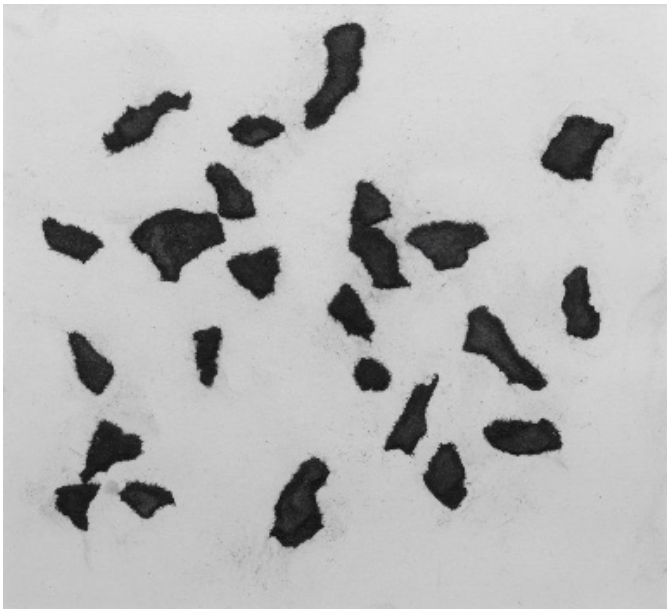
Pitch.Concept

LYS QUINTERO
CART 253
PROFESSOR : PIPPIN BARR



The main Idea of my final project it's a series of interactive graphic images or interactive art. A series of images generated by code that can change with the position or click of the mouse.

This series is reminiscent of the Dadaist work of artist Jean Arp. Specially his series of "laws of chance", because he is using the randomness and position to create his art: "Arp felt that he could incorporate chance within artistic production, comparing the role of the artist to a plant bearing fruit. According to the Laws of Chance shows Arp playing with random composition, in this case dropping painted pieces of paper onto a surface. Torn Woodcut was made in a similar way in 1954, using the pieces of a Dada print he had made in 1920 "(Gallery label, April 2008).

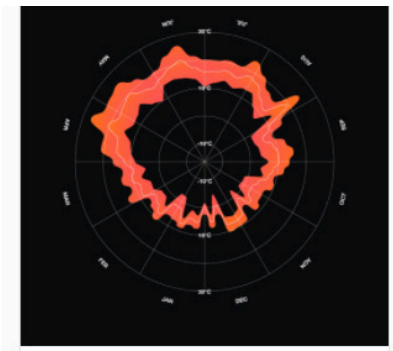
NOVEMBER 6 ,2017

Pitch.Concept

The goal is that the user can create unexpected graphics (random) that can be a tool to create something physical afterwards. It can also be therapeutic, thanks to the benefits of color therapy and also a way to find inspiration in the unexpected. The abstract forms that make up the composition can also thave some behavior. Like bringing the art into another level, a level of autonomy of the form. As if the form was a being or had its own consciousness ...

As I really like Arp's aesthetic of abstraction, I decided to work with organic forms similar to those found in the works of Jean Arp. The forms are going to be kinds of: "blobs". A blob in processing, can be seen to turn a simple circle into a more "liquid" form, the corners start to deform and so we get an abstract and organic object. According to Daniel Shiffman: A technique using beginShape () and endShape () along with perlin noise is used.

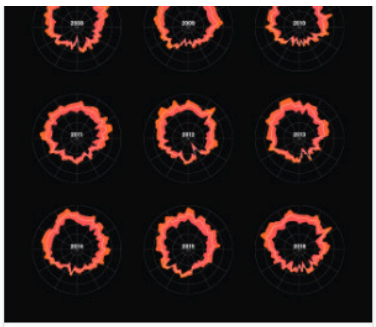
Jean Arp



Adding more things so people can actually understand the graphic. (All temperatures of 2016 in the Netherlands)

#pttrnz #graph #datavisuali #weather #weatherdata #data #computer #processing #p5 #illustrator #improving #experiment #proces #creativecode #code #computerart

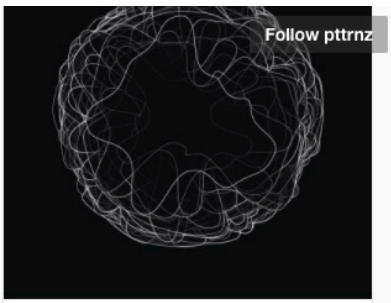
51 notes



More weather beziers fueled with temperature data of the last 9 years in the Netherlands.

