Math 155B - Introduction to Computer Graphics – Spring 2020

Instructor: Sam Buss, University of California, San Diego

Final Project, Create an individual project

Due date: Friday, June 5 (recommended), Wednesday, June 10 (also fine)

Project plans must be discussed with Professor Buss later than **Tuesday, June 2.** If you discuss the project in depth with Jon, a less in-depth discussion with Professor Buss is OK, but still, you must discuss it with Professor Buss.

Projects may be graded upon completion.

Goals: Design and create a significant OpenGL or RayTrace program. Create a PDF file documenting your project. Evaluation will be based on technical and artistic merits. One-on-one grading with Professor Buss and/or Jon Pham; in addition, an executable with all accompanying files needed to run the program must be uploaded to the google drive shared folder, along with your most significant source code.

What to hand in: Grading will be individual by zoom as usual. Turn into the shared google drive:

1. **A PDF document describing your project.** Include your name, but not your PID. Unless you request otherwise, it may be shown to the class during the project demo session during finals week, and anonymously in future years. This should be 1-3 pages in length, and include **(a) A paragraph or several paragraphs describing what is included in your project. (b) Documentation explaining how to run the program, including keyboard controls, etc. (c) One or more pictures showing your scene.** To create images, you use may use CNTL-ALT-PRINTSCREEN. You may also use RgbImage to export high-quality bitmap files, from either OpenGL or from the RayTrace project.
2. **An executable file and any necessary texture files or .glsl files.** Professor Buss and Jon Pham should be able to run your code inside this folder. Check this works on your computer in a separate folder before doing the upload. (Please!) Do NOT include files generated by Visual Studio.
3. **Your significant source code.** Do not upload project files, solution files, or files you did not modify.

We will schedule towards the end of final exam week to demo everyone's projects to the rest of the class.

INSTRUCTIONS:

1. **Pick a project of your choosing.** Guidelines for this include:
   a. Your program must use the RayTrace code or OpenGL or GLSL in the spirit of this course.
   b. You should spend approximately 15 total hours of work on the final project. Once you find yourself spending over 10 hours, please think about how to complete a good project within a reasonable time.
   c. Design a project that can be implemented in stages, so if you get stuck on one part and cannot complete everything as planned, you will still have a project to demo!
   d. Grading is based on technical merits, artistic design, and creativity.

   e. Your project should not be an adaptation of code from outside Math 155A/B (e.g., downloaded code). In some cases, we can give an exception to this, but only if your project includes a substantial extension of the other code and only if you discuss this with Professor Buss ahead of time.
   f. Some suggested projects are listed below under "4.". However, you are encouraged to be creative in designing a project.
2. **You must discuss with Professor Buss your plans for the final project by Tuesday June 2, and optionally with Jon Pham.**
3. **Turn in the items listed above.**
4. **Some suggested project areas include:** (see also last quarter's suggestions for final projects).
   a. **A RayTrace scene.**
   b. **Build a scene with animation.** A few sample ideas.
      i. Use quaternions or B-splines/Bezier curves to control motion or orientation. Try to combine this with an interesting scene.
      ii. Extend the Glsl water waves project to include more features, e.g. include water in a scene and simulate reflection and/or refraction with texture mapping.
      iii. Make a simple video game. *Warning: lots of work!*
      iv. Make a movie by exporting a sequence of RayTrace images.
   c. **Try some new uses of Shader programs; or multi-pass OpenGL rendering. These would require learning some new features on your own.** You should be be sure to include original coding if you take one of these options. Possibilities include:
      i. Write a fragment shader that performs edge detection
      ii. Write a fragment shader for "cel" images (a kind of non-random dithering)
      iii. Write a geometry shader that uses edge-adjacencies to draw Catmull-Rom or Overhauser spline curves.
      iv. Other options could include shadow mapping, or learning how to use a compute shader.
   d. Or, be creative with your own ideas!
5. Grading is based on technical merit, artistic merit, and creativity.