CSC 305 Assignment 3 – Final Project

Due Monday July 25th, 2016

Contents

Overview

In this assignment, you will put the skills you learned in assignments 1 and 2 to the test, and implement a more advanced ray tracer or real-time renderer. A selection of projects will be suggested. If you want to propose your own custom project, please meet with the TA (Nicolas) to decide on project deliverables and a grading scheme.

The suggested projects include 1 real-time rendering project, and 2 offline rendering projects. The real-time rendering project consists of implementing a procedurally generated virtual world, and the offline rendering projects consists of implementing photon mapping, or a better ray tracer.

Rules for Using Third Party Libraries

For this project, you are expected to produce an original work.

You may reuse third-party libraries for this project, but only for tasks unrelated to the purpose of the assignment. For example, you may use a vector math library (eg: glm) or an image loading library (eg: stb) or a window creation library (eg: SDL). In contrast, for example, you may not use a terrain generation library or a photon mapping library to implement the assignment.

In general, **rely only on pseudo-code and explanations in plain English**. In your software engineering career, this turns out to be an important practice for legal reasons.
See: http://nothings.org/stb/stb_opus.html

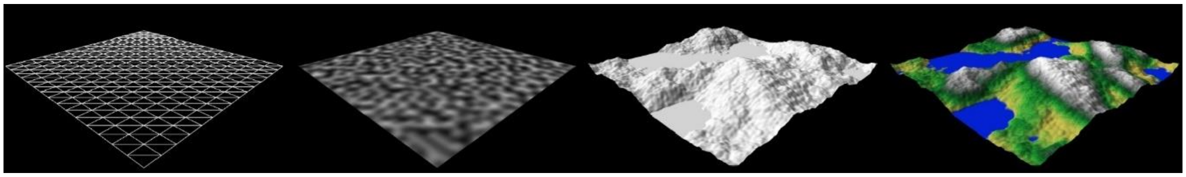
Publishing Results

Once the projects are done, you'll have the opportunity to publish your project on the http://gfx.uvic.ca website. You can submit a screenshot and/or a YouTube video link, and some notes about the implementation.

The projects will be posted here: http://gfx.uvic.ca/teaching/icg/2016/summer/index.md

If things go well, we could do a class share-and-tell session too, to show our work.

Project 1: Virtual World (Real-Time Rendering)



You will use **procedural methods** to generate a virtual world. This goal can be subdivided into three parts:

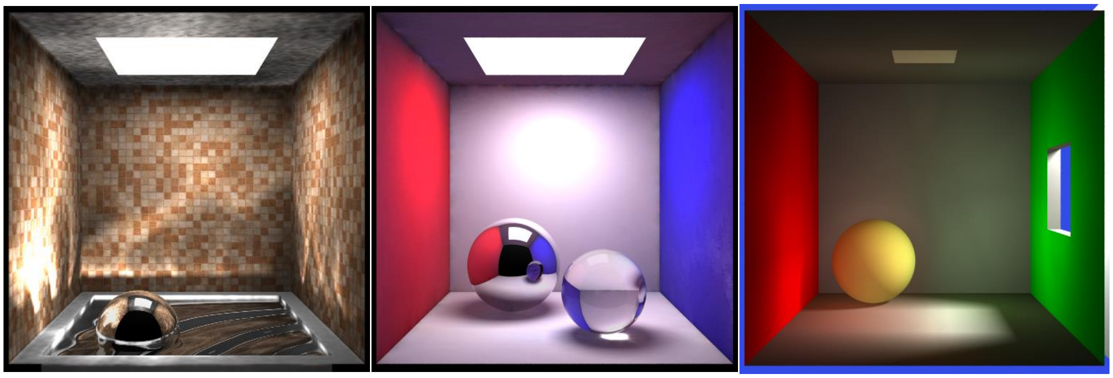
- geometry (procedural generation on either CPU or GPU)
- rendering (terrain shading, texturing, deferred rendering)
- animation (camera animation, particle systems, etc..)

The sub-tasks in these three parts are subdivided into two categories: basic and advanced. Tasks in the Basic category will allow you to obtain a **B-**. The advanced tasks can be used to improve your grade and **difficulty** is annotated in parenthesis beside each item. A real time world having a visual quality as shown in the teaser image above will award you an **A+**.

