1 Program 3- Art Nouveau curves and generative elements

Goals

The goals for this program (homework) are:

1. Practice using a `loop` control structure to *generate* patterns/scenes
2. Practice using functions to re-draw parts of a scene
3. Practice using `random` to produce desirable colors and design layout
4. Practice using structured layout
5. Practive using parametric curves to control curved patterns in a sketch
6. Apply repetition of drawing certain elements in a sketch

Modality and due date

Pairs (to be assigned by the instructor) - DUE Friday Oct. 28 at 11:59pm

Overview

Art Nouveau: Art nouveau is a style inspired by natural forms with a focus on curved lines. "This describes a decorative style popular from the last decade of the 19th century to the beginning of the First World War. It was characterised by an elaborate ornamental style based on asymmetrical lines, frequently depicting flowers, leaves or tendrils, or in the flowing hair of a female. " [1]

[1] http://www.artmovements.co.uk/artnouveau.htm

Taking our inspiration from this art movement, we will focus on curved lines in particular ‘parametric curves’ that can be used to control plotting out circular and curved forms with a focus on spirals. In addition, this lab is inspired by the use of repetitive organic shapes (for example the rich tiled elements in many of Gustov Klimt’s pieces).

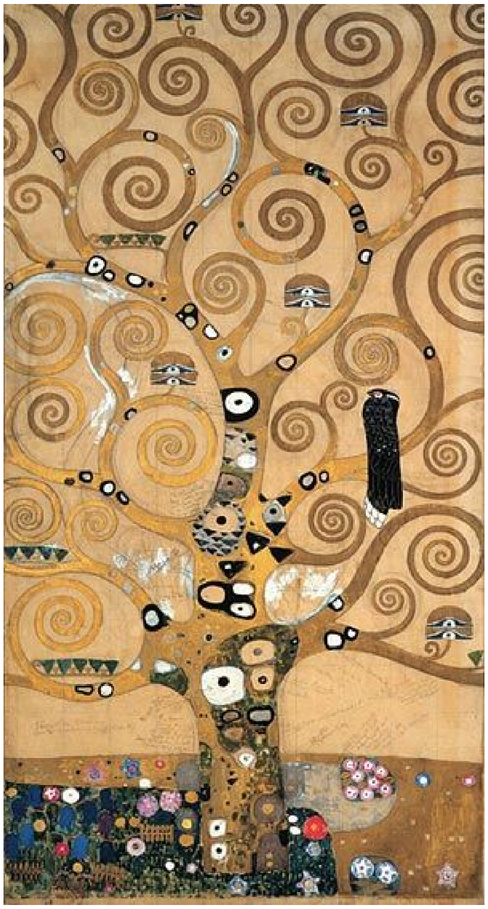

Figure 1: Example of an art nouveau image from Gustov Klimt ‘Tree of Life’

Details

Task: You must create an art nouveau style sketch using p5.js which includes both curved elements and repeated ’tile’ like aspects. In addition, your sketch must animate such that either the curves are drawn over time or the tiles are drawn over time – that is, the curves ‘grow’ over time or the tiles are laid down until the sketch is complete (do not worry if you use random and the image seems to swim or vary per frame, as long as each individual frame looks decent). Your project must:

- include at least 4 copies of a stylized parametric curve. This curve must be generated via a loop (either plotting vertices of plotting shapes).

The curve must appear at least 4 times with different parameters. The curve must be drawn via a function (that possibly animates over time (drawing more and more of the curve over time)).

- include a tiled aspect (some design component that is repeated at least 10 times to add texture to your scene) (that possibly animates over time (drawing more and more of the tiles over time) – **either** the curves or the tiles (or both) must animate. The tiles must be visually complex (at least 4 different shapes/colors)
- be at least 400 x 400
- be in color
- use `random` appropriately to try to create a more organic look

To complete this program, you must:

- find a reference image (check the art nouveau pinterest board for examples - in general you are looking for an art nouveau piece with curves and repetition (Klimt is a good artist to consider)).
- design your own sketch which includes both curves and repeated elements
- write the code to create one curve (using a loop) and encapsulate that in a function that can be re-used in your sketch to draw the 4 different versions of that curve (you can draw more then 4). Your curve must be somewhat complex (more complex then a single circle or sin curve - please ask for approval if you are unsure).
- write code to also draw some repeated design element at least 10 times in your sketch (aka a tile with visual complexity at least 4 different shapes/colors)
- once you have the complete sketch working (statically) then add an animation component to either the curves, tiles or both, such that those elements appear to 'grow' (either changing length or number) over time. Do not worry if you use random and the image seems to swim or vary per frame as you animate, as long as each individual frame looks decent. We will learn about a way to prevent this later, but for now, it can just be an aspect of your animation.

