CPSC 1070
Lab Project
Oct. 28, 2019

Introduction

This is a one day project to be completed in lab on Monday. If you have not finished by the end of lab, you can turn it in using the `handin` system by Friday. I you use `handin`, remember to use your lab section for CPSC 1071-001, or 002, or 003, not your class section. The TA will grade you on what you have either shown during the lab or in `handin`.

Your task this week is to write a C++ EZ Draw program, that reads image names from a file, loads the images, draws a 3x3 grid, and displays image textures in the grid squares.

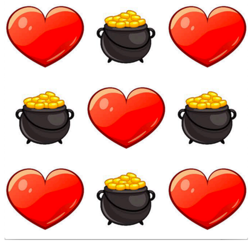
Getting set up for the problem

Download the zip file `grid.zip` to get the C++ version of EZ Draw that is linked on the Lab Schedule page of the course website for Oct. 28. Unzip this directory, to create a new directory named `grid`. Move into this directory and copy `ezdraw.h` and `libezdraw.a` to your `ezdraw` directory at the top of your directory hierarchy. The other files you should find in the grid directory are a `Makefile`, code files named `makegrid.cpp`, `Grid.h` and `Grid.cpp`, a file named `image-names.txt` and images named `heart.bmp` and `gold.bmp`.

The `image-names.txt` file contains the two filenames of the image files on two lines.

If you don't want to use the supplied images, just make your own images, move them into the `grid` directory and edit the `image-names.txt` file so it has the names of your BMP files. Your images should be square and 200x200 pixels in dimension.

The programming task



The program you are to complete should load the images named in `image-names.txt`, open a 600x600 EZ Draw window, create textures for the images, and display a 3x3 grid in the window that has the two images alternating like a checkerboard. Everything should be done in C++, using *iostream* I/O system, not the *stdio.*

The program `makegrid.cpp` and the `Makefile` are ready to use and should not need to be modified. But the `makegrid` program makes heavy use of the `Grid` object, so it will not work unless the `Grid` object is defined properly. The declaration of `Grid` is complete in the file `Grid.h`, so you know the internal data structure of `Grid`, and the calling sequences of all of its methods, but you need to complete the definition code for `Grid` in `Grid.cpp`.