#weather #bezier #graphic #design #generative design #generative art #art #datavisualization #visual #proces #process #experiment

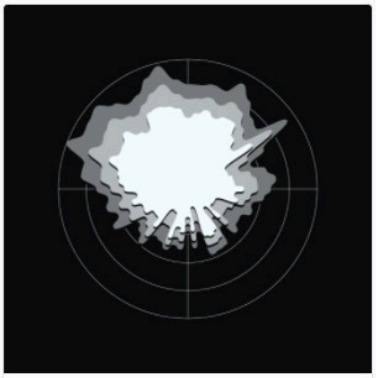
17 notes



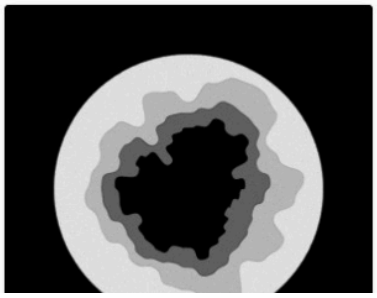
sketch-15102017-14005.png

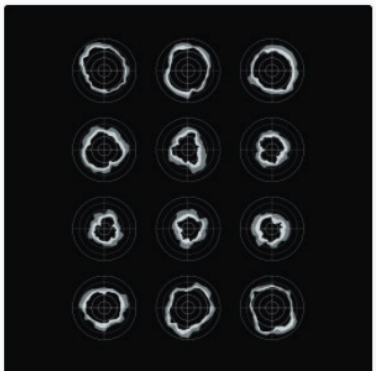
#generative #desgin #processing #computer art #art #p5 #bezier #scribbles

21 notes



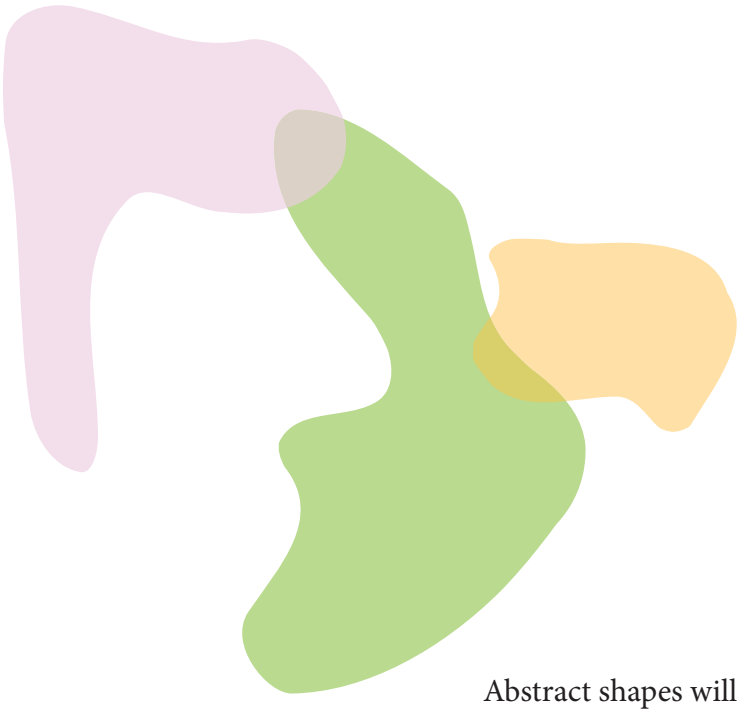
All temperatures of 2016. The smallest shape represents the minimum temperatures and the largerst shape the max





