Twitter Poetry

The combination of semantics (concepts), computation (code), and an aesthetic domain (e.g. music) has been visited in the previous assignment and our discussions. This assignment will push you to consider a new concept that is appropriate to the realms of social media (Twitter) and poetry. Consider computational works by Bogost and Montfort to consider how the concept of poetry can be reinterpreted beyond the classical literary sense.

Your assignment is to create an interactive poem using content grabbed from Twitter. While you are not required to use any particular language, Processing will be encouraged and focused on in class. Your program should grab content from Twitter in real-time (and possibly other online sources) to create a data-driven, poetic expression based on tweets. For single technical requirements, you must create your own user-defined class and use it in your program. Make sure you have a clear concept, appropriate code solution, and final product that communicates your vision clearly.

Some suggestions:

- Consider using something like WordNet or RiTa to get syntactic & semantic information about words
- Consider using a context-free grammar for parsing / generating content
- Think about how this piece takes advantage of being on a computer vs. on the written page
- Consider the works referenced above as inspiration for thinking of poetry outside of the standard literary frame

For submission, you will upload your project to t-square:

1. an artist's statement in a PDF containing a thorough description of the final aesthetic and algorithmic choices made. Clearly describe the algorithms and data structures used and how they serve the aesthetic goals of the project.
2. a link to a 30-60s video of your work with voiceover, explaining the work to a general audience, on your online portfolio
3. your entire design journal
4. your code (.zip up the folder that contains your .pde and, if applicable, other files)

5. instructions for how to run the program in a separate PDF called "instructions.pdf". Include URLs for any libraries required to run your code.

Your project will be graded on:

- (10%) Clearness and conciseness (i.e. don't ramble) of artist's statement
- (15%) Aesthetic creativity
- (15%) Computational creativity
- (20%) Polished execution of design concept
- (20%) Meeting project requirements (e.g. submitting the design journal, using a user-defined class, etc.)
- (10%) Clear commenting for all code sections, user-defined functions, and algorithms (everywhere useful!)
- (10%) Appropriate use of conventions (e.g. descriptive variable names, user-defined functions and iteration where applicable, etc.)

Tuesday Oct 4

Semantics: the meaning of words

- The dog bits me

Syntax
The dog bit me

- noun(the dog) - verb(bit) - direct object (me)
- noun phrase (the dog) - verb phrase (bit me)

Computers aren't too bad at syntax.

Natural language is hard
"Time flies quickly"

- noun phrase (time) - very phrase (flies quickly)
- all very phrase ((you) time flies quickly)

Context free grammar:
A sentence can be broken down into many elements: a noun phrase + verb phrase

Recursive (Ask Albith)
S → NP + VP
NP → Determinant + NP OR noun phrase
VP → V OR V + Direct Object
DO → Determinant + N OR noun

Poetry Generation Program that Use Recursive
Rita: https://rednoise.org/rita/
Oracle: docs.oracle.com

random() → returns a float → turn into an integer to use as index number
use "get" instead of "=" because "=" might accidentally put stuff into variable instead of getting it.

Things to look up and consider
Create Poem Generator in Processing
loadStrings() allows you to load URL
Generate word list according to syllabus count

Rhyming is hard → Rita can do
Speech detect program in Java

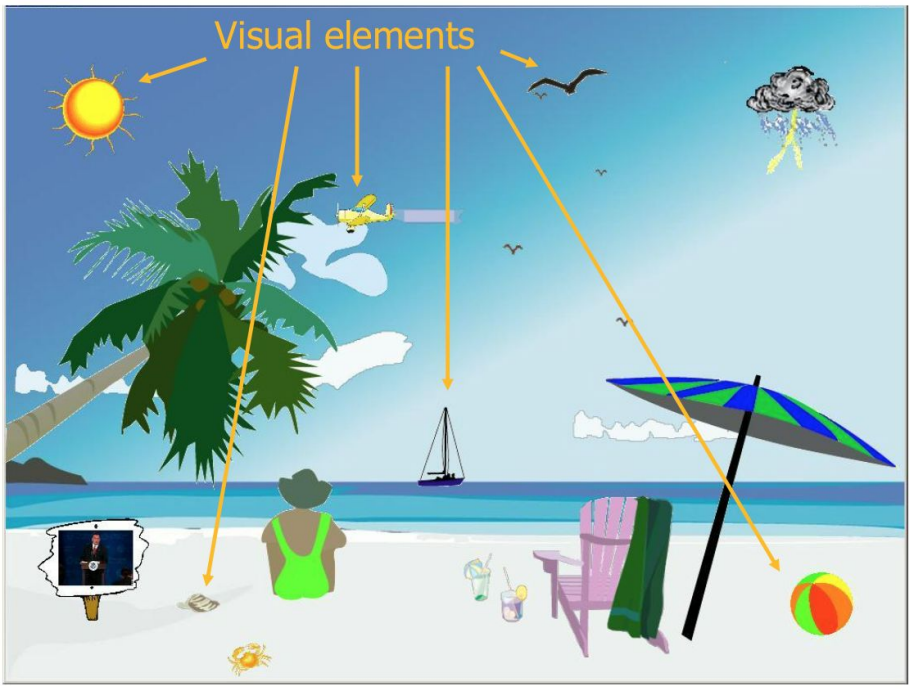
Weird days #coffeeday #alldaybreakfastday #neutella day
How do people tweet in the morning/evening, nice words/mean words
Can specify from what day you can pull from; live with a week.