Submitting your sketch and grading rubric

You and your partner are required to post your sketch to your webpages. Edit the html page that loads the script to include instructions on where to click. The html must also include reference art. In addition, you must handin your final script code and the associated html page (that loads the script). Be sure that your sketch code has your name and your partners name

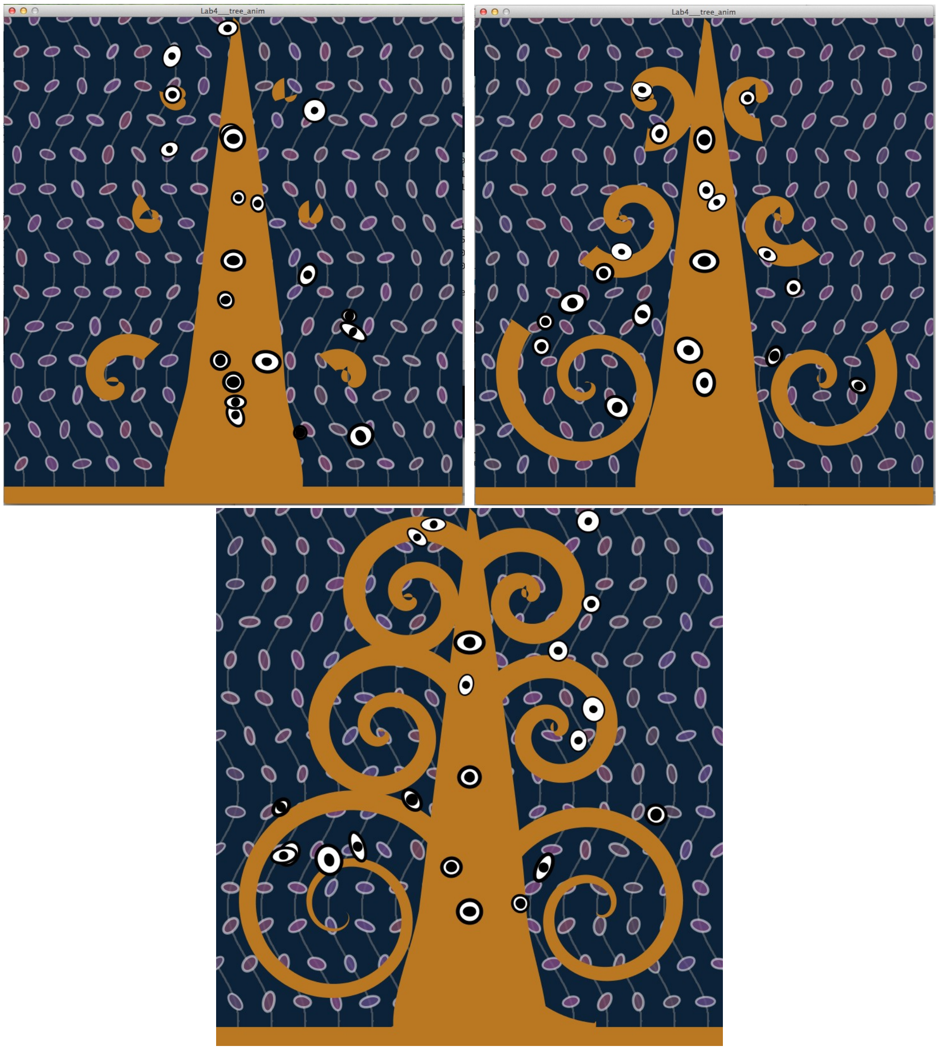

Figure 2: Three different frames from a p5.js sketches of an art nouveau style tree with multiple curved elements (spirals for branches) and sin curve vines in the background, in addition to repeated oval leaves along the vines and ‘eyes’ in the tree. Notice that the position of the leaves on the vines and the eyes vary in each frame, this is due to using random and is acceptable for this program.

as a comment. You and your partner will also be demoing your solution in lecture or lab. Your score will be broken down 75% for meeting the technical requirements and 25% for aesthetics. Each pair will be required to meet with the instructor to explain what they contributed to the sketch. The expectation is that you both worked equally hard and both understand all aspects of the sketch.

Rubric for Grading: Your sketch will be graded based on:

- 25 points: four stylized interesting parametric curves created via loops drawn via a function
- 25 points: tiled background elements (visually complex 4 shapes or colors)
- 25 points: Either tiles or curves (or both) are animated over time and random() is used appropriately to vary sketch elements
- 25 points: Sketches are interesting looking art nouveau scene

Note to receive any points, your sketch must be on your webpage with html including reference art

Resources:

```
/*ZJ Wood - example CPE 123 code for animated parametric curve of a spiral */
var stopHere;

function setup() {
  createCanvas(400, 400);
  stopHere = 0;
}

function drawSpiral(cx, cy, radi) {
  var sx, sy, sradi;
  stroke(255);
  fill(0, 125, 240);
  sradi = 0;
  for (var i=0; i < stopHere; i++) {
    sx = cx + sradi*sin(radians(i));
    sy = cy + sradi*cos(radians(i));
    ellipse(sx, sy, sradi/5, sradi/5);
    sradi = sradi + radi/720;
  }

  //advance the loop parameter so each frame it animates more
  //as long as we haven’t reach 720 (visually pleasing)
  if ( stopHere < 720) {
   stopHere = stopHere + 1;
  }
}

function draw() {
  background(12, 34, 56);
  drawSpiral(width/2, height/2, 80);
  drawSpiral(100, 100, 60);
}
```