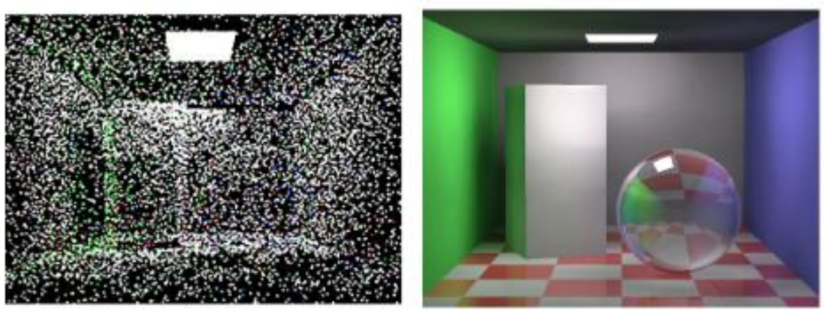
You can find the detailed assignment specification on this page:

https://github.com/ataiya/icg/wiki/Assignment%233a-Virtual-World

Project 2: Photon Mapping (Offline Rendering)



Examples from http://web.cs.wpi.edu/~emmanuel/courses/cs563/write_ups/zackw/photon_mapping/PhotonMapping.html



(b) 10,000 photons $r = 2.0$

Examples from http://www.cs.mtu.edu/~shene/PUBLICATIONS/2005/photon.pdf

You will use photon mapping to render a Cornell box with effects like caustics, light bleeding, and participating media. This goal can be subdivided into three parts:

- ray tracing (implementing an efficient ray tracer with enhanced sampling)
- photon mapping (implementing efficient storage/lookup of the photon map)
- lighting (implementing new lighting effects using the photon map)

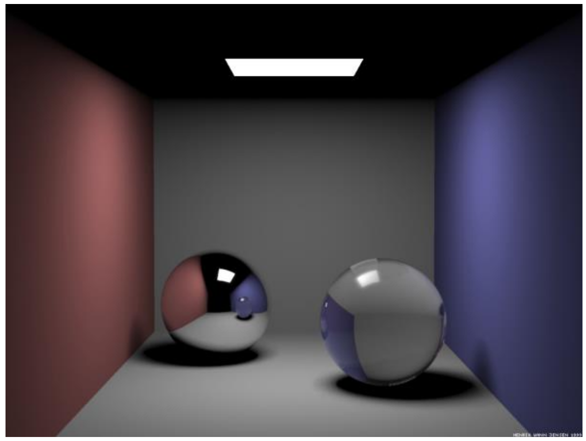
See the "References" section at this end of this article for good reading material on the topic of photon mapping.

The sub-tasks in these three parts are subdivided into two categories: basic and advanced. Tasks in the Basic category will allow you to obtain a **B-**. The advanced tasks can be used to improve your grade and **difficulty** is annotated in parenthesis beside each item. Renderings having a visual quality as shown in the teaser image above will award you an **A+**.

You can find the detailed assignment specification on this page:

https://github.com/ataiya/icg/wiki/Assignment%233b-Photon-Mapping

Project 3: Ray Tracer Enhancements (Offline Rendering)



This project is a subset of project 2 that focuses exclusively on improving your ray tracer, rather than implementing photon mapping. For this assignment, please add the following improvements to your ray tracer from assignment 1:

Modelling

- Perform simple object-space transformations of your object, moving the spheres around with transformation matrix calculations, or better a scene graph.
- This can also include the instancing method outlined in class. For example, to produce an ellipsoid from a primitive sphere.
- Include a triangle mesh as a model to be rendered.
- Implement CSG between simple half-space primitives.

Rendering

- Create a cube room with a cube, a matte sphere, a reflective and a transparent refractive sphere in it. Include an area light source with a relatively large surface area, and render a soft shadow effect from the objects.
- Using jittered anti-aliasing to enhance the quality of your rendered image.

Efficiency

Accelerate the ray tracer by using either:

a. Uniform space subdivision.
b. Bounding spheres or hierarchical bounding spheres.
c. KD tree or any other scheme.

Benchmark your ray tracing program: How much time did your optimization save for your ray tracing program? What is the relationship of the time saving and the complexity of your scene?

Feel free to propose your own features but talk to the instructor or the TA first!