

Introduction to Computer Graphics

Project

My original due dates were too slack. Thus I am proposing a tightened schedule. However, in order to keep to advertisements when grading is concerned, marks will not actually be deducted if you do not meet the new “guidelines”. (The corresponding absolute deadlines are in parentheses). A substantive project is required from grad students, and you are strongly advised to attempt to meet the new guidelines for due dates.

Due: Oct 23 (none), Nov 3 (Nov 7), Nov 20 (Nov 28), Dec 9, Dec 11, 2003, 11:59 PM

Credit (U-grad): About 20 points (Relative, and roughly absolute weighting)

Credit (Grad): Approximately 30 points (Relative, and roughly absolute weighting)

Grad students and honors students have to do a project. In the case of grad students, it is expected to be a substantive project, ideally supporting some research. Non-honors undergrads who are interested can be treated as honors students in consultation with the instructor. Some additional concession regarding assignments may be possible in this case.

Projects may be done singly or in groups of up to 4 or so. Grads and undergrads can mix in projects. Projects may be proposed by the student, or some of the instructors suggestions can be used. In those cases there would be additional supervision and input from the instructor. So far we have the “fungus” project, the “browser” project and the “fluorescent surfaces” project. The “fluorescent surfaces” project will be co-supervised by the TA who has a research interest in this.

Any number of these projects can use the stereo display, and the project groups will share expertise learned using this tool. A demo is planned for Wednesday, October 15, 2:00. We will gather near the elevators on the seventh floor of Gould-Simpson 1:45, and then walk over to the Triestman center (Music, 137) at 2:00 sharp.

As these are research oriented projects, there are no limitations on using any OpenGL facilities, any code and/or tools available on the internet, or any ideas already published. A literature search is always a good idea. Someone may have already solved your problem better than what you were about to do. Best to work from that point. The idea is to go from the current state and do something new.

For most projects, it is expected that the platform is Linux. However, if there is a compelling reason to use another platform, then this should be argued for early on and settled by the proposal due date. Projects using the stereo display will need to compile and run on the specific machine used to drive that advice. It is highly recommended that you use some form of conditional compilation to make your code compile, run, and do something reasonable when used on machines with ordinary displays (such as the graphics machines). This will make development easier.

Time line:

Midnight October 22: (Choose partners, topic, and platform (0%))—I need to know by E-mail who you are working with, and topic or at least a topic area or direction . If you want to use special hardware (stereo display, graphics cards, etc.), or non-Linux platforms, this needs to be negotiated as soon as possible, so that you can adjust your proposal accordingly. Similarly, you should explain what research your project is intended to support (if it is not obvious as is the case with the projects proposed by me), so that any adjustments can be dealt with as soon as possible.

Noon Monday Nov 2 (Nov 7): (Proposal (10%))—I will need a proposal regarding what you are going to do (including a rough timeline), who you are going to do it with, and some details regarding what you hope to accomplish by the first review. The proposal can be send to me via e-mail. The new guideline date is set as it is because I will be away part of that week. If you get it to me by noon Monday then I will comment by Tuesday. Otherwise you may not get comments for about a week.

Midnight Nov 20 (Nov 28): (Review (10%))—I will need a short report. This could be handled in person, together with a subsequent E-mail summary of what was discussed and agreed upon. I will need to know how far you have gotten regarding your projection for this point in time, and what kind of problems you are running into. We can negotiate new final specs if necessary.

Noon Dec 9: (Demo/Presentation (10% plus some part of the 50% “ideas and execution” part of the grade))—Around this date a demo/presentation session will be set up. This is an excellent time to claim most of the “ideas and execution” part of the grade (the 50%).

Dec 11: (Proposed final due date for projects (rest of “ideas and execution” 50% and 20% for write-up))---You need to submit a project write-up explaining what you did and why, what you learned, and what someone carrying on the work should know. The exact form of the write-up is somewhat situation dependent and can be negotiated. For research oriented projects that may be picked up by someone other than yourself, it is critical that your hard work is not lost. This requires the above mentioned write-up, a readme file linking how to run the program, and how the code relates to the various things that are accomplished. The code itself should be well enough documented that someone else has some chance of extending it. If, on the other hand, the project is ongoing research that part of your group is going to continue, or if it is essentially finished, then some of the requirements here can be relaxed in favour for a more detailed write-up about the project

itself. Ideally it would be able to form the start of a research paper on the topic. If much of what you have done by the due date does not work, then a write-up explaining what you have tried and what you learned is critical.

Deliverables

You must electronically submit the project write-up described above, source code, Makefiles, a README containing any relevant information about how to build the program on the platform you used and how to run it on the various data sets as appropriate, and any relevant information connecting your project description to source code (e.g., “The first approach described in section 2.1 is implemented in the file *first.c* which contains code that is executed when the program is called with the *-first* flag”).

The turnin name is cs433_project.