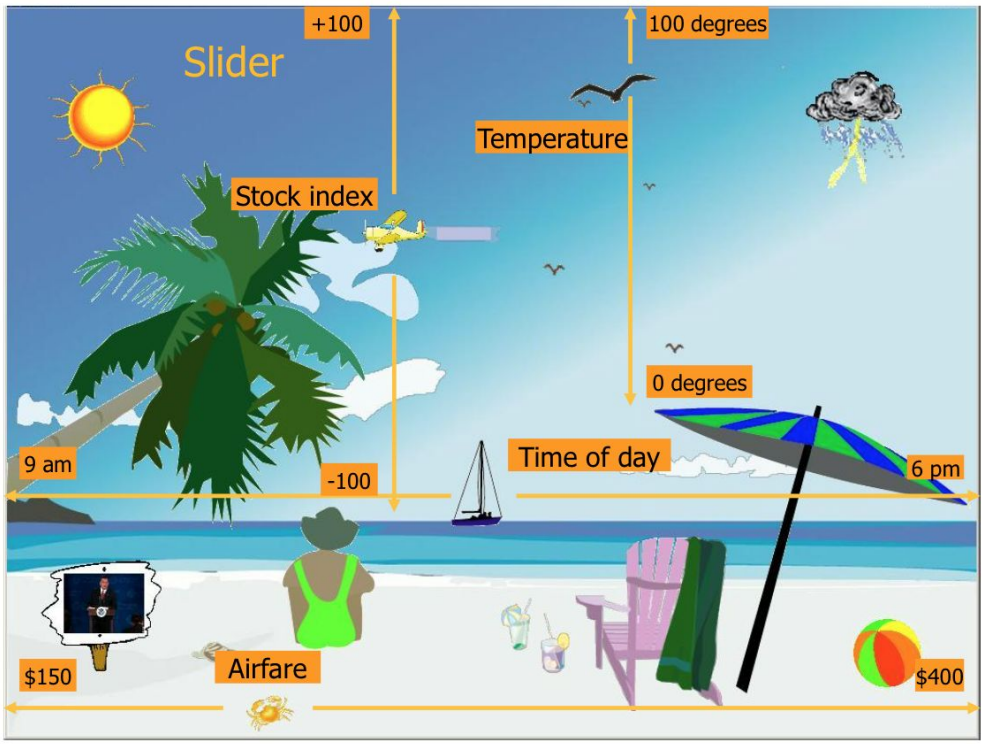
Use inspirational quote and song lyrics to generate poem to generate meme.

Clicking on different parts of the world and scene change/background color change
Has to be live data
Regions, certain time of the day

Wednesday Oct 7

Tutorial: http://codasign.com/tutorials/processing-and-twitter/
Idea: What does Twitter say my room (ambient area) look like?





Appearance
Scaler
Populater
Projector
Visual mapping and polls data sources to maintain current representation

- To do:
    - Decide what “room” or space metaphor I want to adopt
    - Map the appearances to specific parameters Twitter can pull from the tweets
        - geocode
            - The location is preferentially taking from the Geotagging API, but will fall back to their Twitter profile. The parameter value is specified by “latitude,longitude,radius”, where radius units must be specified as either “mi” (miles) or “km” (kilometers). Note that you cannot use the near operator via the API to geocode arbitrary locations; however you can use this geocode parameter to search near geocodes directly. A maximum of 1,000 distinct “sub-regions” will be considered when using the radius modifier.
            - **Example Values**: 37.781157,-122.398720,1mi
        - lang

