Homework 8 & 9: Creativity and Processing

Goal: We have introduced Processing, discussed key features like functions and parameters, and shown how to control action on the screen. 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 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.

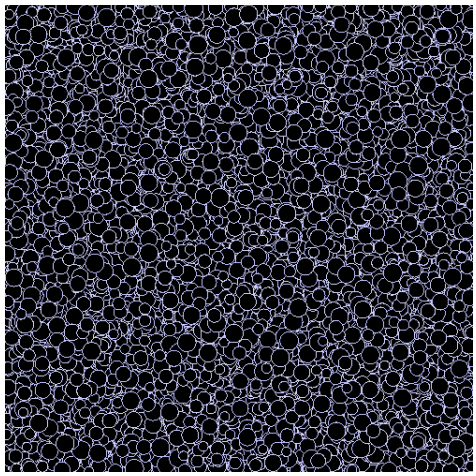
Examples

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**. All programs are available on the calendar page.

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.

```
/*  Its raining in Seattle                                    */
int i;                              // iteration variable
float x;                            // size of droplet

void setup( ){
  size(500, 500);                   // set canvas
  background(0);                    // pick black for the app
  frameRate(5);                     // more fun to go slow
  fill(0);
}

void draw( ) {
  if (random(0,1) < 0.5) {          // dark or light drop?
    stroke(180,180,255);            // dark this time
  } else {
    stroke(200,200, 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

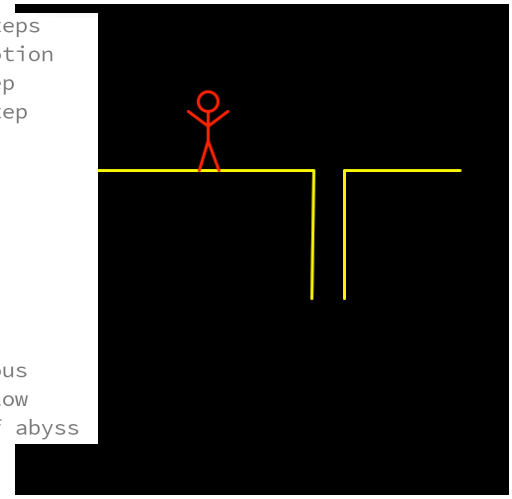
Processing comes with various ways to make shapes, and so it is very flexible. Here a heart has been drawn with a few basic shapes, and colored red. The program then turns down the red some, and then turns it back up again, making it look like it is beating.

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

void setup( ){
  size(500, 500);                       // set canvas
  background(64);                       // 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 < 180 ) {                     //is it too dark?
    dir=1;                                //yes, start to lighten
  }
  if (tinto > 255) {                      //is it full color
    dir=-1;                               //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.

```
int walk=0;                             // Which leg steps
int x=0;                                // main body motion
int stepl = 0;                          // left leg step
int stepr = 0;                          // right leg step
void setup( ){
  size(500, 500);
  frameRate(30);
  smooth();
  noFill();
  strokeWeight(3);                      // ticker line
}
void draw( ){
  background(0);                        // erase previous
  stroke(255,255,0);                    // land is yellow
  line(50, 170, 302, 170);              // land left of abyss
```

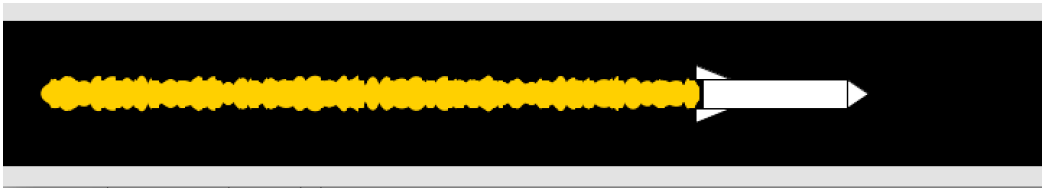


```
  line(302, 170, 300, 300);            // left side of abyss
  line(333, 170, 450, 170);            // right side of abyss
  line(333, 170, 333, 300);            // right of abyss
  stroke(255,0,0);                     // stick man is read
  ellipse(100+x, 100, 20, 20);         // head
  line(100+x, 110, 100+x, 140);        // body
  line(100+x, 125, 80+x, 110);         // left arm
  line(100+x, 125, 120+x, 110);        // right arm
  if (walk == 0) {                     // which leg is moving
    stepl = stepl + 2;                 // left, go faster
  } else {
    stepr = stepr + 2;                 // right, go faster
  }
  line(100+x, 140, 90+stepl, 170);     // actually move left
  line(100+x, 140, 110+stepr, 170);    // actually move right
  if (abs(stepl-stepr) >= 15) {        // stride over with?
    walk = 1-walk;                     // yes, switch legs
  }
  x=x+1;                               // move rest of body
}
```

Rocket

Notice how the programmer made the rocket move but the exhaust remains on the screen.

```
void draw( ) {
  smoke( );
  vehicle(color(0));
  x = x+1;
  vehicle(color(255));
}
void smoke( ) {
  noStroke( );
  float d;
  fill(255, 200, 0);
  ellipse((x+40)-(x%10), 50, max(10, random(20)), max(15, random(25)));
}
void vehicle(color c) {
  stroke(0);
  fill(c);
  rect(40+x, 40, 100, 20);
  triangle(140+x, 40, 155+x, 50, 140+x, 60);
  triangle(35+x, 40, 35+x, 30, 60+x, 40);
  triangle(35+x, 60, 35+x, 70, 60+x, 60);
}
```



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 for a few moments (or a few hours!). Remember, you know how to control action on the screen using the mouse. You should try to use the topics that are covered on or before Lecture 8, 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 – and understand it – 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. Turn in two programs at the first due date, turn in the last two programs on the second due date.