COMP SCI/ SFWR 3GC3: Computer Graphics Assignment 1

Due: Thursday, October 5, 2017 at 12:00pm (Noon not Midnight).
Accepted Late until Tuesday, October 10, 2017, 12:00pm, at a 20% per day penalty.
This assignment is worth 7% of your final grade.

Part A – Setup SVN

You have been provided an SVN trunk located at:

https://websvn.cas.mcmaster.ca/3gc3/<macid>

Where <macid> is your MacID used to log onto Mosaic and Avenue.
Ie. My MacID is gwosdzto, so my svn address is:
https://websvn.cas.mcmaster.ca/se3gc3/gwosdzto

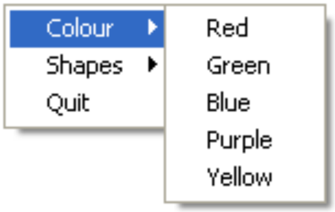
Check out a working directory of your SVN to your local machine.
Create Folders A1, A2, A3, Project, Test. You structure should be:

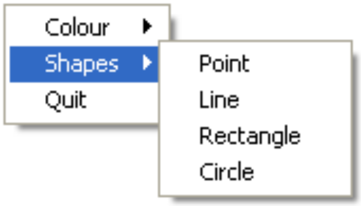
```
<macid>
|->A1
|->A2
|->A3
|->Project
|->Test
```

In Test, make a program which simply prints “Hello World” to the command prompt. When ‘make’ is called, a program Hello.x should be generated. Make sure all source files are in Test folder.

Part B – Simple Paint Program

Write a simple paint program in C++ using OpenGL/GLUT. The program should support drawing several primitive types and multiple colours (e.g., like MS Paint). Colours and shapes should be chosen via GLUT menus similar to the following:





You will need to keep track of both the current drawing colour and current shape. For each shape type supported, you must implement your own drawing function (i.e., see our discussion of raster algorithms). In other words, you cannot use the built-in primitive types such as GL_LINES, GL_POLYGON, etc, except for points. In all cases, only the point primitive type may be used, i.e., you must write your own functions to draw lines, rectangles, etc. using points.

Left clicking should draw primitives in the current colour, of the current type (as selected from the menu). You should support the following shapes:

a) Points – draw a dot at the point clicked with the mouse. Clicking and dragging should draw points constantly, i.e., free-form drawing. You can use the GL_POINT primitive for these.

b) Lines – draw a line between two subsequently clicked points. A few line drawing algorithms were discussed in class – full marks will only be awarded here for implementing Bresenham's algorithm!

c) Rectangles – draw a rectangle with top-left corner specified by the first click, and the bottom right corner specified by a second click.

d) Circles – draw a circle centered at the position of the first click, with its radius set by a second click (i.e., the length of the vector between the two clicks).

Note that this program needs to be double buffered – simply adding new colour to the frame buffer (i.e., by drawing primitives) will be unacceptable. Make sure to keep a list of objects and colours that will be "remembered" until the scene is cleared.

Do use: glClear, glutSwapBuffers, glutPostRedisplay, glut_Double_buffer.

You should include the following functionality (e.g., as GLUT menu items, or keys – include a reference with your code):

a) Drawing the primitive types above with your own drawing routines
b) Changing drawing colour
c) Clear the screen.
d) Quit the program
e) Random drawing colour
f) Radial PaintBrush option, which draws multiple points as brush around the centre point of the document (this will be one of the harder feature requests, make sure above is implemented first)

Use makefile so that when 'make paint' is called, a program called "Paint.x" will be compiled. Place these files into A1 folder.

Submission Notes

All programs must initially print out a list of keyboard/mouse/menu commands and how they are used in the program to the command prompt. The marker should not have to look at your source code to figure out how to run a program!! Marks will be deducted otherwise!

You have the option of implementing your assignment on your platform of choice, BUT please make sure your programs can be fully compiled and executed on the departmental machines (e.g., ITB 237). Also be sure to include any additional libraries that you use. If the TA has to hunt for a specific library to compile your code, you will lose marks.

Submit your source code, makefile, readme, and any other resources to A1 folder of your SVN trunk. All source code and makefile should be located in A1 and not a subfolder. When 'make' is called, it should produce programs named: "Paint.x" according to their source files.

Familiarize yourself with the department's policy on plagiarism and the university regulations on plagiarism and academic misconduct. Plagiarism will not be tolerated, and will be dealt with harshly.

Have fun, and be creative!