http://pttrnz.tumblr.com/

Media



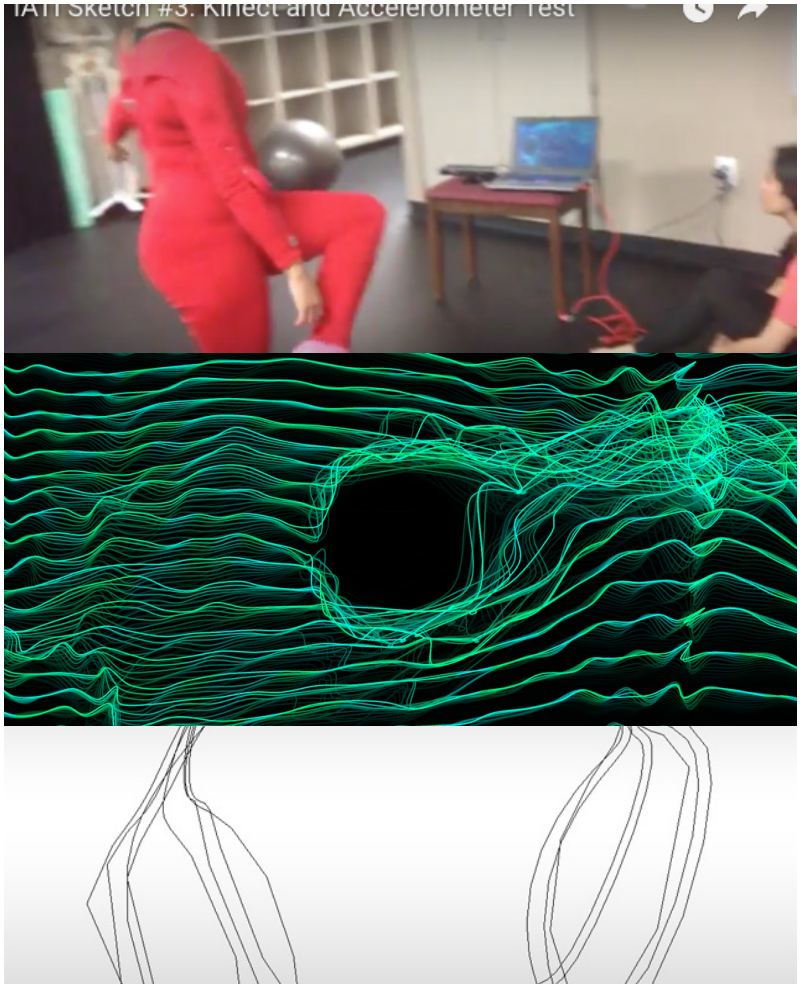
Abstract shapes will move with the interaction of the user. Then the user can change and play with the compositions with certain shapes. And also play with random placement thanks to the random () function

Interactivity inspirations

Antoinette Bumatay :
SUBMERGED: PROCESSING SKETCH EVOLUTIO

- Moving the shapes with the movement detection (Kinect)

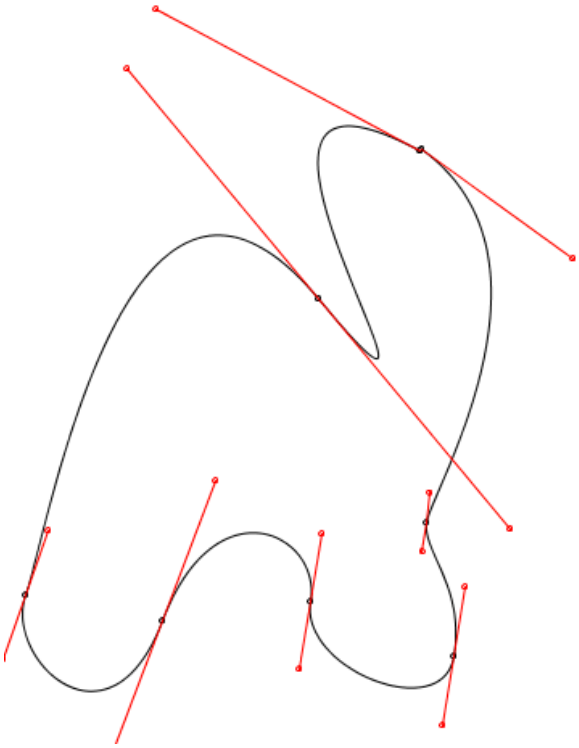
- For my project I'm thinking of doing something more simple , like detection of color for the movement.