        - Restricts tweets to the given language, given by an ISO 639-1 code. Language detection is best-effort.
        - **Example Values**: eu
- locale
    - Specify the language of the query you are sending (onlyja is currently effective). This is intended for language-specific consumers and the default should work in the majority of cases.
    - **Example Values**: ja
- result_type
    - Optional. Specifies what type of search results you would prefer to receive. The current default is “mixed.” Valid values include:
    - * mixed: Include both popular and real time results in the response.
    - * recent: return only the most recent results in the response
    - * popular: return only the most popular results in the response.
    - **Example Values**: mixed, recent, popular
- count
    - The number of tweets to return per page, up to a maximum of 100. Defaults to 15. This was formerly the “rpp” parameter in the old Search API.
    - **Example Values**: 100
- until
    - Returns tweets created before the given date. Date should be formatted as YYYY-MM-DD. Keep in mind that the search

index has a 7-day limit. In other words, no tweets will be found for a date older than one week.

    - **Example Values**: 2015-07-19

- since_id
    - Returns results with an ID greater than (that is, more recent than) the specified ID. There are limits to the number of Tweets which can be accessed through the API. If the limit of Tweets has occured since the since_id, the since_id will be forced to the oldest ID available.
    - **Example Values**: 12345
- sdfsd
- sfd
-

Using Twitter to map what a person's room will look like at different times of the day.

Thursday Oct 8

http://ashley-bovan.co.uk/ can download zipfiles for word lists

Dreamingthedream.com

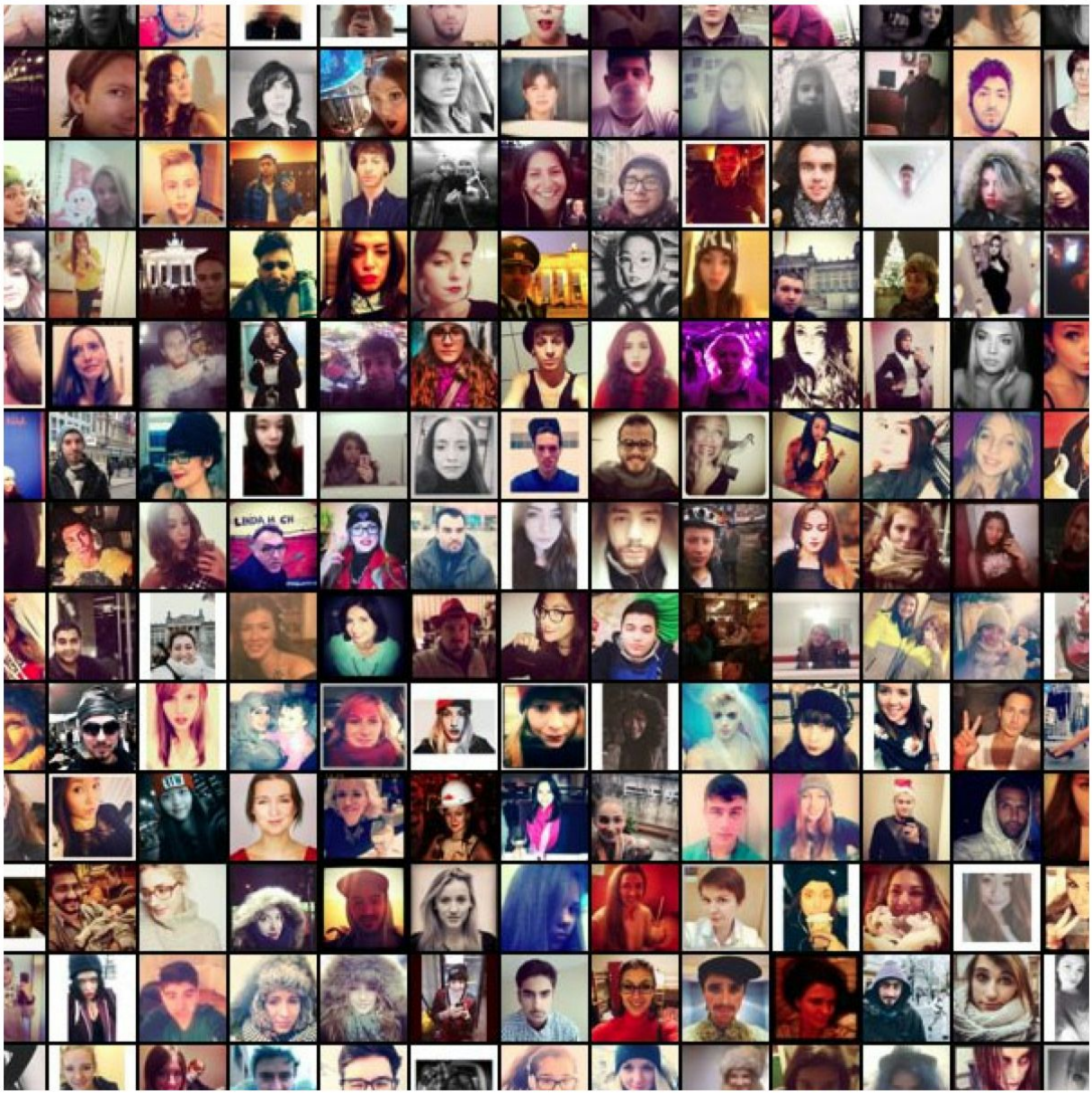
Type in keyword of your dream and generate an image of your dream meaning

