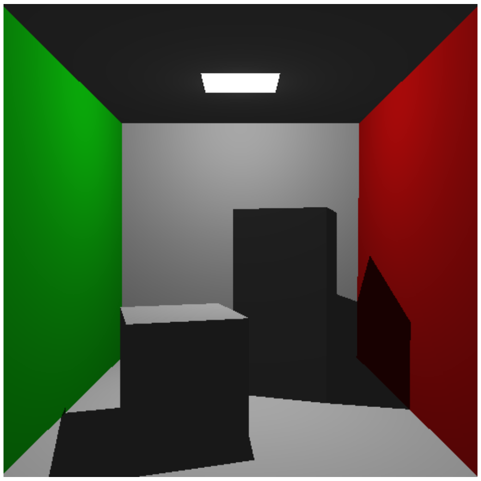
Program 4

Parallel Hardware Ray Tracing

Due: 11:59:59 PM, October 6, 2011

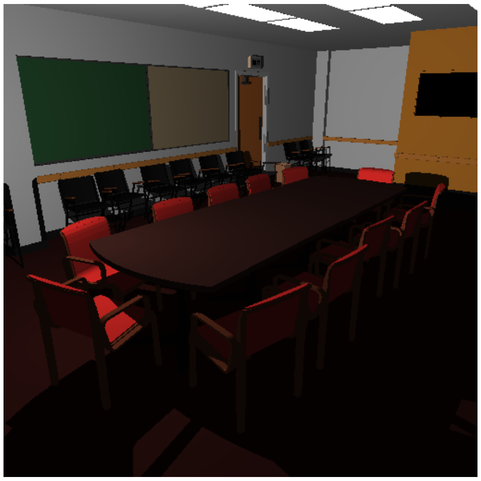
Before you start, make sure to 'svn up' and make the simulator directory.
In this assignment you will implement bounding volume hierarchy traversal.

1. **Part 1** (20 points): The BVH is automatically built and loaded in TRaX global memory, starting at the location pointed to by `start_scene = loadi(0, 8)`. See the lecture 8 and 9 slides (Sept 15, 20) for details. Adapt your program 3 ray tracer so that instead of looping over all triangles in memory, it traverses the BVH, and render the Cornell scene using BVH traversal:

   Use the simhwrt command-line argument `--write-dot <depth>` to help you see the structure of the BVH while debugging. Render this scene using the BVH in the TRaX simulator, and save the output to a file, linked to your handin web page.

2. **Part 2** (60 points): Render the conference scene with the BVH. If everything form part 1 works correctly, nothing about your ray tracer should have to change, and this should be a simple matter of loading a different scene:

   The ambient light color is [0.4f, 0.4f, 0.4f], `Kd` is 0.7f, `Ka` is 0.3f for every material.

To change the scene loaded by `run_rt`, edit the Makefile, and comment out these three lines:

```
#VIEWFILE=\"${SIMDIR}/views/cornell_obj.view\"
#MODELFILE=\"${SIMDIR}/test_models/cornell/CornellBox.obj\"
#LIGHTFILE=\"${SIMDIR}/lights/cornell.light\"
```

Then uncomment these three lines:

```
VIEWFILE=\"${SIMDIR}/views/conference.view\"
MODELFILE=\"${SIMDIR}/test_models/conference.obj\"
LIGHTFILE=\"${SIMDIR}/lights/conference.light\"
```

To render this scene in the TRaX simulator, use the following command:

```
./simhwrt --load-assembly <path to your assembly>
--model test_models/conference.obj
--view-file views/conference.view --light-file lights/conference.light
```

Required TRaX output

In this assignment, we finally have a somewhat interesting ray tracer. We want to see how it performs using a full set of TRaX resources. Once you have verified that your BVH traversal works correctly, run the simulator again with the following command, and link the output to your handing web page:

```
./simhwrt --load-assembly <path to your assembly>
--model test_models/conference.obj
--view-file views/conference.view --light-file lights/conference.light
--num-cores 20 --num-thread-procs 32 --num-l2s 4
--num-icaches 2 --num-icache-banks 16
```

3. **Creative Image** (20 points): Generate a creative image of your choice with the ray tracer you have built so far using any 3D model in obj format. A google search for "free 3d model" turns up some interesting results. We will also make available a repository of popular research models. Note that not all models come with a material file, which is fine, in this case just use gray scale shading (give every triangle the same material color). Also note that almost no models come with camera and light info. You will have to experiment by editing a view-file and light-file to place the camera and light to create the scene you desire.

4. **simhwrt output**: Link all output files mentioned above to your web page.

5. **Code listing**: Link your source code to your web page (preferably .tar). All of the code that you use should be included. You will not be graded on the quality of your code or comments, only on the presence of your source. We will verify that the code you hand in will produce the image(s) turned in.

6. **Time required**: How many hours did it take you to complete this assignment?

7. **Diffuculty**: What was the hardest part of this assignment, why?

You will not be graded on these last two items. They will be used to help improve the class in future assignments and in future years.

By midnight on the due date, you should send e-mail to teach-cs6965@list.eng.utah.edu with a URL for your assignment web page.