

**CPE 123: Introduction to Computing – Computational Art –
Fall 2016
Program 1 – Animated and Remixed Blexbolex**

Due date: Friday, 10/7 11:59pm

Modality:

- This is a pair assignment. Working from your own processing sketch for Lab 2 (Blexbolex), you will be partnered with another student in order to combine your two sketches and create a unique animation. You may talk to other students and your instructor to see what they are creating and how they did it. As always, with the exception of your partner, you are not allowed to ‘cut and paste’ code from another student or a tutorial in order to complete this assignment, nor can any segment of your code (which generates a visual component) exactly match another student’s code for this assignment (with the exception of your partner).

Objectives:

- Practice using p5.js
- Practice using simple p5.js primitives and shapes
- Practice using functions to encapsulate parts of your code for re-use
- Practice integrating two student’s code
- Practice scaling and translating shapes into a single coordinate frame
- Practice animating via transforms and variables
- Practice developing your aesthetic
- Make an interesting ‘story’ combining two existing designs and adding animation

Requirements:

For this assignment you must generate a combined scene from two different Blexbolex designs that incorporate animation of the main characters. Your final sketch must be either twice the width or height of the original two Blexbolex sketch reproductions.

Your final combined sketch must have all the original elements placed into a new unified sketch that are either side by side or one of top of the other (please select the option that makes the most sense).

Your final sketch must use functions and animation of the two main characters.

Step 1: Start by modifying your own individual Blexbolex sketch to use functions. In addition, to the **setup** and **draw** functions we have already been working with, you will need to write your own function to draw the element that will be animated. I also highly recommend including a method to draw any background elements in your sketch.

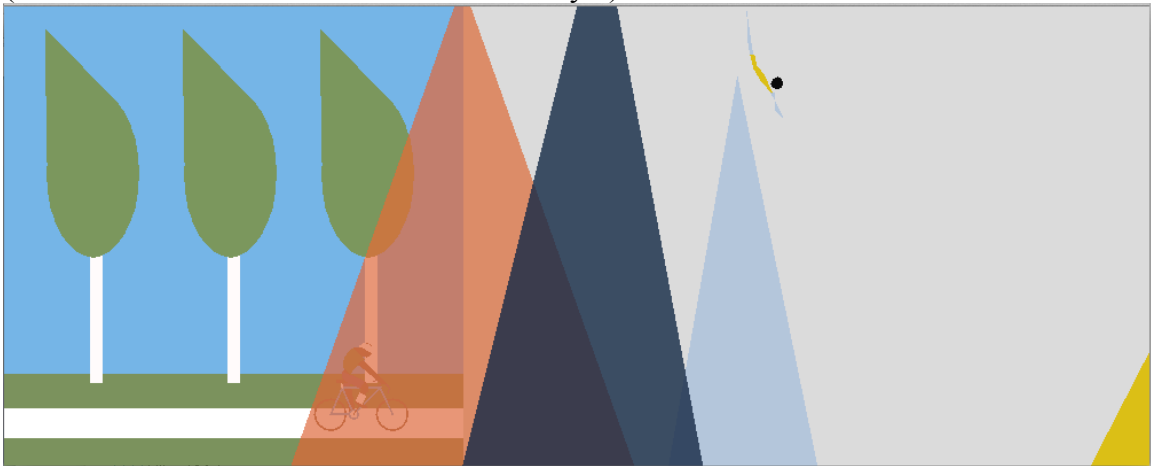
Step 2: Combine your sketch with your partners. Be sure to modify the size of your canvas to accommodate both sketch’s elements. Decide on their placement and offset one of the other’s background and main characters to have them appear side by side. See Figure 1 for an example.

Step 3: Add the necessary variables to make both main characters animate. At this time I only expect very simple animation. (ie translation via incrementing a variable).

You and your partner are required to handin a final image and your final sketch code using polylearn. **Be sure that your code has your name and your partner's name as a 'comment'.** You and your partner will also be demoing your solution in lecture or lab.



Figure 1: A frame near the start of the animation, showing the combined Blexbolex sketches (note the diver hasn't entered the animation yet).



A later frame with the biker behind the mountains and the diver starting her dive.



Another later frame showing both characters continuing their journeys.

Grading rubric:

25 points: Code combined to make a reasonable scene

25 points: one character animates in a reasonable way including variables to control transforms

25 points: use of functions to encapsulate portions of the drawing

25 points: Sketch is expressive, interesting and appealing

Some useful commands to look into:

strokeJoin, strokeWeight, translate, scale, rotate, push, pop

Here is a very simple example program using the necessary methods to animate very simply using transforms:

```
//a variable to control location
var tx = 0;

//function to set up the sketch size
function setup() {
  createCanvas(400, 400);
  smooth();
}

//function to draw my background
function drawSet(offsetX, offsetY) {
  fill(26, 29, 137);
  rect(offsetX, offsetY, 100, 200);
  fill(72, 75, 198);
  rect(offsetX+100, offsetY+50, 100, 150);
}

//function to draw my character
function drawChar(tx, ty) {
  push();
  translate(tx, ty);
  fill(33, 137, 26);
  ellipse(0, 0, 40, 40);
  fill(110, 26, 137);
  ellipse(-5, -5, 5, 5);
  ellipse(5, -5, 5, 5);
  pop();
}

//function to draw all elements of my sketch
function draw() {
  background(254);
  drawSet(100, 100);
  drawChar(tx, 270);
  tx = tx + 1;
}
```