Idea Generation

World Map with #selfie

Example: Selfie City: a Visualization-Centric Analysis of Online Self-Portraits
http://infosthetics.com/archives/2014/02/selfie_city_a_visualization-centric_analysis_of_self-portraits.html?utm_campaign=buffer&utm_medium=social&utm_content=buffer094d5&utm_source=twitter.com

MORE



Mood Flower/Mood-ometer

Compose strings of different moods such as:

happy{#thrilled, #excited, #happy, #glad,...} → red to pink

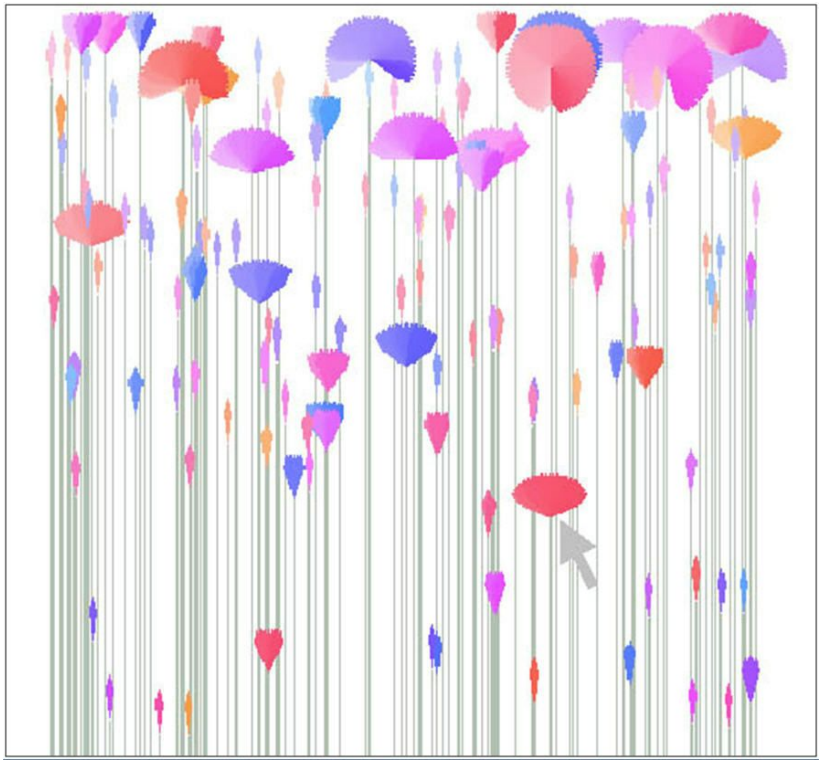
sad {#blue, #sad, #depressed, #crying,...} → blue to grey

somber{ … } → yellow to brown

jealousy{#jealousy, #salty} → green

… ...

Search for hashtags within each string. Assign a range of rgb color to each string and color the flowers based on what people are feeling like right now on Twitter.



Mood Clouds (words' color represent mood):
#engage, #funeral, #exam, #restaurant, #date, #depressed, #wedding. Each hashtag has a different background for the sky. The cloud will be transparent to show the color.

Making tweets into inspiration memes
Substituting words with inspirational word list

Night sky meteor and shooting stars
Based on a topic: mom, dad, grandma, grandpa, friend with hashtag #rip

Cat lover vs. dog lover substitute the words "cats" and "dogs" with "my boyfriend" or "my girlfriend"
Show whether they love their pets as much as they love their human partners.

Soccer vs. football vs. baseball vs. basketball rain
Depending on the location. The rain droplets will be the shape of the balls the user clicks on. European has more soccer rain, and American has more baseball and football rain.

