

CMPT 485/829: Project Specification

Winter 2012

1 Purpose

The purpose of this project is to give you the opportunity to explore a computer graphics topic or algorithm, that you think might be interesting, in depth. The class time in this course is focused on a broad coverage of computer graphics, and the assignments are short. This project should give you some depth of understanding in a topic of your choosing.

2 Overview

CMPT 485 students are expected to work in groups of 2-3 students, and CMPT 829 students are to work individually. If the scope of a proposed CMPT 829 project is substantial, then permission may be granted to work in a group of two; permission must be obtained via email before commencing with the project.

This project is intended to be a significant undertaking. Each CMPT 485 student is expected to contribute approximately 40-50 hours over the course of the term to the completion of the project. CMPT 829 students are expected to double the approximate time-effort of CMPT 485 students. So, keep these time-estimates in mind when choosing your project, and begin work on your project early to avoid a time-crunch at the end of the term.

There are two project types available for you to choose from: an implementation, or a survey.

3 Implementation Project

The main deliverable in an implementation project is a piece of software that implements a computer graphics technique or algorithm(s). The implementation may be of a single algorithm if the algorithm is particularly involved, or requires a substantial amount of code be developed to support the algorithm (ex: a real-time ray tracer, or a global illumination algorithm for rasterized graphics). Alternatively, the implementation may be of several different algorithms that accomplish the same task in different ways (ex: texture generation, or shadow algorithms for rasterized graphics).

It is recommended that a proposed implementation projects have a series of incremental deliverables so that software that accomplishes something of some significance is produced in the event that your full proposal turns out to be too ambitious.

3.1 Deliverables

3.1.1 Project Proposal

The project proposal should be short, and to the point; a single page is a good approximate maximum length. It should outline:

- What you intend to implement;
- A rough time line of milestones indicating what will have been implemented by when;

- A list of resources (articles, web sites, book chapters, et cetera) that describe the algorithm(s) you intend to implement. URLs to the resources should be included where possible.

3.1.2 Status Reports

A moodle forum thread will be created for your group. In your thread you must provide at least one status report every 2 weeks after the project proposal has been submitted. These status reports should be short. They should indicate what has been accomplished since the last report, difficulties encountered, and contain screen shots of imagery created by your implementation-so-far when possible.

The intent of these status reports is to help you stay on your time line, give the instructor a sense of where you are on your time line, and to show off what you have accomplished to the other groups.

3.1.3 Software

The main deliverable for this project is the software that you create. This software must:

- Compile and run on the Linux machines in S360.
- Be self-contained – it cannot require installation of other software, or libraries on the system or in a user account. We may be able to arrange for some third party libraries to be installed on the S360 machines; if there is something that you want to use, but is not installed, then please ask.
- Include a Makefile, or other command-line build capability (ex: cmake or autoconf);
- Include a readme.txt file describing how to build and run the software, as well as a summary breakdown of which source file contains which aspects of the implementation.

You may use code made available by other developers provided that:

- You have okayed its use with your instructor.
- You have permission to do so (ex: open source licence) from the author. Note that you must adhere to the conditions of the code's licence.
- It is support for the main purpose of your project (ex: a library to load a mesh, or to write images to disk).
- You disclose its use and where you obtained it from both in your readme.txt file, and as documentation in your source files.

Basically, external code is likely okay provided that it does not defeat the purpose of your project, and that you are up-front and honest about its use. Be ethical, and use good judgement.

3.1.4 Written Report

You must submit a written report that does not exceed 10 pages in length. The font used must be 12 point, and either Times or Palatino. This written report must be organized into the following sections:

1. Introduction and problem description
2. Description of algorithm(s)
 - Brief summaries of algorithms employed. Should be enough detail to give the reader a rough understanding of the algorithm. Cite academic publications, or book chapters for in-depth detail.
3. Details of implementation
 - Must include a discussion of the structure of your code base: how you structured the relationships of your code (class/object, data, and algorithmic relationships)
4. Post-mortem
 - How well you adhered to your schedule, how much of your proposal you accomplished, difficulties encountered, interesting things learned, reflection of what would be done differently, et cetera.
5. References
 - A list of every reference (article, book chapter, web site, example source code) that materially contributed to completion of the project. Include URL to the reference where possible.

3.1.5 Video Demo

You must author a narrated video demonstration of your software. This video should be at least 3 minutes in length, and may absolutely not exceed 5 minutes in length. Time permitting, all video demos will be played for the class during the last two scheduled classes of the term.

Some instructions for how to capture a narrated screencast will be posted on the course Moodle in advance of the due date for the video demo.

4 Survey Project

The main deliverable in a survey project is a written survey report on a topic of computer graphics. The topic chosen should be broad enough that there are more than a handful of published sources on the topic, but narrow enough to facilitate a reasonably comprehensive discussion of the topic.

