King Fahd University of Petroleum & Minerals
College of Computing and Mathematics

Information and Computer Science Department

ICS 415: Computer Graphics (3-0-3)

First Semester 2023-2024 (Term 231)

Due date: September 30, 2023 **Homework #** 01

Problem # 01

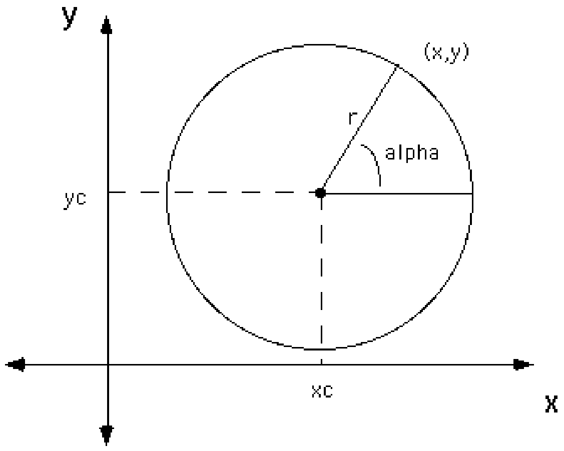
[25 POINTS]

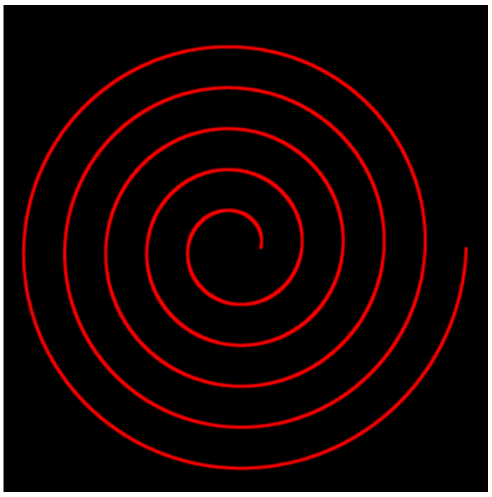
In this problem, you will generate the coordinates for vertices around a spiral-shaped curve (Archimedean Spiral) and use these coordinates with WebGL to display a spiral. The Archimedean spiral is the trajectory of a point moving uniformly on a straight line of a plane, this line turning itself uniformly around one of its points.

First, recall how the (x, y) coordinates of evenly spaced points around a circle can be generated using polar coordinates. The radius of the circle, r, remains fixed while the angle alpha shown below increases from 0 deg to 360 degrees around the circle (or equivalently, from 0 to 2*PI radians). The coordinates of the center of the circle are (xc, yc).

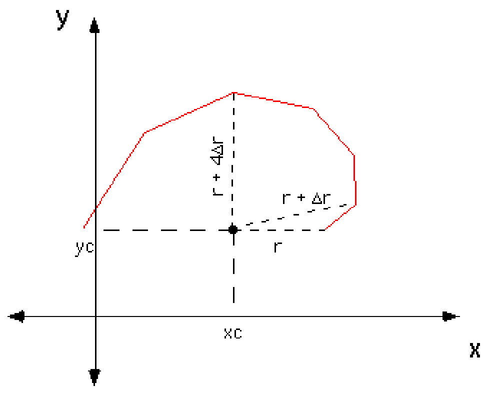
x = xc + r * cos (alpha)

y = yc + r * sin (alpha)





The coordinates for the vertices around a spiral figure can be created by varying the radius r as the angle alpha changes, as illustrated below:



For this assignment, you can assume that the distance between the turns and the tightness of the spiral, both are constant. Such as:

// Constant controlling the distance between turns
const a = 0.1;

// Constant controlling the tightness of the spiral
const b = 0.025;

// Spiral equation
r = a + b * alpha

[Note: you may want to change the a and b parameters to control the shape of the spiral.]

Getting Started

1. You can start with triangle.html and triangle.js files. Change the name of triangle.html to spiral.html.

2. Edit spiral.html to change the reference to "triangle.js" to "spiral.js" in the script tag. Also, add an HTML comment with your name, date, and assignment number near the top of the spiral.html file.

