COMS W4160— Computer Graphics Spring 2014

Final Project

out: Apr 29, Tuesday
due: 11:59PM, May 14

Finally, here comes the last programming project of this semester. We would like to have the final project as open as possible. You are encouraged to apply what you learned in this class to create any cool and crazy applications. In terms of programming language, both C++ and Java are allowed.

1 Getting Started

To get started, you can use the starter code or even your code of previous assignments as a base code. If you are reusing previous code, you must show very different results or significant improvements from your previous code. Please specify the differences or improvements clearly in your report. Starting from the scratch is also OK if you feel necessary.

Group Work You are allowed to work in a group or by yourself. Each group should consist of *at most* 3 people. For group submission, we expect to have significantly more involved work done than those from a single-person submission in order to earn the same level of grade.

2 Programming Requirements

Although this is an open project, you need to demonstrate what you learned about computer graphics and show the quality of your implementation. Here are specific goals your code should satisfy:

1. **Make a plan:** As a first step, youll want to come up with an idea or vision for what youd like to produce. If its a video game, think out the game play and come up with a simple design on paper, and think about what you need to implement in order for it to come together. If you are generating an animation, think about the animation scenarios and the key motion effects you would like to achieve. If you are improve the Monte Carlo, think about the specific effects (e.g., caustics) that can highlight your program. You need to *submit a description of your plans* in your project report. A good planning will also show in your finish results.

2. **Implement your plane:** Next you implement your plan. Both C++ and Java are allowed. For C++ programmer, if you are using external libraries, make sure they are either existing in CLIC machines or included in your submission. You should also provide a detailed step-by-step instruction to tell our TAs how to compile your code. For example, you can provide a CMake file or a Makefile. For Java programmer, please include any used external libraries (as jar files) in your submission. Please also provide instructions about what libraries are needed and how to compile the code. Like all the previous programming assignments, a failure of programming will lead some loss of points.

3. **Show your results:** You should submit the results to demonstrate your code. The results can be either a movie or a set of images (for rendering programs). As examples, if you are making a video game, include a movie to show the screen shots of your game; if you are generating animations, include a movie to show the animation; if you are creating rendering effects, include a set of images.

3 How Will It Be Graded?

Your grades will be based on a few criteria: 1) the clarity of your report, 2) the quality of your implementation, 3) the "awesomeness" of your results related to computer graphics, and 4) whether your code can be compiled successfully. Although different people will have very different submissions, we will keep our grading standards as objective as possible. That being said, we expect you to tell us how to appreciate your code and results. This can be achieved by highlighting your features in your report or annotating specific effects in your movies. You should think about your report and your movies/images as a presentation that can tell the audience your exciting job.

Submissions from a group will receive higher standards for grading. We expect the group submission to show higher implementation quality and demonstrate more complex results. In short, we will estimate the workload of your project, and expect it to be proportional to the head count of your group ($1 + 1 \geq 2$?)

4 Ideas for Projects

Before staring to code, you should have a clear idea about what you plan to do. Your plan needs to be practical and implementable in a short project. While we highly encourage creativity and imagination, here are a few ideas if you get stuck. Some of the ideas are similar to what we provided for the first programming assignment, but as a final project after a whole semester learning, we expect you to achieve more advanced effects.

- **A video game** that is fun to play, visually appealing with certain effects (e.g., soft shadows, textures, motion blurs, etc.). The user interface should also be more sophisticated compared with the PA-1 submissions.
- **Model something** that is familiar and visually interesting, such as a tree or a building, or a familiar kids toy. Model a meadow in central park, and see how many blades of grass you can draw and animate at one time. Model an alien world with strange creatures that move around, and have interesting behaviors. Model an underwater world, with interesting creatures and float around with them. Model artificial life in an aquarium. You can apply spline curves and other techniques learned in the class. You are encouraged to use your rendering code to show nicely rendered animations. If your project is focused on modeling, you are allowed to use any existing renderer (e.g., Indego, Mitsuba, etc.) to generate nice images. In that case, you should also include the screen shots of your programs to show the modeling.
- **Rigid body simulations** can be easily implemented. For example, you can show a few Lego pieces bouncing around on a ground or a bunny sliding down from a stair.
- **Basic fluid simulation:** If you like physics, you can implement a basic fluid simulation to show the natural motion of smokes or waters. Some details of fluid simulation can be found from the paper `http://dl.acm.org/citation.cfm?id=311548` or `http://www.cs.ubc.ca/~rbridson/fluidsimulation/`.
- **Implement a SIGGRAPH research papers:** You can find a interesting research papers availabe online (e.g., from `http://kesen.realtimerendering.com/`), understand it and implement their nice results.
- **Improve your renderer:** You can improve your Monte Carlo renderer by implementing different algorithms, such as dealing with ray refraction, achieving caustics by transparent objects, rendering a volume of materials (e.g., smokes and fogs).

5 Submission and FAQ

Submission Checklist: Submit your assignment as a zip file via courseworks. Your submission must consist of the following parts:

1. **Documented code:** Include all of your code and libraries, with usage instructions. Your code should be reasonably documented and be readable/understandable by the TAs. If the TAs are not able to run and use your program, they will be unable to grade it. Try to avoid using obscure packages or OS-dependent libraries, especially for C++ implementations. To ensure the TAs can grade your assignment, it is highly suggested to compile your code on CLIC machines, and include details of compiling and running your code on CLIC.

2. **Brief report:** Include a description of what youve attempt, special features youve implemented, and any instructions on how to run/use your program. Please *highlight specific features and effects* that you would like to tell us and briefly describe the algorithms you used to achieve them. If we couldn't notice your wonderful effects, how can we grade it? So it is important to emphasize them clearly in the report. In compliance with Columbia's Code of Academic Integrity, please include references to any external sources or discussions that were used to achieve your results.

3. **Video/Image highlight!:** Include a video that highlights what youve achieved. The video footage should be in a resolution of 960×540, and be no longer than 40 seconds. If it is better to highlight your code using images, please feel free to do so. The image resolution should be large enough to see clearly the effects you achieved.