CS 242 Final Project Proposal

Improved Ray Tracer & Interface

Alex Aulabaugh (aulabau2) - Moderator: Bei Zhang

1. **Abstract**
   1. **Project Purpose**
      To improve and extend the ray tracer built as a part of the CS 419 curriculum. (Sp2016)
   2. **Background/Motivation**
      Building the ray tracer in CS 419 was fun and challenging - but I feel as though I walked away with an unfinished project. My existing code is extremely sloppy, leaks memory everywhere, and doesn't support a number of possible features that could take it to the next level. It also lacks tools to dynamically create scenes without changing the code.

2. **Technical Specifications**
   1. **Platform:** Windows, Linux, MacOS
   2. **Programming Languages:** C++, Java
   3. **Stylistic Conventions:** camelCase naming conventions, function signature commenting
   4. **SDK:** None
   5. **IDE:** Eclipse
   6. **Tools/Interfaces:** libpng, Swing
   7. **Target Audience:** Those who are interested in rendering cool images

3. **Functional Specifications**
   1. **Features**
      - All existing features from CS 419 including
        - Support for planes, spheres, triangles
        - Orthographic & perspective styles
        - Anti-Aliasing
        - Uniform Grid acceleration structure
        - Point and area lights
        - Image textures
        - Reflective surfaces
        - Refraction
        - Phong shading
        - Triangle meshes
      - KD Tree Acceleration Structure
      - Support for general polygons
      - Support for Toruses
      - Swing interface to define scene and run tracer
      - Display rendering statistics
   2. **Scope of Project**
      Application is limited to existing obj files and textures.

4. **Timeline**
   1. **Week 1 - Rewrite Ray Tracer**
      1. Re-familiarize myself with ray tracing principles.
      2. Using existing code as a guide, create entirely new tracer with better organization and fewer memory issues in Java.

    1. Implement orthographic rendering
    2. Implement support for basic geometry
    3. Implement phong shading
    4. Implement perspective rendering
    5. Implement image textures
2. **Week 2 - Improved Acceleration Structures, More Geometry**
    1. Implement at least the KD Tree acceleration structure, something I was unable to complete in CS 419.
    2. Implement support for at least toruses and general polygons.
    3. Continue to flesh out new tracer
        1. Implement reflection & refraction
        2. Implement area lights if not done
        3. Implement anti-aliasing
3. **Week 3 - Interface**
    1. Build Swing wrapper for ray tracer.
    2. Allow user to specify geometry from GUI.
    3. Allow user to specify ray tracer settings from GUI.
    4. Allow user to save scene settings.
4. **Week 4 - Loose Ends, Additional Ray Tracer Features**
    1. Complete all unfinished work from previous weeks.
        1. Implement triangle meshes if not done
    2. Allow user to load Textures from GUI.
    3. Allow user to load Triangle Meshes from GUI.
    4. Flesh out all features from CS 419.
    5. Use Ray Tracing From The Ground Up (http://www.raytracegroundup.com/) to implement more ray tracing features.
    6. Render some cool sample scenes.

5. Future Enhancements

More features can be added to the GUI, such as a preview of the scene based on shooting a few dozen rays before committing to the render. More acceleration structures could also be supported.