3. Change the name of the triangle.js file to spiral.js. Use Javascript comments to add your name, date, and assignment number to the top of the file.

4. Modify the spiral.js program to create a WebGL program that displays a spiral:

    - Write a function named spiral( ) that generates the (x,y) coordinates for successive vertices around a spiral (Hint: use a for loop). You should push each (x, y) vertex onto the points array.

- Call your spiral function from the init( ) function (in place of the code that generates the vertices for the triangle).
- The function should have variables (these can all be local to the function) corresponding to the following information:
    - number of vertices
    - an initial angle, alpha, to indicate where to draw the first point on the spiral
    - an amount to increment alpha for each new vertex along the spiral
    - an initial radius r for the first vertex on the spiral
    - an amount to increment r for each new vertex along the spiral
    - the coordinates xcenter, ycenter that indicate the location of the center of the spiral
- You will need to use the Math.sin( ) and Math.cos( ) functions from the JavaScript math library. Note that the sin and cos functions in JavaScript require the angle to be specified in radians. If you give alpha and its increment in degrees, you must convert to radians by multiplying by Math.PI/180.0. (Pi is specified in the Javascript math library by the constant, Math.PI)

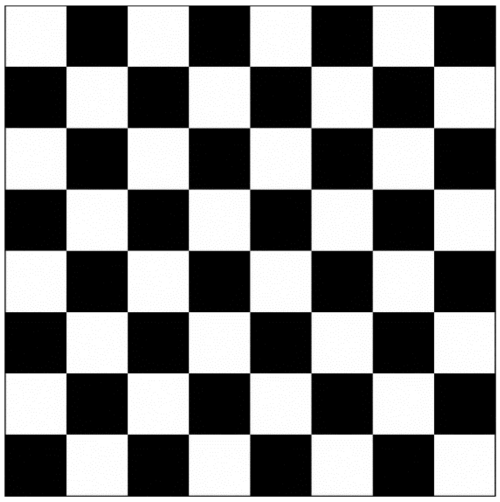
What to submit for Problem # 01?

1. spiral.html
2. spiral.js
3. spiral.png (image screenshot of your spiral)

Problem # 02

[25 POINTS]

In this problem, you will write a program using JavaScript and WebGL to display a chessboard (8x8 grid of alternate black and white square boxes) and respond to user input for a simple game. WebGL should be used to display the graphics of the game and allow the player to select the boxes. You should create a well-designed program, making effective use of functions, informative variable names, appropriate data types, good comments, and so on. The aim of this assignment is to combine JavaScript and WebGL programming to create a simple game with a particular graphical interface for your program.



Your task will be to write a WebGL program using JavaScript to display the chessboard and allow the users to select the boxes from the chessboard by doing the mouse left-click. Every time the user left-clicks on any of the boxes, the program should change the color of that box. If the box is originally Black, change it to White color, if it is White, change it to Black color. After every change, the updated state of the chessboard should be displayed graphically in the Drawing window, using WebGL graphics commands. Your solution will be evaluated for programming style and correctness, as well as the quality of your graphical display.

To check the color of a pixel where the user clicked in WebGL, you can use the gl.readPixels() function. For example:

```
// Read the pixel color at the clicked location
const pixelData = new Uint8Array(4); // RGBA color values will be stored here
gl.readPixels(clickX, clickY, 1, 1, gl.RGBA, gl.UNSIGNED_BYTE, pixelData);

// Convert pixelData to a human-readable format (0-255 range)
const red = pixelData[0];
const green = pixelData[1];
const blue = pixelData[2];
const alpha = pixelData[3] / 255.0; // Normalize alpha value to range [0, 1]
```

Getting Started

You may want to start by copying one of your previous programs. Name your JavaScript program file chessboard.js, and the html file, chessboard.html. Use WebGL to draw the game board with the initial set-up such that all the boxes have Black and White colors alternatively in the graphics window. Do not simply write a render function that contains a long list of gl.drawArrays( ). Instead, use loops to draw several lines at once where possible.

What to submit for Problem # 02?

1. chessboard.html
2. chessboard.js
3. chessboard.png (image screenshot of your chessboard)