is it full color
    //start to darken
  }
  fill(tinto,0,0); //set the color
}
```

Walking Man
In this program a simple stick figure is animated to walk forward ... will he fall into the abyss? Run it to find out. Notice that the body can be moved simply using our standard move-to-the-right techniques, but the legs must work differently. It’s not complex, but it gives the idea of a stick man walking.

```java
int walk = 0; // Which leg steps
int x = 0; // main body motion
int step1 = 0; // left leg step
int stepr = 0; // right leg step

void setup() {
    size(300, 300);
    frameRate(30);
    smooth();
    noFill();
    strokeWeight(3); // ticker line
}

void draw() {
    background(0); // erase previous
    stroke(255, 255, 0); // land is yellow
    line(50, 170, 300, 170); // land left of abyss
    line(300, 170, 50, 300); // left side of abyss
    line(300, 170, -50, 170); // right side of abyss
    line(170, 300, 170, 300); // right of abyss
    stroke(255, 0, 0); // stick man is read
    ellipse(100 + x, 100, 20, 20); // head
    line(100 + x, 110, 100 + x, 140); // body
    line(100 + x, 125, 90 + x, 110); // left arm
    line(100 + x, 125, 120 + x, 110); // right arm
    if (walk == 0) {
        step1 = step1 + 2; // which leg is moving
        if (step1 < 100) {
            step1 = 0;
        } else {
            step1 = step1 + 2; // left, go faster
        }
    } else {
        stepr = stepr + 2; // right, go faster
    }
    line(100 + x, 140, 90 + step1, 170); // actually move left
    line(100 + x, 140, 110 + stepr, 170); // actually move right
    if (abs(step1 - stepr) >= 15) {
        // stride over with?
        walk = 1 - walk;
    }
    x = x + 1; // move rest of body
}
```

**Assignment.** Write four programs to do whatever you want (but don’t copy the examples above), and try to make them clever or interesting or cute or have some property that would interest a viewer. You should try to use those you have learned in your former Processing homework, because those are the basics and one goal of this assignment is to practice the basics. But, if you need some other feature of processing that you find in the reference page, go ahead and use it. The goal is creativity ... but don’t spend forever on it either.

**Wrap up.** You’ve used programming to invent new visual applications.

**Turn In.** Please submit your four files (with the names of both team members at the top in a comment of each file) to the eCommons site through the “Assignments” link on the left. Each group should only submit one copy of programs. One partner in each group should submit the .pde files, the other partner should submit a text document with the names of BOTH partners in it. You are expected to complete this assignment together and both should be present when the submission is made to ensure it is done correctly.