CS106 W19 - Assignment 04

Due: Friday, February 1st, 11:59 PM

This assignment about “Advanced Shapes”. These topics were in class on the week of January 28th. If you need to refresh yourself, review the slides that were presented in class and the demo code examples that were provided. They are both on LEARN.

QUESTION ONE

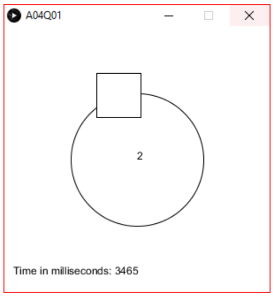
You are to create a Reaction Time game. Your game should mimic what is shown in the YouTube video. https://youtu.be/w4xSzWZ72qw Save your solution as A04Q01.

You must have a rotating rectangle. If the mouse is pressed on the rectangle the game starts. A clock begins. The user must press the rectangle again 4 more times. The resulting time is the player’s score and represents the player’s reaction time in milliseconds to hit the rectangle 4 times.

On the next page you are given starter code and the required `polar()` function. You only need to create the `mousePressed()` function. You may modify any of the starter code if you like, but it is not necessary.

Requirements:

1. The hit test must work correctly.
2. The “`turns`” and “`timePassed`” must update and display correctly.



```
Time in milliseconds: 3465
```

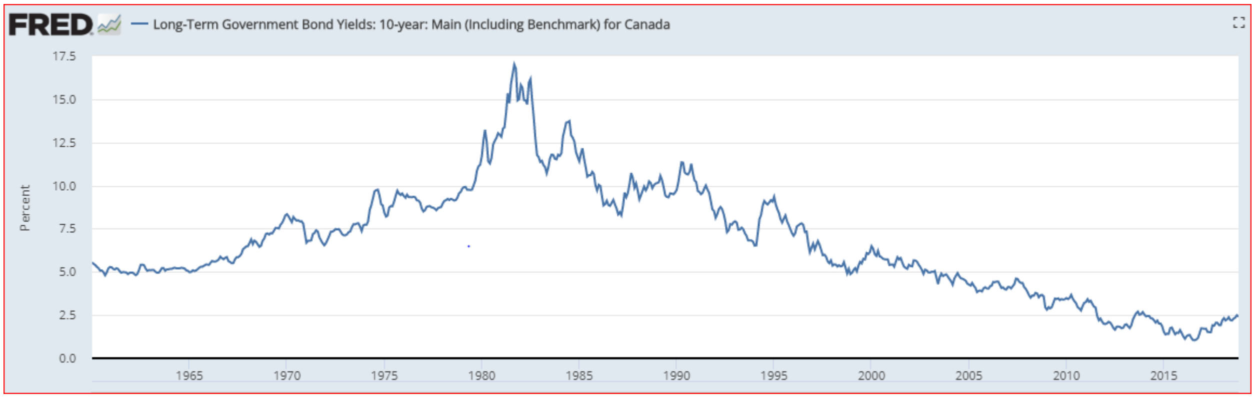
Starter code and `polar()` function.

```
float r = 75;
float theta = 0;
PVector pt;
int turns = 0;
int startTime = 0;
int timePassed = 0;
boolean gameOver = false;

void setup() {
  size(300, 300);
  background(255);
  rectMode(CENTER);
}

void draw() {
  if (!gameOver) {
    //  If it is not gameOver, determine the position
    //      of the rectangle.
    pt = polar(r, theta);
    fill(255);
    theta = theta + 1;

    // If it is not gameOver, continuously update the timer.
    if (turns > 0 && turns < 5) {
      timePassed = millis() - startTime;
    }
  }

  // Draw everything regardless of whether it is gameOver or not.
  background(255);
  fill(255);
  ellipse(width / 2, height / 2, r * 2, r * 2);
  rect(width / 2 + pt.x, height / 2 + pt.y, 50, 50);
  fill(0);
  text(turns, width / 2, height / 2);
  text("Time in milliseconds: " + timePassed, 10, height -20);
}
```

```
PVector polar(float r, float theta){
  float dt = radians(theta);
  float x = r * cos(dt);
  float y = r * sin(dt);
  return new PVector(x, y);
}
```

QUESTION TWO: Canadian Historical Interest Rates

The website https://fred.stlouisfed.org contains various data. It has data for the monthly interest rates in Canada from 1960 to 2018. Shown in a graph the data looks as follows:



The graph shows, for example, that interest rates in the very early 1980's were historically very high.

