

# CS535: Final Assignment

**Out:** October 29, 2024

**Back/Due:** last week of classes

## Objective

Now that you have an understanding of some fundamental components of interactive computer graphics, the focus of this final assignment is to give you the freedom to extend your knowledge in one particular area of interest to you. Below are several suggested individual projects. Alternatively, you may also suggest your own project – if you propose a team project, the workload should be commensurate with the increased project size; teams of two people are possible; teams of three people will rarely be approved unless the project proposal is significantly well organized and ambitious. All projects will be demonstrated to a public audience on the due date. The demo includes a live presentation of your project/program and a short PowerPoint presentation explaining how it works. The time slot is to be determined but roughly 10 to 15 minutes per person.

**\*\*\* Please send me an email by November 4<sup>th</sup> indicating which project you will be doing. PLEASE MAKE THE EMAIL TITLE “CS535 Final Project” – I repeat please make the email title “CS535 Final Project” – I get lots of emails and it will be lost if you make your title something random like “app”.**

**\*\*\* If you are proposing a self-project, please do so in writing with a short one-page project proposal before Nov 4<sup>th</sup> – this way I can get you feedback and all projects can be setup by November 4<sup>th</sup>.**

**\*\*\* If you do neither of the above, I will randomly assign a project to you.**

As a prelude to choosing a project, I recommend you browse through the papers (mostly PDFs) at <http://kesen.realtimerendering.com/>, in particular for the SIGGRAPH USA/SIGGRAPH Asia, Eurographics, and Symposium on Interactive 3D Graphics conferences. Browsing through the papers might inspire you with regards to options. Most, if not all, of the below projects have previous works published at the aforementioned venues. Depending on your interest and project objectives, I can point you to several specific papers which you can read for background information.

Below are guidelines for the projects – I encourage all to go above and beyond the descriptions given. Grading will be based on how well your *implement* the project and *demonstrate* it (live demo and PowerPoint presentation), your *understanding* of the general concepts, and the *features and functionality* of your program. In other words, if you do something but don't know how it works or how to show it off, you get serious points off. Furthermore, you must implement the project in its entirety and no downloaded or previous implemented code is permitted unless explicitly approved by the instructor (I am not trying to discourage using software libraries but rather trying to prevent you from getting credit from what someone else implements ☺).

During November 11-15 week, each of you will describe briefly, via a PowerPoint presentation in class, what your project is and will give a summary of the project status so far (i.e., mid-project review), including a summary of previous and related research literature. I *\*strongly\** encourage you to use this opportunity to get preliminary feedback so that there are no surprises on demo-day.

Your final assignment is worth 30% of your grade and the fast forward is 5%.

You may of course setup an appointment with me if you have questions.

This final assignment will take effort and I vehemently encourage you to **start working on the assignment well before the due date** – i.e. today! As with the previous assignments, I prefer a nicely working, well-implemented and demonstrated project as opposed to a half-working buggy project with lots of half-baked complex features.

## 1. Collision Detection

Collision detection is the art of detecting if two objects collide (or have already penetrated each other) and then performing some sort of action (e.g., a bounce). Exact collision detection for general objects is hard and time consuming.

The goal of this final assignment is to implement an approximate collision detection scheme using hierarchical spatial subdivisions. The assignment should have a large static scene (e.g., one of the previous models) and then allow one or more objects to bounce around the scene. The allowed complexity of the moving object and physical correctness of the bounce operation depends on the sophistication of your project. Your program should exhibit interactive performance.

A simple scenario is to create an octree data structure for the large static scene and to create a moving octree data structure for a moving object with an approximately convex shape. The collision algorithm is thus to intersect two octree data structures. If there is an intersection, you perform a bounce operation. The approximate nature of the algorithm comes from only intersecting octree nodes. You may also have more than one object bouncing within the scene.

Additional features include supporting angular velocity, more complex moving objects, and having a virtual “cannon” out of which the moving objects are shot out from – this allows for a basic game setup.

## 2. GPU Programming

This project is rather open and consists of developing a system that exploits GPU usage and programming. It can be a more sophisticated illumination algorithm (e.g., such as some form of Ambient Occlusion). It can be a physically based simulation (e.g., improving upon your previous water project). Or another idea.

### 3. Non-Photorealistic Rendering (NPR)

This project entails developing and implementing an NPR technique. The details can be found in the relevant papers. The tentative, but not all inclusive list, is the following:

- An overview: <http://www.red3d.com/cwr/npr/>
- Exaggerated Shading: implement something akin to the Exaggerated Shading by Rusinkiewicz et al. 2006  
[\[http://gfx.cs.princeton.edu/pubs/Rusinkiewicz\\_2006\\_ESF/exaggerated\\_shading.pdf\]](http://gfx.cs.princeton.edu/pubs/Rusinkiewicz_2006_ESF/exaggerated_shading.pdf). Some of these styles are rather simplistic so you will have to specify what techniques.
- Pen-and-Ink: implement some form of pen-and-ink using a precreated tonal map (e.g., textures of different density/darkness), Computer Generated Pen-and-Ink Illustration by Winkenbach et al. 1994  
[\[ftp://ftp.cs.washington.edu/tr/1994/01/UW-CSE-94-01-08b.d/UW-CSE-94-01-08b.pdf\]](ftp://ftp.cs.washington.edu/tr/1994/01/UW-CSE-94-01-08b.d/UW-CSE-94-01-08b.pdf)
- Point-based Stippling (or Stipple Drawing). See the work of Floating Points: A Method for Computing Stipple Drawings by Deussen et al. in Eurographics 2000.  
[\[http://www.cs.princeton.edu/courses/archive/fall00/cs597b/papers/stipple\\_deussen.pdf\]](http://www.cs.princeton.edu/courses/archive/fall00/cs597b/papers/stipple_deussen.pdf)

### 4. Procedural Modeling

This project is to develop a system for procedural modeling. You may choose one or more areas of focus such as procedural methods for plants, buildings, or terrain. The project should allow for a “grammar” to be written which can be then be converted to 3D geometry and rendered on the fly. The grammar is to be read, parsed, and applied. The resulting model is to be displayed using OpenGL and the 3D navigation tools you have developed in previous projects.

In class, we will talk about L-system grammars, shape-grammars, and split grammars. Your task is to choose a version of one of these grammar options and to explore from there what you can do.

For pictures and inspiration about plants, take a look at:  
<http://algorithmicbotany.org/papers>.

For the same about urban modeling, take a look at:  
<http://www.esri.com/software/cityengine>

<http://wiki.cs.purdue.edu/cgvlab/doku.php?id=projects:urban>

<http://www.iit.edu/~krawczyk/nks2003/>

If you are interested, I have particular papers I could refer you too. In general, this project is very flexible and the objective is for you to focus on creating a way to develop something visually interesting: write some simple text description and generate nice 3D content – don't forget to have fun!

## **Demo Day**

Final Assignment demo day is last week of classes. Closer to that date we will setup a schedule via a democratic algorithm. Essentially, we will put up department-wide announcements, provide food and snacks, and have a demo-fest. Each student will present and demonstrate her/his assignment. You must also provide on that same day a CD or USB key containing your source code, binaries, project and presentation – program must be executable from the CD/key. No extensions, no late penalties, no late passes, and no exceptions to this due date will be given.

If you have more questions, please ask!

Good luck!