The title: CS3451 Fall 2014, Project1 TRIAL VISUALIZATION
 Author:    Bo Chen          Shen Yang



Project 1 TRIAL VISUALIZATION:
Implement different modes of manipulating a blue point by teleporting, when the mouse is pressed.

We have created 3 modes of the movement of circles, from the basic to smooth modes.  In the first mode, we just create the basic trail to contain 30 last location of points, by using queue to contain it. In the second mode, we are updating the location of each point to make it move forward to next point's location by using midpoint of two points.

Here are some codes we did in the program:

```
void drag(float x, float y){

    for (int j = 0; j < 1; j++) {
      for (int i = 0; i < trailx.length-1; i++ ){
          newTrailX[i] = (trailx[i] + trailx[i+1])/2;
          newtraily[i] = (traily[i] + traily[i+1])/2;
      }
      newTrailX[trailx.length-1] = x;
      newtraily[trailx.length-1] = y;
    }
    trailx = newTrailX;
    traily = newtraily;
}

void push(float x, float y){
    for(int i = 0; i < trailx.length-1; i ++){
        trailx[i] = trailx[i+1];
        traily[i] = traily[i+1];
    }
    trailx[trailx.length-1] = x;
    traily[trailx.length-1] = y;
}
```

Here is how our program works:

$$P_i = P_{i+1} \text{ (Mode 1)}$$

$$P_i = \frac{(p_i + p_{i+1})}{2} \text{ (Mode 2)}$$

*Mode 2 && 1*

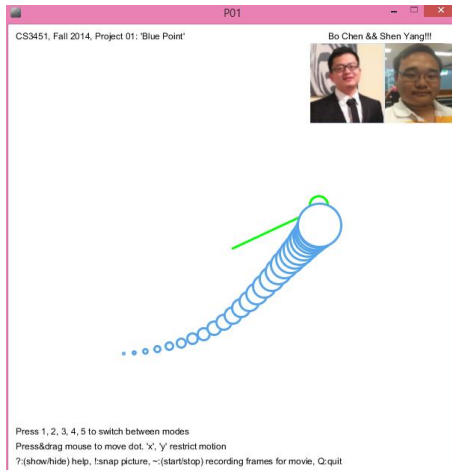

*Mode 1*



*Mode 2*

In our mode3, we got the idea from mode2! In the mode2, the circle has been moving too fast toward the next point, so in mode 3 we repeat the process of getting average. This makes it more smooth than mode2 and slows the trail's disappearing time.

```
void mode3(float x, float y) {

    // Repeating drag(x, y) twice since it repeats the average.
    drag(x, y);
    drag(x, y);
} // end of mode3
```

*Mode 3*



$$P_i = \frac{\dfrac{p_i + p_{i+1}}{2} + \dfrac{p_{i+1} + p_{i+2}}{2}}{2}$$

Pros and Cons between mode 1, 2, 3:
- Mode 1 has a simple algorithm and is easy to understand, also saves memory and time.
  - However, mode 1 does not feel smooth due to the spaces between the circles, and the animation does not feel continuous.
- Mode 2 uses a new array to compute the new positions of the circles. The animation is better than mode 1 (smoother).
- In mode 3, it uses the same amount of memory as mode 2 but more time, and the animation is has a better feel than mode 2.
  - However, the more it repeats the averaging process, the faster the circles shrink and disappear. Too much averaging may cause the trail to be hard to track.

Extra credit:
A color changing feature. We keep all the color in an array, to keep changing the trail's color. It looks way cooler than before.

There are two extra modes we have to create in this project just for extra fun and extra credit. In the mode 4, we display the acceleration and speed of circle movement by changing circle size. As the mouse moves faster, the circle will become

bigger, if the mouse moves slower, the circles are smaller. In other words, the size of the circle is proportional to the distance between the current and the one before.

```
// If it is not mode 4, it shrinks and disappear
if (!mode4) {
    showDisk(trailX[i], trailY[i], i);
} else if (mode4) { // If the mode is 4, the size is proportional to the distance of the current and circle before.
    if (i < 29) {
        showDisk(trailX[i], trailY[i], (trailX[i+1]-trailX[i])*2);
    }
}
```
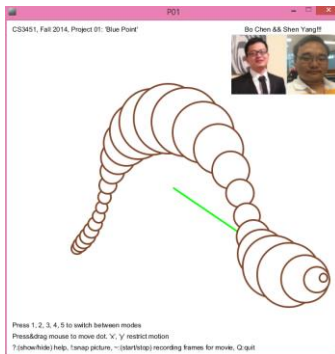


*Mode 4*

The mode 5, we call it High Mode! By using a random number to keep changing the position of the circles, it will continue shake. If you like it please thumb up!