24-780 Engineering Computation Problem Set 08

You need to create a ZIP file (It may appear as a compressed folder in Windows) and submit the ZIP file via the 24-780 Canvas course. The file name of the ZIP file must be:

PS08-YourAndrewID.zip

For example, if your Andrew account is *hummingbird@andrew.cmu.edu*, the file name must be:

PS08-hummingbird.zip

If your ZIP file does not comply with this naming rule, you will automatically lose 5% credit from this assignment. If we are not able to identify who submitted the file, you will lose another 5% credit. If we finally are not able to connect you and the submitted ZIP file, you will receive 0 point for this assignment. Therefore, please make sure you strictly adhere to this naming rule before submitting a file.

The ZIP file needs to be submitted to the 24-780 Canvas course. If you find a mistake in the previous submission, you can re-submit the ZIP file with no penalty as long as it is before the submission deadline.

Notice that the grade will be given to the final submission only. If you submit multiple files, the earlier version will be discarded. Therefore, if you re-submit a ZIP file, the ZIP file MUST include all the required files. Also, if your final version is submitted after the submission deadline, late-submission policy will be applied no matter how early your earlier version was submitted.

Make sure you upload your Zip file to the correct location. If you did not upload your assignment to the correct location, you will lose 5%.

The ZIP file needs to include:

- C++ source file of your program (ps8-1.cpp)
- C++ source file of your program (ps8-2.cpp)

Submission Due: Please see Canvas.

START EARLY!

Unless you are already a good programmer, there is no way to finish the assignment overnight.

PS8-1 Bitmap drawing and pixel-value inquiry (ps8-1.cpp) [30 points]

Complete the following program so that the program opens 300x300 window and draws a 16x16 bitmap defined in the char pattern[]. The bitmap is 16x16 and character code of '1' represents a black pixel, and '.' represents an empty pixel. A pixel of the bitmap must be magnified to 16x16 square on the window. The program must terminate when the user presses ESC key.

When the user moves the mouse cursor over the window, print on the console window (1) mouse cursor coordinate relative to the top left corner of the window, and (2) '1' or '.' depending on the pixel under the mouse cursor. (Don't try to read the color from OpenGL. Use the pixel value stored in char pattern[].)

For example, when the mouse cursor is moved over to (50,50), your program must print:

```
50 50 1
```

Or, if the mouse cursor is moved over to (3,3), your program must print:

```
3 3 .
```

The program must not print anything if the mouse cursor is not over the bitmap.

The program must be saved as ps8-1.cpp and must be included in the Zip file. (Base code is available from the Canvas.)

Test your program with the compiler server! If you see a red-line, you may be using a platform- or compiler-specific feature, which may give you 50% point-deduction!

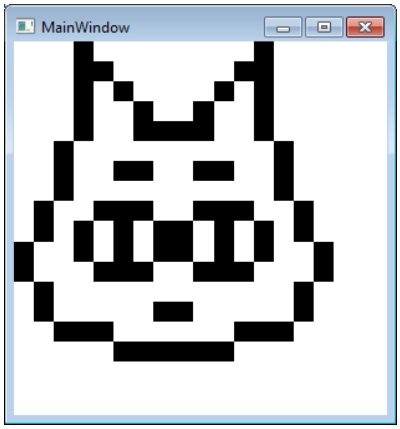
```
#include <stdio.h>
#include "fssimplewindow.h"

int main(void)
{
    // Character for the cell (x,y) will be pattern[y*16+x]
    char pattern[]=
    {
        // By the way, if you want to define a long string, you can write like this.
        "...1.......1..."
        "...11......11..."
        "...1.1....1.1..."
        "...1..1..1..1..."
        "...1..1111..1..."
        "..1.........1.."
        "..1..11..11..1.."
        "..1.........1.."
        ".1..111..111..1."
        ".1.1.1.11.1.1.1."
        "1..1.1.11.1.1..1"
        "1...111..111...1"
        ".1...........1."
        ".1.....11.....1."
        "..111......111.."
        ".....111111....."
    };

    FsOpenWindow(16,16,300,300,0);

    while(FSKEY_ESC!=FsInkey())
    {
        FsPollDevice();
        int lb,mb,rb,mx,my;
        if(FSMOUSEEVENT_MOVE==FsGetMouseEvent(lb,mb,rb,mx,my))
        {
        }
        FsSleep(10);
    }

    return 0;
}
```



Running image

PS8-2 Pixel value inquiry (ps8-2.cpp) [70 points]

Write a program (called ps8-2.cpp) that:

(1) Prints "Enter File Name:" on the console window.
(2) Takes file name as input from the console window. (Use Fgets member function of the TextString class we did in class.) (*)
(3) Reads a picture file (.PNG file).
(4) Opens the graphics window that is wider and taller by 10 pixels compared to the picture. (If the picture is 800x600, the window size must be 810x610)
(5) Draws a picture on the top-left corner of the window.
(6) Prints on the console window the mouse cursor coordinate and the pixel value (R,G,B, each ranges from 0 to 255) under the mouse cursor. Do not print anything if the mouse cursor is not over the picture. (Hint: A pixel consists of four unsigned char values (R,G,B,A) if you read the file by YsRawPngDecoder class. You don't have to print Alpha value.)
(7) Terminates when the user presses ESC key.

Be careful about the coordinate transformation. The lower left corner is (0,0) of the picture coordinate, while top left corner is (0,0) of the window coordinate.

The CPP file must be saved as ps8-2.cpp and included in the zip file.

5% Extra Point: When the user holds the mouse-left button down, draw 5x5 pixel blue square around the mouse cursor on the PNG image. But, your program must not crash when the user does so near the edge of the picture. Once the square is drawn, the pixel-value for the updated pixel should be (R,G,B)=(0,0,255).

Test your program with the compiler server! If you see a red-line, you may be using a platform- or compiler-specific feature, which may give you 50% point-deduction!

(*) In XCode, make sure to turn off SandBox. Also enter full-path file name