CSc 155 – Advanced Computer Graphics
Fall 2015   Scott Gordon

Assignment #4 – Shadows and other Advanced Techniques

Due Date: Friday, December 11th [3½ weeks]

In this assignment, you will implement a 3D scene that incorporates shadows (through shadow-mapping), and a selection of other techniques of your choice, such as bump/normal/height mapping, the use of tessellation or geometry shaders, and/or atmospheric effects. You are free to reuse your work from A2 and A3, or you can draw an entirely different scene.

Required Elements

The program must retain the following features from A3: at least three objects, camera control, materials, a positional light that can be moved around, and at least two textures. You are not required to retain instancing, clipping, or animation. You are free to keep or change the implementation of any of the features from the way you did them before, as long as each capability is present.

In addition, your program is to include the following new features:

- **Shadow-Mapping**

  Use the standard shadow-mapping technique, with either the default framebuffer or your own framebuffer. You must make a reasonable attempt to reduce artifacts such as shadow-acne.

- **plus, any FOUR of the following six choices:**

  - **Normal/Bump Mapping** (procedural or with a texture)

    You may derive your bump maps procedurally, or create a normal map image. If you use an image that was created by someone else, be sure to give credit in the code and in your report. Half credit for height mapping in the vertex shader.

  - **Tessellation shader and height map**

    Use a tesselation shader to produce a large grid and then combine it with a height map and a color and/or a texture. You can use it for terrain or for some other purpose. Although it is desirable to adjust precision based on distance, this isn't a requirement.

    or - **Cubic Bezier surface via Tessellation**

    Create a model that includes a cubic bezier surface, generated using a tesselation evaluation shader and a control shader. The surface should be solid (GL_FILL) and colored or textured.

  - **Primitive modification via Geometry shader**

    Alter one of your models using a geometry shader. Your alteration can take the form of adding new primitives, deleting some of the primitives, or altering some of the primitives.

  - **Fog** *or* **Transparency**

  - **Environment Mapping** *(such as for generating a mirror object)*

  - **Noise** *(such as for generating clouds)*

Additional Notes

- Please include a short (1-page) *REPORT* describing what features you have implemented. Also describe any features that your code attempts to implement but are incomplete (that will help me give partial credit). The Report should also give clear instructions on how to run and recognize all the features of your program. It is preferable that the report be in PDF format.
- It isn't required that every feature you include be present simultaneously in your scene. You can instead use buttons or other controls to show the features, or alternate between them. However, that isn't required – it usually is more interesting if they are all present at once.
- As for A3, try to design your graphical elements in a way that is logical and makes visual sense.
- As with previous assignments, your program must be contained in a package whose name is exactly "**`a4`**" (lower case). And, as before, it must be possible to run the program from a command prompt by typing the command "**`java a4.Starter`**". Please ensure (1) that the submitted ZIP file includes the correct package directory hierarchy, (2) that you have included all required files (e.g. texture files, bump maps, etc.), and (3) that the appropriate files are referenced only using *relative-paths*.
- All code you submit must be strictly your own (except that you are free to use code provided in the Lecture Slides, or posted on the course SacCT content page).
- Grading will be as follows:
  - ➢ 25% for shadow-mapping,
  - ➢ 40% for the four remaining requirements (10% each).
  - ➢ 15% for other existing requirements (models, textures, camera control, lighting, etc.)
  - ➢ 5% for the readme document
  - ➢ 5% for code organization
  - ➢ 10% for submitting on time.

Deliverables

Submit to SacCT, a *single* ZIP file containing (1) all Java source code and GLSL shader code, (2) compiled (.class) files, in the proper subdirectory **`a4`**, (3) any data files needed to run your program (e.g. texture maps, height maps, etc.) in the proper folder(s), and (4) your [.pdf] report describing the features and use of your program, as outlined above.