Technical approach

Programming concepts/techniques:

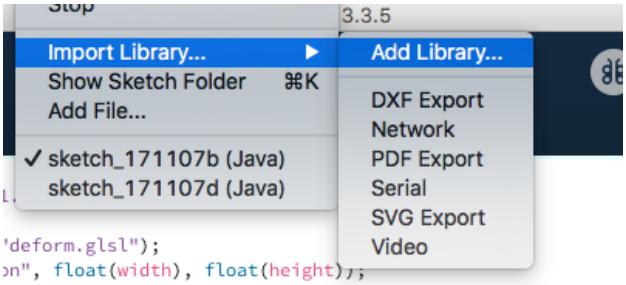
noise ()
random()t
class / objects
collisions
PShape
Bezier
Arrays
Loops



For the technical approach, I will use some notions that we saw in class I think I use the noise () function. I'm going to make classes with the abstract forms that I'm going to create with the vectors (either the Bezier on processing, or just import a shape with PShape ().) I'm going to animate the movement of its shapes with and use collisions. shapes will follow the mouse or a color in the image of the camera.

Technical research

Libraries :
- Video Library
- Loom 1.0.0 Cora Johnson-Roberson
Patterns that change over time, flexibly mapped to audiovisual output.
-Free Transform 0.4 Bartosh Polonski
Transform textures interactively




Loom

```
/**
 * SoundSample uses the Sound library to play a sample according to a pattern.
 *
 * Be sure you have installed the Sound library from the Processing
 * Contribution Manager (Sketch -> Import Library -> Add Library).
 */

import com.corajr.loom.*;
import processing.sound.*;
SoundFile snare;

Loom loom;
Pattern pattern;

void setup() {
  size(400,400);

  loom = new Loom(this);
  pattern = new Pattern(loom);

  SoundFile snare = new SoundFile(this, "snare.aif");
  pattern.extend("010101");
  pattern.asColor(#000000, #FFFFFF);
  pattern.asSoundFile(snare);

  pattern.loop();

  loom.play();
}

void draw() {
  background(pattern.asColor());
}
```