Halloween theme

Pumpkins being masked and carved into scary face as Halloween approaches.

Transform tweets into famous architecture:
Paris tweets construct an eiffel tower outline: #eiffel tower #selfie #france #paris
New York tweets construct an empire state building outline
Italy tweets construct an Pisa tower outline.

Breast cancer awareness month
Survival story in the shape of the pink ribbon
#rip → meteor

How many people got married each month this year, in the shape of a white dress #justmarried, dove, ribbon, #bouquet

Cecil the lion and the dentist
Sarcastic representation of the lion hunting the dentist

Some kind of emotion with cracked screen

Friday Oct 16

```
import twitter4j.conf.*;
import twitter4j.*;
import twitter4j.auth.*;
import twitter4j.api.*;
import java.util.*;

Twitter twitter;

/*
Starry nightsky poetic interaction for Breast Cancer Awareness survivors
Author: Biqing Li
```

```
*/

//DECLARE
ShootingStar myShootingStar;

// the twinlking star locations
int[] starX = new int[500];
int[] starY = new int[500];
color[] starColor = new color[1000];
int starSize = 4; // the size of the stars

// the tail of the shooting star
int[] shootX = new int[30];
int[] shootY = new int[30];
int METEOR_SIZE = 8; // initial size when it first appears
float meteorSize = METEOR_SIZE; // size as it fades

// distance a shooting star moves each frame - varies with each new shooting star
float ssDeltaX, ssDeltaY;
// -1 indicates no shooting star, this is used to fade out the star
int ssTimer = -1;
// starting point of a new shooting star, picked randomly
int startX, startY;


void setup() {
  size(1000,800);

  ConfigurationBuilder cb = new ConfigurationBuilder();
  cb.setOAuthConsumerKey("SSLkyRqHSnQr7i88dp1oDk5FN");

cb.setOAuthConsumerSecret("Woa1Yjx116bb6Qwdn20wmhpmdHpmxcBwD60bKlq7Q
MVDIsL93n");

cb.setOAuthAccessToken("480462489-u1LZwX7mCdEpDgIHnFHg7XwA5CndcpZSrPq
hzN7F");

cb.setOAuthAccessTokenSecret("X63iEo292Soc3ZDFvD701zlWqzLoyHdqRwMzInZhb
g7kS");
```

```
  TwitterFactory tf = new TwitterFactory(cb.build());

  twitter = tf.getInstance();

  //INITIALIZE
  myShootingStar = new ShootingStar();


  // create the star locations
  for (int i = 0; i < starX.length; i++) {
          starX[i] =(int)random(width);
          starY[i] = (int)random(height);
          starColor[i] = color((int)random(252,255));
  }
}

void draw() {
  //background(0,0,50); // dark blue night sky
   background(255,102,178); //pink sky

  //CALL FUNCTIONALITY
          myShootingStar.run();
  // draw the stars

  noStroke();
  strokeWeight(1);
  for (int i = 0; i < starX.length; i++) {
          fill(random(50,255)); // makes them twinkle
          if (random(10) < 1) {
          starColor[i] = (int)random(100,255);
