Worcester Polytechnic Institute

Computer Science

CS 543, Fall Semester 2013 Homework 4, Due by class time on Tuesday, November 19, 2013

Homework 4 Overview

In this project you will add more realism to your park scene (with L-System trees) from homework 3, by adding texturing, shadows and fog

Homework 4 Preparation

- **Make a slight change to the lookAt function in your current mat.h:** The lookAt function that you currently have in mat.h may have slight artifacts if you try to build on it. Specifically, you should change the following two lines from:

  ```
  vec4 u = normalize(cross(up,n));
  vec4 v = normalize(cross(n,u));


  to:

  vec4 u = vec4(normalize(cross(up,n)),0.0);
  vec4 v = vec4(normalize(cross(n,u)),0.0);
  ```

- **Understand how to load textures using the libbmpread library:** In this project you shall load textures in the bmp file format and use them to texture parts of your scene. The following [Visual Studio solution] contains a working program that reads in and displays a bmp file usain_bolt.bmp using the libbmpread library. 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. Study this example focusing on how the bmp files are being read in. You can also get more documentation on the libbmpread library website [Here] .

- **Reduce scene from 5 to 2 trees to increase rendering speed:** Some students may have very slow forest scenes from homework 3. To improve speed, reduce the number of trees rendered in your park scene from 5 to 2 trees, PLY files and a ground plane. If you did not get the trees working in homework 3, just create a test scene with 3 PLY files of your choice (including a car PLY file) and place them on a ground plane. If you are unable to create a test scene with 3 PLY files on a ground plane, you will receive a grade of 0 for this homework.

Homework 4 Specifics

The specific features you shall implement in this homework are:

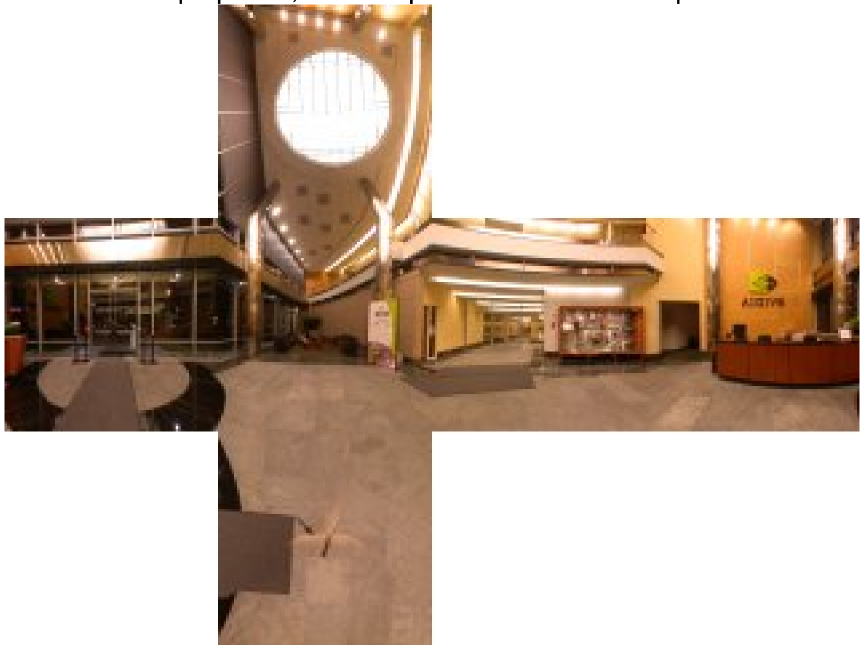
- **Texture the floor plane with grass or stone:** Texture map some grass onto the floor of your scene to make it look like grass is growing. Use the following grass texture [grass.bmp] or make it look like the floor is paved with stone by texturing using a stone texture [stones.bmp] Use the libbmpread library above to read in your bmp texture files. Do not try to stretch these small textures over a large floor. Map each texture to the bottom left corner of the floor undistorted and then use the appropriate parameters to repeat the texture to cover the entire floor. The example program in Appendix A.8 on page 638 of your text gives the complete working code of a program that does texturing.

- **Shadows:** Section 4.10 of your textbook describes a simple technique to render shadows using projection. Implement this shadow algorithm such that the shadows of the trees and meshes are projected onto the floor plane. The shadow of the entire scene should be updated as the camera moves.

- **Fog:** Implement fog as described in class.

- **Environment Mapping:** Add in reflective and refractive environment mapping using cubemaps to your scene. The example in section 7.9 (page 393) of your text presents a complete working example of an reflection map. The class slides present how to implement refraction. Use the following cube map with its [cube map sides in 6 images] as your environment map

  For reference purposes, the complete environment map looks like this:



Summary of Your program behavior

Control your scene using the following keystrokes.

- **Key D:** Toggle shadows ON/OFF. When ON, the shadows show up and when OFF the shadows do not show up.

- **Key A:** Toggle ON/OFF between a grass texture and stone texture. When ON, the floor is textured with grass and when OFF the floor is textured using the stone texture.
- **key F:** Toggle between fog that changes linearly with depth and fog changing exponentially with depth
- **Key T:** Toggle reflection ON/OFF. When ON, all PLY objects are drawn with reflection. When OFF, the PLY objects are drawn with no reflection (same as you had in HW3)
- **Key V:** Toggle refraction ON/OFF. When ON, all PLY objects are drawn with refraction. When OFF, the PLY objects are drawn with no refraction (same as you had in HW3)
- **Key K:** Toggle drawing PLY files with a mixture of reflection and refraction ON/OFF. Randomly select which PLY objects to make reflective or refractive. Since this choice is randomized, every time you toggle the K key ON, a different set of PLY objects are reflective vs refractive. When toggled ON, all PLY objects are drawn with either reflection or refraction. When OFF, the PLY objects are drawn with no reflection or refraction (same as you had in HW3)

Notes: No OpenGL fixed function commands (glBegin, glVertex, etc) or immediate mode drawing commands should be used in your program. All drawing should be done using shaders, retained mode, Vertex Buffer Objects, and glDrawArrays similar to the code in your textbook (and in your previous projects)

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_hw4.zip*