CS 543, Fall 2013 Homework 5, Due by class time (6PM) on Tuesday, December 3, 2013

Homework 5 Overview

In this project you shall implement some of the image manipulation operations we discussed in class. The following [Visual Studio solution] contains a working program that reads in and displays a bmp file usain_bolt.bmp. Your task will be to program various GLSL fragment shaders that implement various operations that we learned in class and control the effects interactively using keystrokes. To start off, download the Visual Studio Solution, compile it and run it. I have tested it and it works in the Zoolab.

To read in the usain_bolt.bmp file, the program uses use the [libbmpread library] which is a tiny, fast bitmap (.bmp) image file loader. Specifically, libbmpread is implemented as two files bmpread.c and bmpread.h which have been included in the Visual Studio starter code. bmp file loading functions and data structures are then used in the starter program to load the bmp file.

The Assignment

You should implement some of the image-manipulation operations previously discussed in class. Your program behavior should be as follows.

User hits:

- **Key 'O':** Display original picture (usain_bolt.bmp)
- **Key 'L':** Display picture with Luminance effect (gray scale)
- **Key 'N':** Display image negative
- **Key 'D':** Display image with edge detection effect
- **Key 'E':** Display image with embossing effect
- **Key 'T':** Display image with Toon rendering
- **Key 'W':** Display image with Twirl Transformation
- **Key 'P':** Display image with Ripple Transformation
- **Key 'S':** Display image with Spherical Transformation

Submitting Your Work

Make sure to double-check that everything works before submitting. Submit all your executable and source files. Put all your work files (Visual Studio solution, OpenGL program, shaders, executable and input files into a folder and zip it. Essentially, after your project is complete, just zip the project directory created by Visual Studio. Submit your project using web-based turnin.

Create documentation for your program and submit it along with the project inside the zip file. Your documentation can be either a pure ASCII text or Microsoft Word file. The documentation does not have to be long. Briefly describe the structure of your program, what each file turned in contains. Explain briefly what each module does and tie in your filenames. Most importantly, give clear instructions on how to compile and run your program. MAKE SURE IT RUNS before submission. Name your zip file according to the convention *FirstName_lastName_hw5.zip*

mailto:emmanuel@cs.wpi.edu