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. 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. 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:

ps08-1.cpp
ps08-2.cpp

Submission Please see Canvas

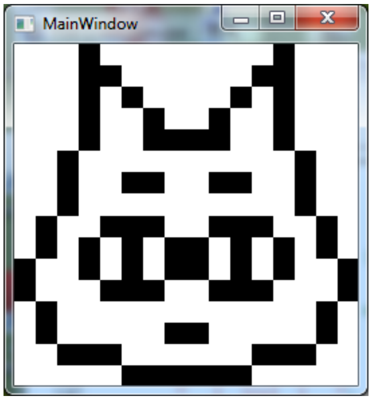
PS8-1 Bitmap drawing (30 points)

Complete the following program so that the program 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 must be magnified to 16x16 square on the window.

The program must be saved as ps8-1.cpp and must be included in the Zip file.

(You can cut & paste the following code from PDF to start.)

```
#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(16,16,256,256,0);

    // Fill your code here.

    while(FSKEY_NULL==FsInkey())
    {
        FsPollDevice();
    }

    return 0;
}
```



Running image

PS8-2 Line drawing (70 points)

Download shape1.txt, shape2.txt, shape3.txt, and shape4.txt from Canvas. Content of each file is as follows:

```
1st line: Number of lines
2nd line: x1 y1 x2 y2 r g b   (line coordinates and color)
3nd line: x1 y1 x2 y2 r g b   (line coordinates and color)
                 :
                 :
```

Y-coordinate in the data file is taken such that the bottom of the window is zero. Therefore, you need to flip the Y-coordinate upside down to get a correct picture.

Your assignment is make a **class** that reads the file content, and draw lines on the screen. The picture will fit in the 800x600 window. For full credit, your class must be safe for copy, i.e., it must have a copy constructor and copy-assignment operators which are free of memory leak and safe for self copy. Also, const qualifiers must be used all appropriate places. As we discussed in class, some const qualifiers are optional. No points will be taken off for missing of having those optional const qualifiers. But, points will be taken if you have const qualifiers where it must not, or not having const qualifiers where it needs to be.

Write a C++ program that takes a file name (one of the input text files) as input from the user, and show the picture described in the specified text file. You can use a console input or InputString function to take file name. Your program must include a class that reads and stores the content of the file. The class must have at least two member functions (1) reading a file, and (2) drawing the picture. Assume that someone else may want to use your class in the future. Therefore the class must be free of memory leak and must not crash due to memory exhaustion. The program must close when the user presses the ESC key.

Since the purpose of the assignment is to practice memory management, do not use storage class provided by the Standard Template Library (std::vector, std::list, etc.)

The file name of the C++ program must be ps8-2.cpp and must be included in the Zip file.

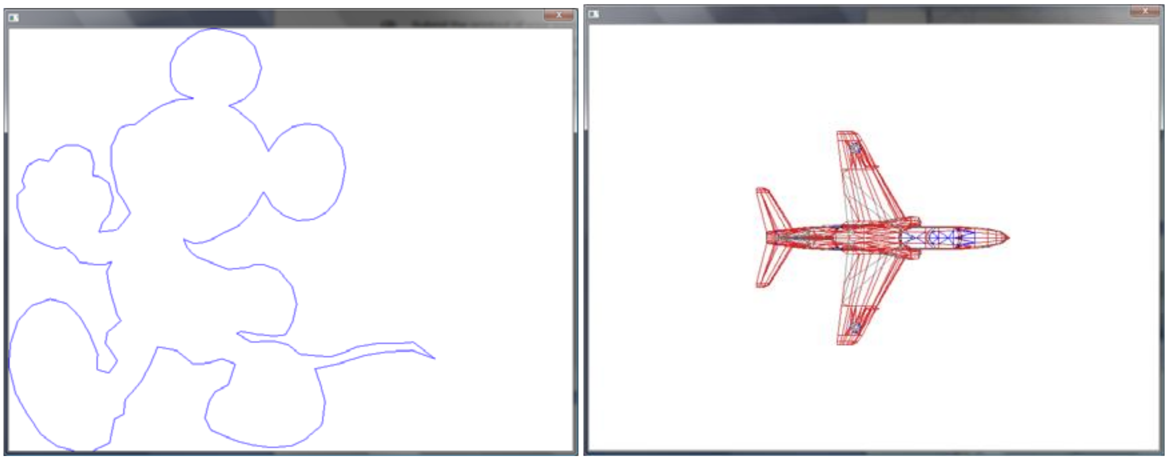
Start from the base code on the next page. You can add your code to the base code, but your code needs to be written so that it tests your copy constructor and copy-assignment operator.

2 extra points if you are able to move the picture by mouse. It should be done by (1) pressing the mouse left button anywhere on the graphics window, (2) moving the mouse cursor while the left button is held down. The moving should stop when the user releases the left button. The displacement must be exactly same as the displacement of the mouse cursor. The picture must move back to the original location when the user presses the SPACE key.

3 more extra points if you are able to zoom (1.1 times larger) by pressing Z key and unzoom (1.1 times smaller) by pressing M key. It must not shrink smaller than 0.1 times the original size. The scaling must be reset (back to the original size) when the user resets the position by pressing the SPACE key. The scaling must be calculated from the center (average of all coordinates).

Base Code

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

// Define your own Drawing class.

char *MyFgets(char str[],int limit,FILE *fp)
{
    if(nullptr!=fgets(str,limit,fp))
    {
        int l=0;
        for(l=0; 0!=str[l]; ++l)
        {
        }
        for(l=l-1; 0<=l && ('\n'==str[l] || '\r'==str[l]); --l)
        {
            str[l]=0;
        }
        return str;
    }
    return nullptr;
}

Drawing LoadFile(const char fn[])
{
    Drawing d;
    if(0==d.Load(fn))
    {
        printf("Cannot read the file.\n");
        exit(1);
    }
    return d;
}

int main(void)
{
    char fn[256];
    printf("Enter File Name>");
    MyFgets(fn,255,stdin);

    Drawing load=LoadFile(fn),drawing;
    drawing=load; // Force using operator=

    FsOpenWindow(16,16,800,600,1);
    for(;;)
    {
        FsPollDevice();
        auto key=FsInkey();
        if(FSKEY_ESC==key)
        {
            break;
        }
        glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
        drawing.Draw();
        FsSwapBuffers();
        FsSleep(10);
    }
    return 0;
}
```



Running images