Existing code for Bezier :

```
ArrayList<PVector> anchorPoints; //store the anchor points
ArrayList<PVector> controlPoints; //store the control points
boolean DRAWING; // Am I drawing?
int pointCounter; //how many anchor points are there so far?
int selectedControlPoint; //What control point, if any, is the mouse on? If none, give me -1, otherwise the index of the control point in the arraylist
int selectedAnchorPoint; //What anchor point, if any, is the mouse on? If none, give me -1, otherwise the index of the anchor point in the arraylist
boolean controlLocked; //Lock the controlpoints preceding and following each anchor point together?

void setup() {
  size(800, 800);
  smooth();
  anchorPoints=new ArrayList<PVector>();
  pointCounter=0;
  controlPoints=new ArrayList<PVector>();
  DRAWING=false;
  selectedControlPoint=-1;
  selectedAnchorPoint=-1;
  controlLocked=true;
  textAlign(CENTER);
  textSize(12);
}

void draw() {
  background(255);
  stroke(0);
  noFill();
  //draw the current Bezier curve, see Processing reference on bezierVertex();
  if (anchorPoints.size()>0) {
    beginShape();
    vertex(anchorPoints.get(0).x, anchorPoints.get(0).y);
    for (int i=0; i<anchorPoints.size ()-1; i++) {
      bezierVertex(controlPoints.get(2*i).x, controlPoints.get(2*i).y, controlPoints.get(2*i+1).x, controlPoints.get(2*i+1).y, anchorPoints.get(i+1).x, anchorPoints.get(i+1).y);
    }
    endShape();
  }

  //draw the points
  for (PVector p : anchorPoints) {
    if (mouseOver(p,4)) {//mouse within 4 pixels of point?
      ellipse(p.x, p.y, 6, 6);
    } else {
      ellipse(p.x, p.y, 3, 3);
    }
  }
  stroke(255, 0, 0);
  for (PVector p : controlPoints) {
    if (mouseOver(p,4)) {//mouse within 4 pixels of point?
      ellipse(p.x, p.y, 6, 6);
    } else {
      ellipse(p.x, p.y, 3, 3);
    }
  }

  //draw the connections between control points and anchor points
  for (int i=0;i<controlPoints.size();i++) {
    //find the associated anchor point: control point 0-> anchor 0, control points 1 and 2 -> anchor point 1, etc
    // integer division: 1/2=0, 2/2=1,3/2=1, etc
    PVector anchor=anchorPoints.get((i+1)/2);
    line(anchor.x, anchor.y,controlPoints.get(i).x,controlPoints.get(i).y);
  }
  HUD();

}

void HUD(){
  pushStyle();
  fill(0);
  noStroke();
  text("Left-click to start adding anchor points. Existing curve will be cleared", width/2,height-100);
  text("Right-click to stop adding anchor points.", width/2,height-80);
  text("Left-click and drag anchor or control point to move.", width/2,height-60);
  text("Press spacebar to toggle lock mode. If enabled, control points move together.", width/2,height-40);
  popStyle();
}

void addControlPoints(){

  PVector currentAnchorPoint=anchorPoints.get(pointCounter);
  PVector previousAnchorPoint=anchorPoints.get(pointCounter-1);
  if(!controlLocked || pointCounter<2){
     //if not locked together, create two new control points along new segment
     PVector control1st =PVector.lerp(previousAnchorPoint, currentAnchorPoint, 1.0/3.0);
     PVector control2nd =PVector.lerp(previousAnchorPoint, currentAnchorPoint, 2.0/3.0);
     controlPoints.add(control1st);
     controlPoints.add(control2nd);
  }else{
     //if locked together, vector from previous control point to anchor point is same as vector from anchor point to next control point
    PVector previousControlPoint=controlPoints.get(controlPoints.size()-1);
    PVector fromPreviousControlToAnchor=PVector.sub(previousAnchorPoint,previousControlPoint);
    PVector control1st =PVector.add(previousAnchorPoint, fromPreviousControlToAnchor);
    controlPoints.add(control1st);
    PVector control2nd =PVector.lerp(previousAnchorPoint, currentAnchorPoint, 2.0/3.0);
    controlPoints.add(control2nd);
  }
}



//INTERFACE

void keyPressed(){
  if(key==' '){
 controlLocked=!controlLocked;
  }
}
```

```
void mousePressed() {
  if (mouseButton == LEFT) {
    leftMouse();
  } else if (mouseButton == RIGHT) {
    rightMouse();
  }
}

void leftMouse() {
  //Did I click on a control point? If yes, do nothing and let mouseDragged() handle it.
  selectedControlPoint=checkMouseOverControl(16.0);
  //Or did I click on an anchor point? If yes, do nothing and let mouseDragged() handle it.
  selectedAnchorPoint=checkMouseOverAnchor(16.0);

  //I guess not, add a new anchor point, and two new control points
  if (selectedControlPoint==-1 && selectedAnchorPoint==-1) {
    if (DRAWING==false) {
      println("Starting curve.");
      DRAWING=true;
      controlPoints.clear();
      anchorPoints.clear();
      println("Adding first anchor point.");
      anchorPoints.add(new PVector(mouseX, mouseY));
      selectedAnchorPoint=0;
      pointCounter=1;
    } else {
      println("Adding next anchor point.");
      anchorPoints.add(new PVector(mouseX, mouseY));
      addControlPoints();
      selectedAnchorPoint=pointCounter;
      pointCounter++;
    }
  }
}

void rightMouse() {
  if (DRAWING==true) {
    println("Ending curve.");
    DRAWING=false ;
  }
}

void mouseDragged()
{
  //if the mousclick that started drag was on a control point, move the control point