A survey paper examines the current state of the art of the topic as described in the literature. A survey paper typically provides a taxonomy or classification of different techniques used in the areas of the topic (if they exist) and briefly described, in the survey author's own words, several representative papers. The author may also offer commentary on the popularity, advantages, disadvantages, and overall effectiveness of different methods/algorithms/techniques. Insights into directions that future research into the topic should, or could, take are welcomed.

4.1 Scope

Your survey should include and discuss a minimum of 15 conference-length references for CMPT 485 groups, or 20 conference-length references for CMPT 829 students. Conference-length is approximately eight pages.

The intent of this project is to survey the state of the art on a topic, and thus the majority of your references should be for peer-reviewed publications from the past five, or so, years.

4.2 Deliverables

4.2.1 Project Proposal

The project proposal should be short, and to the point; a page or two is a good guideline. It should outline:

- A definition of the topic that you will research;
- Abstracts and citations for at least 3 different articles on the topic that cover different methods/algorithms/techniques;
- A short list of journals or conferences that publish articles on the topic;
- A rough time line of your planned work flow.

4.2.2 Status Reports

A moodle forum thread will be created for your group. In your thread you must provide at least one status report every 2 weeks after the project proposal has been submitted. These status reports should be short. They should indicate what has been accomplished since the last report, URLs to any particularly interesting articles read, et cetera.

The intent of these status reports is to help you stay on your time line, give the instructor a sense of where you are on your time line, and to share interesting finds with the other groups.

4.2.3 Written Report

You must submit a written report that does not exceed 15 pages in length. The font used must be 12 point, and either Times or Palatino. The format must be single-column, and be single-spaced.

This report should demonstrate understanding of the topic area, and the references discussed.

5 Some Example Topics Areas

These are just some possible topic areas, and is not intended to be a exhaustive list of all allowable areas:

- Real-time shadows
- Real-time global illumination
- Global illumination for ray tracing

- Real-time ambient occlusion
- Level of detail modelling/rendering
- Mesh decimation algorithms
- Mesh stripification algorithms
- Real-time ray tracing
- Acceleration structures for ray tracing
- Rendering participating media (real-time or ray tracing)
- Non-photo-realistic rendering
- Radiosity
- Texture synthesis
- Shader effects
- Procedural geometry
- Modelling natural phenomena
- Transitioning between key-framed animations
- Inverse kinematics
- Physics-based animation (fluids, smoke, rigid-bodies, et cetera)
- View/image morphing
- Rendering synthetic objects into real photographs

5.1 Resources

Some resources for finding topics, and information for topics:

- Ke-Sen Huang's catalogue of computer graphics publications:
<http://kesen.realtimerendering.com/>
- GPU Gems 1, 2, and 3:
Online at <http://developer.nvidia.com>
- Stanford's CS348B Project listings:
<http://graphics.stanford.edu/courses/#cs348b>

6 Deadlines

- **4:00pm Friday January 13, 2012** – Group formation
 - An email to the instructor, carbon copied to all group members, listing all participants in the group, indicating the type of project (implementation or survey), and suggesting at least one potential topic for the project.
 - Groups of one (CMPT 829 students) must also complete this step.
- **4:00pm Friday January 27, 2012** – Proposal submission
 - A verbal “go-ahead” on the proposed topic and scope should be obtained well in advance of this deadline.
- **11:59pm Sunday April 1, 2012** – (Implementation project only) Video demo
- **5:00pm Wednesday April 4, 2012** – Submission of remaining final deliverables

7 Submission Instructions

- All deliverables to be submitted via Moodle.
 - Written documents must be in pdf format, and software must be submitted as a ZIP or tgz archive.
 - Software must compile and run on the S360 lab Linux machines, and must be self-contained; it cannot require installation of additional software, or libraries into a user-account to compile and run.
 - Software must include a Makefile to build the software, and a readme.txt file explaining how to build and run the software.
 - Video must be playable on the S360 lab Linux machines.
- No late submissions will be accepted, except under extenuating circumstances. Late submission must be approved *prior* to the due date.

8 Evaluation

8.1 Implementation Projects

- Project proposal: **5%** – Based on clarity, quality of writing, and evidence of planning.
- Status reports: **10%** – Based on quality, and frequency.
- Video demo: **10%** – Based on clarity, quality, and thoroughness.
- Written report: **25%** – Based on clarity, spelling and grammar, presentation, and overall quality.
- Software: **50%** – Based on correctness, ease of use, readability, quality of documentation, quality of engineering, and scope.

8.2 Survey Projects

- Project proposal: **5%** – Based on clarity, quality of writing, and evidence of planning.
- Status reports: **10%** – Based on quality, and frequency.
- Written report: **85%** – Based on clarity, spelling and grammar, presentation/organization, overall quality, comprehensiveness, cohesiveness, and demonstrated understanding.