Raster Graphics

CS 452 - Computer Graphics
Assignment 2

For this assignment, you will need to understand the basics of rasterization, and how to convert vector graphics to raster graphics. You will implement five basic rasterization functions.

Download the **"CS452_Assign2_SkelCode_MSVC"** Project from LMS and open it. Build and run the project. If you see a white window, congratulations. You're all set up and good to go. This project is mostly skeleton code that has already been written for you in order for you to visualize your output in a nice way. You are required to add implementations for the following five tasks in **myraster.cpp** file:

1. drawLine
When this function is called with the given arguments, it should paint a line on the given frame buffer with the given color. For this, you must implement the Bresenham algorithm for line drawing, otherwise known as the "midpoint" algorithm. This algorithm was discussed in class.

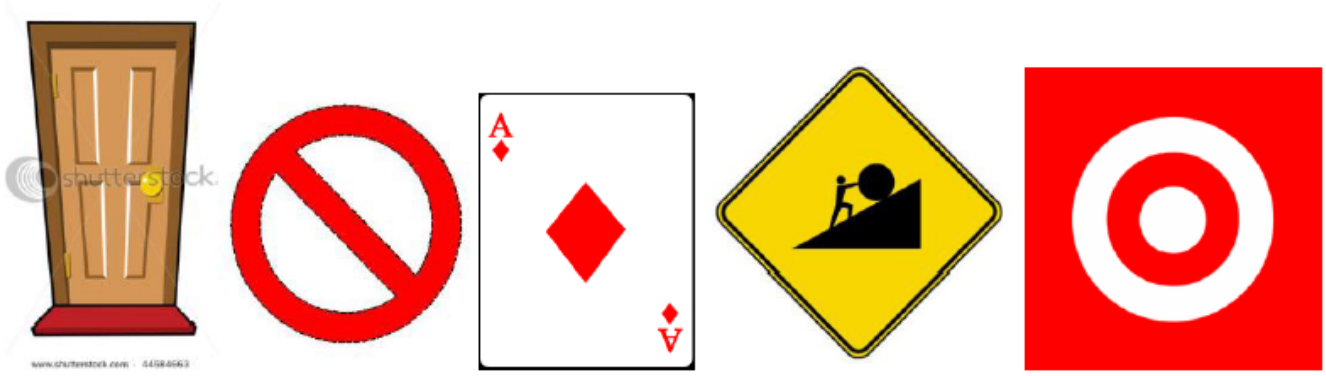
2. drawCircle
When this function is called with the given arguments, it should paint a perfect circle on the given frame buffer with the given color. This is also doable with the midpoint algorithm discussed in class.

3. floodFill
This function mimics the paint bucket tool present in the Paint application in Microsoft Windows. The color previously present at the seed point in the frame buffer is the "source" color. The source color is to be replaced with the "destination" color, which is passed as an argument to the function. For every pixel painted with the destination color, the four pixels surrounding that pixel (up, down, left, right) are also to be checked. If they are also colored with the source color, then they should be painted with the destination color. If they are not colored with the source color, then they should not be painted with the destination color.

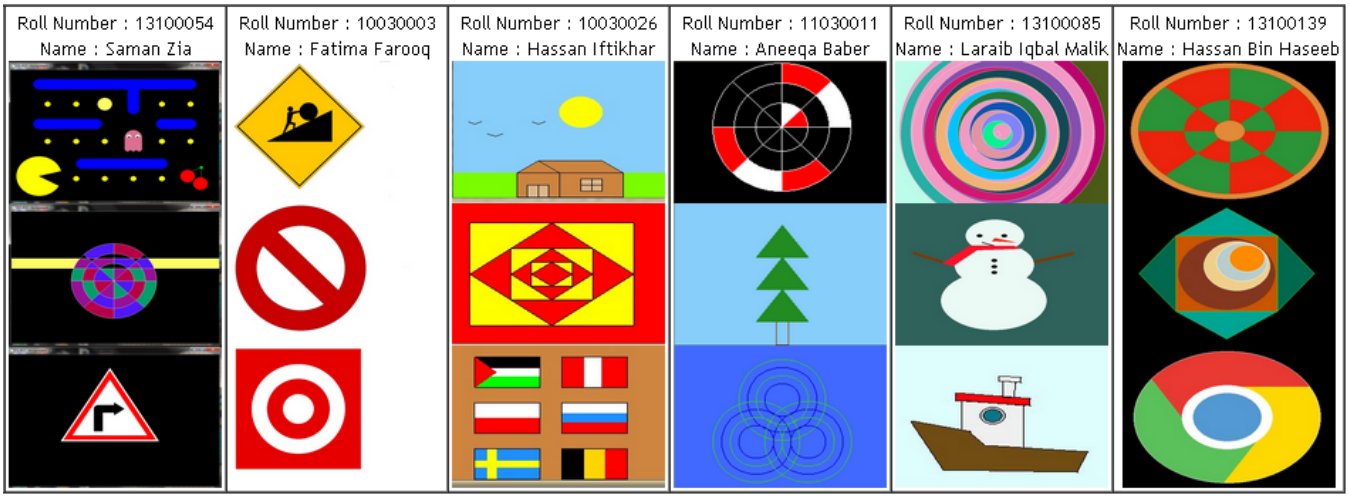
4. boundaryFill
This function is very similar to floodFill, except that it continues to seed neighboring pixels, unless the pixel matches the color of the boundary. The initial color at the seed location does not matter.

5. scanLine
This function renders a single polygon with the given edge color and the given fill color. Assume that the array of vertices passed to the function are already in order of connectivity. For example, the first and second vertices are connected, the second and third vertices are connected, the third and fourth vertices are connected, and so on. Also, remember that the first and last vertices are also connected, to make a completely closed polygon. You will call your drawLine function to draw the boundaries, and then implement the scanline algorithm to fill up the polygon, which may be highly concave.

Here are a few examples to show what kind of pictures we expect at the end of this assignment.



Results from last year's submission are also shown below as additional examples



Submission Details

- DUE: Monday Sep 30th, 11:55 p.m.
- Submission will be done on LMS.
- You must submit only your myraster.cpp file from your VC++ project. DO NOT SUBMIT THE ENTIRE PROJECT!. In case you have developed your code using MSYS/MINGW, submit all the files you have modified along with instruction on how to compile the code
- Submit at least 5 snapshots of various images that you have obtained