          }
          fill(starColor[i]);

          ellipse(starX[i], starY[i], starSize, starSize);
  }

  // draw the shooting star
  for (int i = 0; i < shootX.length-1; i++) {
```

```
        int shooterSize = max(0,int(meteorSize*i/shootX.length));
        // to get the tail to disappear need to switch to noStroke when it gets to 0
        if (shooterSize > 0) {
        strokeWeight(shooterSize);
        stroke(255);
        }
        else
        noStroke();
        line(shootX[i], shootY[i], shootX[i+1], shootY[i+1]);
        // ellipse(shootX[i], shootY[i],
meteorSize*i/shootX.length,meteorSize*i/shootX.length);
  }
  meteorSize*=0.94; // shrink the shooting star as it fades

  // move the shooting star along it's path
  for (int i = 0; i < shootX.length-1; i++) {
        shootX[i] = shootX[i+1];
        shootY[i] = shootY[i+1];
  }

  // add the new points into the shooting star as long as it hasn't burnt out
  if (ssTimer >= 0 && ssTimer < shootX.length) {
        shootX[shootX.length-1] = int(startX + ssDeltaX*(ssTimer));
        shootY[shootY.length-1] = int(startY + ssDeltaY*(ssTimer));
        ssTimer++;
        if (ssTimer >= shootX.length) {
        ssTimer = -1; // end the shooting star
        }
  }

  // create a new shooting star with some random probability
  if (random(20) < 1 && ssTimer == -1) {
        newShootingStar();
  }
}

/*
  Starts a new shooting star by randomly picking start and end point.
*/
```

```
void newShootingStar() {
  int endX, endY;
  startX = (int)random(width);
  startY = (int)random(height);
  endX = (int)random(width);
  endY = (int)random(height);
  ssDeltaX = (endX - startX)/(float)(shootX.length);
  ssDeltaY = (endY - startY)/(float)(shootY.length);
  ssTimer = 0; // starts the timer which ends when it reaches shootX.length
  meteorSize = METEOR_SIZE;
  // by filling the array with the start point all lines will essentially form a point initialy
  for (int i = 0; i < shootX.length; i++) {
        shootX[i] = startX;
        shootY[i] = startY;
  }
}


class ShootingStar{
  //GLOBAL VARIABLE
  int[] ShootingStarX=0;
  int[] ShootingStarY=0;
  int[] ShootingStarSize=0;
  float ShootingstarFadeSize = 0;
  float speedX = -2;
  float speedY = -.5;

  //CONSTRUCTOR
  ShootingStar(int[] ShootSize, int[] shootX, int[] shootY, float Meteorsize){
        ShootingStarX = shootX;
        ShootingStarY = shootY;
        ShootingStarSize = ShootSize;
        ShootingstarFadeSize = Meteorsize;

  }

  //FUNCTIONS
  void run(){
  //display();
```

```
  //fade
  //move
  }

  void move(){
   ShootingStarX += speedX;
   ShootingStarY += speedY;

  }

  void fade(){

  }


  void display(){
        ellipse(200,200,20,20);

  }

}
```

