C481 B581 Computer Graphics
Dana Vrajitoru

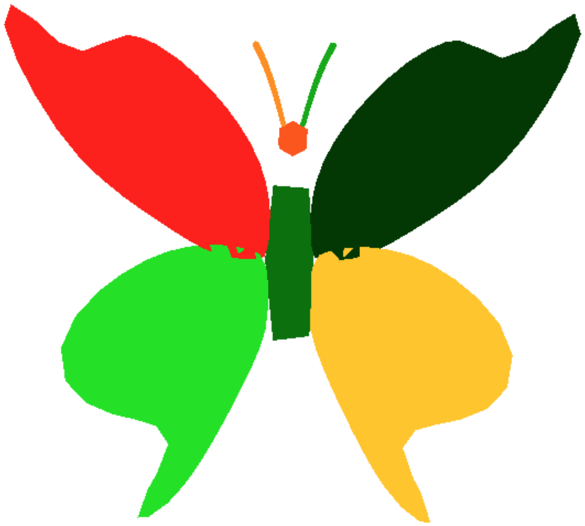
C481 B503 Homework 10

Due date: Monday, April 10, 2017.

In this homework we'll read an object made of triangles from a data file and draw it, then we'll perturb the color to create some procedural texture effects.

Ex. 1 Download the following files in a dedicated folder.
geometry.cc
geometry.h
glheader.h --> same as in homework 8
interface.cc
interface.h
Point3f.cc --> same as in homework 8
Point3f.h
main.cc
Makefile
butterfly.raw

Compile the program with the command `make` and run it with the command `triangles`. The arrow key translate the scene, PageUp and PageDown are used for zoom, and the left-right mouse buttons rotate the scene around x and y. Here is a snapshot of the program using the file `butterfly.raw` above after some scaling down:



a. Declare a global variable `matrix3f points` at the top of the file `geometry.cc`. Modify the function `loadRawFile` to store the points being read from the file in the variable `points` in addition to drawing them in the display list right away.

b. Add a function `draw2()` that draws the window using triangles and the points stored in the variable `points`. The basic color scheme for this one is up to you. You can use the one from the function `loadRawFile` if you prefer.

However, apply a perturbation to the color of each segment to generate a procedural texture. For this, you need to call the `glCcolor3f` function before each vertex and not only before each triangle set. You can multiply each component of the color with a random value between 0.7 and 1.3, for example. You can also try other schemes you can think of to find something that you think looks cool. You can have more than one new function `draw` if you find more than one interesting texture effect.

c. (Extra credit, 4pts) Add another function `draw3()` (or whatever number is next) where you compute the normal to the surface in each point. If a triangle is defined by the points P1, P2, P3, and we call A=P2-P1, B=P3-P1 (vector operations), then the normal N = A x B, the cross-product of the two vectors, and it applies to all 3 points of the triangle. You need to add a light source like in Homework 8 or 9 for the normals to take effect. Then you can specify the normal in each point while drawing the triangle strips. You can try perturbing the normal the way you perturbed the color in the previous function to see what the effect is.

d. Add global variable in `interface.cc` that lets us choose the option of the function `draw` that we want to use for the display. In the function `display` in the interface, use the value of this variable to call the corresponding function `draw`.

A**Ex. 2 (Extra credit, 3pts)** Create a file of your own that can be read by this program representing whatever object you want to create. Note that if you create an object with Blender and you export it as a `Wavefront (.obj)` object (such as key.obj), then you can extract from that all the lines marked with `v ...` and those should be the coordinates of the vertices composing the object (such as key.raw). However, they are not organized in triangles. Here are some explanations on the .obj files: http://paulbourke.net/dataformats/obj/

Upload to Canvas: all the files that you modified/added.