  if (selectedControlPoint>-1) {
    //move the control point;
    controlPoints.get(selectedControlPoint).set(mouseX, mouseY);
    //if the control points are locked move the associated control point
    if(controlLocked && selectedControlPoint>0 && selectedControlPoint<controlPoints.size()-1){
      //find the other control point, not possibel if first or last control point in curve
      int otherControlPoint=(selectedControlPoint%2==0)?selectedControlPoint-1:selectedControlPoint+1;
      //find the associated anchor point: control point 0-> anchor 0, control points 1 and 2 -> anchor point 1, etc
      int anchorPoint=(selectedControlPoint+1)/2;//integer division: 1/2=0, 2/2=1, 3/2=1, 4/2=2, etc...
       //if locked together, vector from previous control point to anchor point is same as vector from anchor point to next control point
      PVector fromSelectedControlToAnchor=PVector.sub(anchorPoints.get(anchorPoint),controlPoints.get(selectedControlPoint));
      controlPoints.get(otherControlPoint).set(PVector.add(anchorPoints.get(anchorPoint), fromSelectedControlToAnchor));
    }
  }else if(selectedAnchorPoint>-1){
    //If moving an anchor point, keep its control points at the same relative position

    PVector fromAnchorToPreviousControl;
    PVector fromAnchorToNextControl;
    int last=anchorPoints.size()-1;

    PVector anchor=anchorPoints.get(selectedAnchorPoint);
    PVector previousControl;
    PVector nextControl;

    //Handle first anchor point (only next control point)
    if(selectedAnchorPoint==0){
      //Check if any control points exist (not the case if only one anchor point has been drawn)
      if(controlPoints.size()>0){
        nextControl=controlPoints.get(0);
        fromAnchorToNextControl=PVector.sub(nextControl,anchor);
        anchor.set(mouseX, mouseY);
        nextControl.set(PVector.add(anchor, fromAnchorToNextControl));
      }else{
        anchor.set(mouseX, mouseY);
      }
    }
    //Handle last anchor point (only previous control point)
    else if(selectedAnchorPoint==last){
      previousControl=controlPoints.get(controlPoints.size()-1);
      fromAnchorToPreviousControl=PVector.sub(previousControl,anchor);
      anchor.set(mouseX, mouseY);
      previousControl.set(PVector.add(anchor, fromAnchorToPreviousControl));
    }
    //All other anchorpoints have two control points
    else{
      //A0-C0-C1-A1-C2-C3-A2-C4-C5-A3 : previous control = 2*anchor-1, next control = 2*anchor
      nextControl=controlPoints.get(2*selectedAnchorPoint);
      previousControl=controlPoints.get(2*selectedAnchorPoint-1);
      fromAnchorToPreviousControl=PVector.sub(previousControl,anchor);
      fromAnchorToNextControl=PVector.sub(nextControl,anchor);
      anchor.set(mouseX, mouseY);
      nextControl.set(PVector.add(anchor, fromAnchorToNextControl));
      previousControl.set(PVector.add(anchor, fromAnchorToPreviousControl));

    }
  }
}

//Is mouse pointer within a certain distance of a control point? Check all points and return the first one found.
int checkMouseOverControl(float distance) {
  float squareRadius=distance*distance;
  for (int i=0; i<controlPoints.size (); i++) {
    if (mouseOver(controlPoints.get(i), squareRadius)) {
      return i;
    }
  }
  //no control point on mouse
  return -1;
}

//Is mouse pointer within a certain distance of an anchor point? Check all points and return the first one found.
int checkMouseOverAnchor(float distance) {
  float squareRadius=distance*distance;
  for (int i=0; i<anchorPoints.size (); i++) {
    if (mouseOver(anchorPoints.get(i), squareRadius)) {
      return i;
    }
  }
  //no anchor point on mouse
  return -1;
}

//Is mouse pointer within a certain distance of PVector? The cutoff distance is given as a squared distance.
//Squared distances between points are cheap to calculate (no square root).
boolean mouseOver(PVector p, float squareRadius) {
  return ((mouseX-p.x)*(mouseX-p.x)+(mouseY-p.y)*(mouseY-p.y))<squareRadius;
}
```

Technical research : Links

Deformations
https://github.com/alecjacobson/geometry-processing-deformation

https://forum.processing.org/two/search?Search=deformation

Cell :
https://forum.processing.org/two/discussion/19756/deformation

GiFs and animations :
https://forum.processing.org/two/discussion/12737/gifanimation-for-processing-v3

https://stackoverflow.com/questions/22124039/exporting-a-gif-from-a-processing-sketch-w-gif-animation-library

PShape , importing vector , shape into processing
https://www.youtube.com/watch?v=O7smrxFF8Tg

Bezier Curve on processing:
https://www.youtube.com/watch?v=TPSoEMptZNA

http://www.wblut.com/tutorials/processing/bezier-curve-editor/

Inspiration interactive : Forms follows motion or color detection (camera) :

https://antoinettebumatay.com/2014/05/08/submerged-processing-sketch-evolution/

https://www.youtube.com/watch?v=rX5p-QRP6R4

In this short Coding Challenge, I demonstrate how to turn an circle (or any shape you want) into a blob and give the edges have a liquidy / blobby / wobbly look. A technique using beginShape() and endShape() along with perlin noise is used.

LOOM: https://vimeo.com/92316188