Friday Oct 20

Feedback from the class:

1. Pick 1 or 2 words out of a tweet, have them display on the sky and slowly fades away as you hover over the star
2. Get rid of the ribbon altogether

Saturday Oct 21

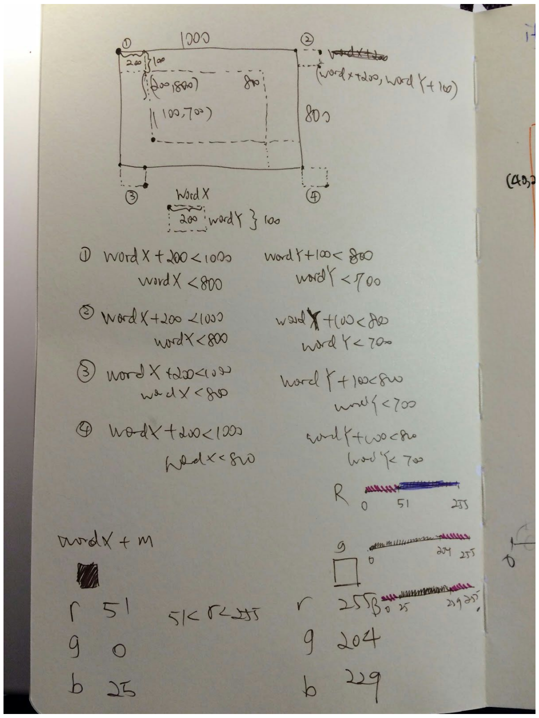
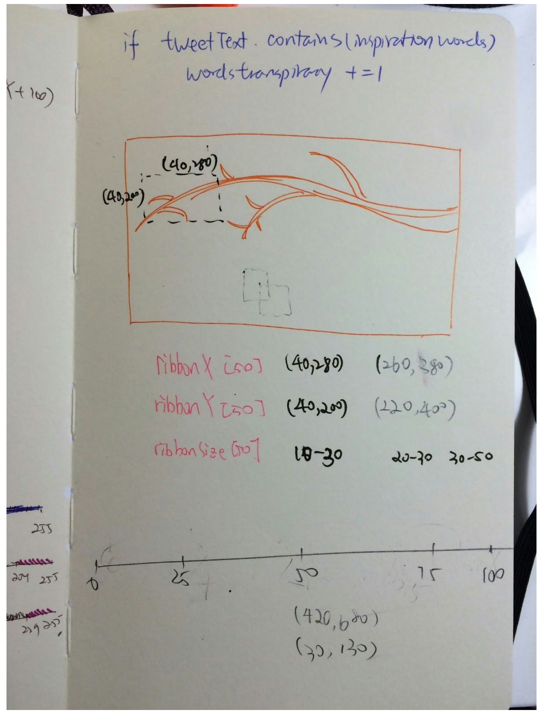
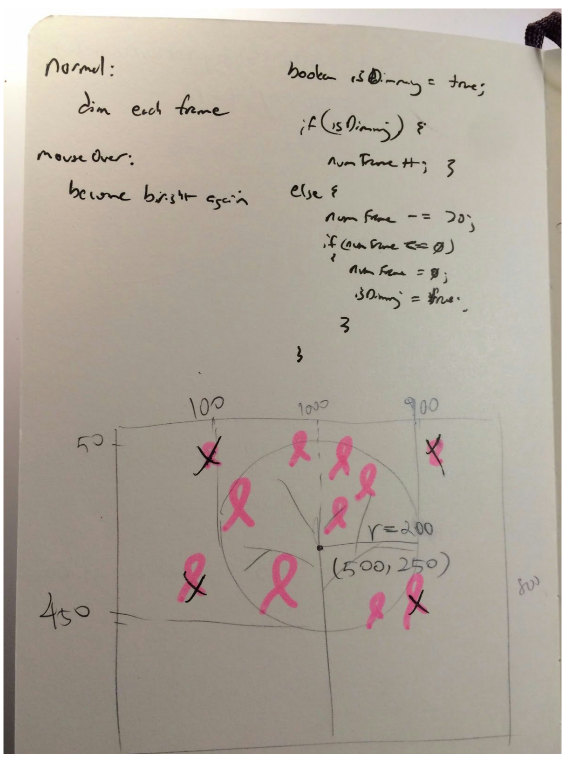
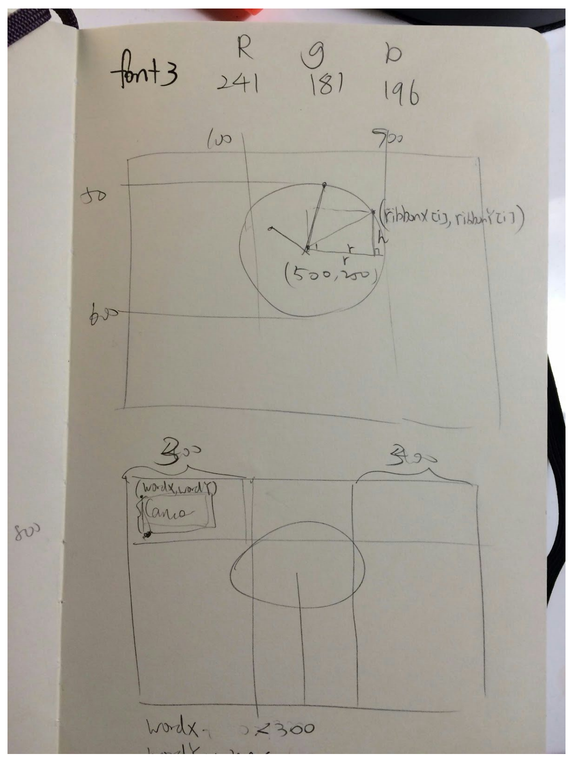
Steps to build the app and to-do list