We have the data in a txt file named `interestRates.txt`. We could load the data using the following commands in Processing:

```
String[] lines;
lines = loadStrings( "interestRates.txt" );
```

The data file `interestRates.txt` has four columns representing the month, day, year and interest rate. The data file `interestRates.txt` has 707 lines and starts as follows:

```
1  1  1960     5.54
2  1  1960     5.46
3  1  1960     5.39
4  1  1960     5.29
5  1  1960     5.21
6  1  1960     5.07
…
…
…
```

You can click on the file to see the complete contents of the file.

Write a sketch that reads `interestRates.txt` and draws a graph that mimics the graph on the previous page. Save your solution as A04Q02

You can make the assumption that the interest rate is always between zero and 20%.

Your requirements are as follows:

1. The line on the graph must be drawn correctly.
2. You must have a margin on the bottom (x-axis) to show the years in increments of 5.
3. You must have a margin on the left (y-axis) to show the interest rate in increments of 2.5%.

You must use `map()` to determine the x and y positions for each of the 707 data points.

You must use `beginShape(), vertex(),` and `endShape()` to draw the graph.

This question has a lot of similarities to the demo code examples `SeaIceJanuary.pde` and `SeaIceMonthly.pde`.

Requirements and Grading

QUESTION ONE: (8 marks)

[4 marks] Correctness

- The sketch must meet the requirements listed above.

[2 marks] Coding Style

- Comment your code appropriately. Avoid superfluous comments.
- Correctly and consistently indent your code blocks.
- Use correct inline spacing in function calls, function definitions, and variable declaration and assignment.
- Use good line spacing to chunk sections of your code.
- Pay special attention to inline spacing for your conditional statements
- One or more marks may be deducted for solutions that have obvious inefficiencies.
- Variables that are declared or assigned, but not used.
- Unnecessarily repeating the same code in multiple places.

[2 marks] Visual Design and Creativity

- Higher marks will be given to sketches with extra details for creativity and artistic appeal.

QUESTION TWO: (14 marks)

[10 marks] Correctness

- The sketch must meet the requirements listed above.

[2 marks] Coding Style

- Comment your code appropriately. Avoid superfluous comments.
- Correctly and consistently indent your code blocks.
- Use correct inline spacing in function calls, function definitions, and variable declaration and assignment.
- Use good line spacing to chunk sections of your code.
- Pay special attention to inline spacing for your conditional statements
- One or more marks may be deducted for solutions that have obvious inefficiencies.
- Variables that are declared or assigned, but not used.
- Unnecessarily repeating the same code in multiple places.

[2 marks] Visual Design and Creativity

- Higher marks will be given to sketches with extra details for creativity and artistic appeal.

Submitting

Create a folder “A04_username”, but replace “username” with your UW id. So if your email is “jac926@edu.uwaterloo.ca” you would create a folder “A04_ jac926”.

SAVE your sketches in that folder as “A04Q01_username” and “A04Q02_username”. Again, replace username with your UW id.

Zip your “A04_username” folder (with “username” replaced by your UW id) and submit it the correct assignment dropbox.

It is your responsibility to submit to the correct dropbox with the correct files before the deadline. Otherwise you will have marks deducted.

Academic Integrity

All assignments in CS106 are done individually. Group work and sharing of code is not allowed.

Detecting Plagiarism:

- We monitor Reddit, File Trading Sites, past year CS106 assignments, etc.
- We use Measure Of Software Similarity (MOSS)
    - automatic system for determining the similarity of code

Discipline

Discipline (Policy 71)

- https://uwaterloo.ca/secretariat-general-counsel/policies-procedures-guidelines/policy-71