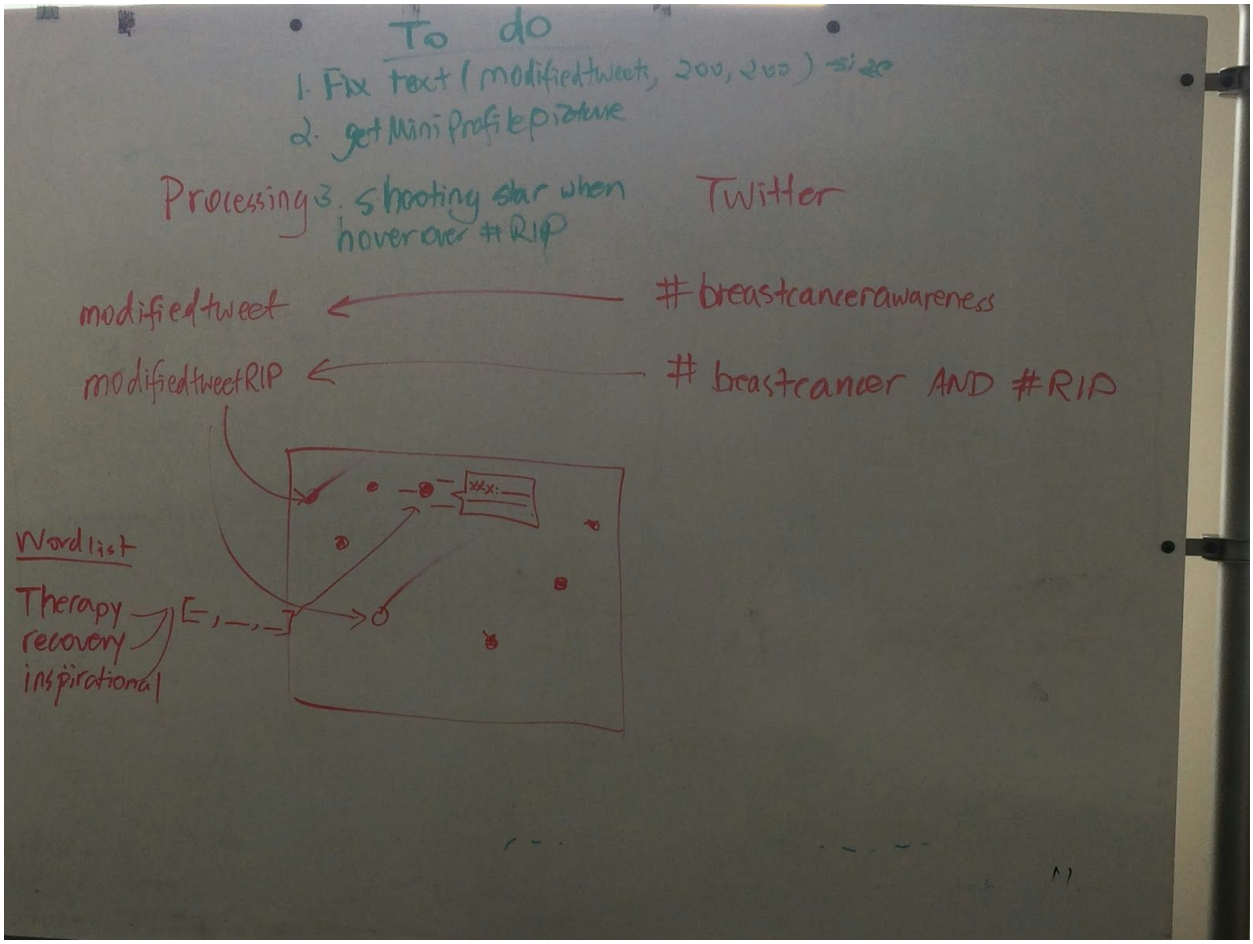
0a. Register as dev with Twitter
0b. Grab twitter4j and install
1 Configuration builder (setAPI keys, tokens,)
2 Create TwitterFactory instance using our configuration build
3 Set the Query
4 Try/Catch (Twitter Exception)
5 Query the Twitter Obeject
6 Catch response in an ArrayList
7 Get the zeroth result as a Status Object

8 Get user name from status
9 get message from status
10 concatenate

Sunday Oct 22

Calculation for star positions


1000
(word X+200, word Y+100)
800
word X
200
word Y } 100
① word X + 200 < 1000
word X < 800
word Y + 100 < 800
word Y < 700
② word X + 200 < 1000
word X < 800
word Y + 100 < 800
word Y < 700
③ word X + 200 < 1000
word X < 800
word Y + 100 < 800
word Y < 700
④ word X + 200 < 1000
word X < 800
word Y + 100 < 800
word Y < 700
word X + m
r 51
g 0
b 255
51 < r < 255
r 255
g 204
b 229

if tweetText.contains(inspiration words)
wordsInspiration += 1
(40,280)
(40,200)
ribbonX [50] (40,280) (260,280)
ribbonY [50] (40,200) (220,400)
ribbonSize [50] 10-30 20-30 30-50
0 25 50 75 100
(420,680)
(30,130)

normal:
dim each frame
mouse over:
become bright again
boolean isDimming = true;
if (isDimming) {
numFrame ++; }
else {
numFrame -= 20;
if (numFrame <= 0)
numFrame = 0;
isDimming = true;
}
}
100
1000
900
50
450
r=200
(500, 250)

font3
R 241
g 181
b 196
100
900
50
600
(ribbonX[i], ribbonY[i])
(500, 250)
300
300
(wordX, wordY)
800



Still need to be debugged:

1. text(tweetText, starX[k],starY[k]); won't let me draw a box to wrap the tweets. Whenever I indicate a box xPos and yPos to the text() function, the tweets disappear altogether. I suspect it is a bug with the customized font I am using.

screenshot of the buggy area

2. Detecting overlap among inspirational phrases