Computer Graphics

COMP 477/6311

Final Project Information

Prof. Tiberiu Popa

Fall 2017

Purpose

The project will give you the opportunity to create a large scale, fully functional graphics application that integrates all the material covered in this course. It is also the most important component of your evaluation for this course.

Please read this document carefully as it contains very important information about the project. Also note that some information might be subject to change so follow the newsgroup messages for updates.

Process

The project will be done in teams of 3-4 undergraduate students or 1-2 graduate students. Other combinations can be accepted if there is a good reason for it and you should check with the instructor.

Each team has to decide what project they will like to do and a project proposal will be submitted using the electronic system by October 11th, 13:00. The proposals do not account for the final grade, but they are very important as they will define your project and the objectives that you are setting for yourself. I will have them ready after a few days. You should address the comments and submit the final version by October 30th, 17:00.

*D*o not wait until the second proposal submission to start your project. You will likely not have time to finish. Start working on the project as soon as can.

The deadline of the project as well as the demo date will be announced in class and posted to moodle. The project is evaluated during the demos, therefore the demos are mandatory. The teams that will not demo will receive a grade of 0.

Project Content

You are free to chose your project as you like, the only constraint is to use the graphics content that you learned in this class. Most popular examples are probably games, but you are strongly encouraged to do other types of projects

such as ray-tracers, simulators, modelling tools, etc. I will present in class a more exhaustive list of possible projects and I encourage you to see me during office hours for additional ideas.

Project Proposal

The proposal has two main purposes: first is to set the goals of your project against which your work will be evaluated and second to give you some feedback related to the difficulty of your enterprise. If your objectives are too difficult I will ask you to scale it down and vice-versa. I will also ask for clarifications if the objectives are too vague.

The proposal has 3 parts and should be maximum 2 pages:

1. A general description of the project (half a page).
2. A motivation section that describes why you chose this project and why it is challenging. (half a page).
3. A set of 10 objectives that you set for yourself. You will be graded based on the success of achieving these objectives as shown below (one page)
4. A list of references. The references can be paper, tutorials, or simply web-links.

I will elaborate in class more about this part and I will also give you some examples of good and bad proposals.

Project Support and Platform

You are free to use any legally obtained resources for your project: code, assets (i.e. models, images), libraries. You can use your personal computer to develop this project. Your project is not required to run on the machines in the lab, but you need to be able to demo it on a machine that you can bring and set-up at Concordia. This gives you more freedom, but also has certain consequences as shown below in the evaluation section.

New this year, we have a new lab (H862) that can be used for project that involve Kinect devices, Occulur Rift devices or Hololens devices. If you need to use this lab contact the instructor to gain access.

Any external resource that you use you must acknowledge in the README file and in the final report (see the submission section for more details) and you cannot take credit for it as an objective. For example, if you take some code from the internet that does shadow maps, you cannot have shadow maps as an objective. But if you extend the code in some clever way like improving it or making it more efficient then you can have it as an objective, but you need to be able to show and explain in detail what you did and what you took as is.

The course staff will help you during the lab time slots, office hours or by appointment. The support is primarily limited to the graphics content of your project and NOT to provide support for external libraries not used in the course

or to debug your code. It is your sole responsibility to design, write and debug your code and learn on your own any external library you want to use.

Code Management and Submission

Virtually any company that does software development uses some source control software. Therefore, since you are up to four people working together, in this class you are strongly encouraged (but not required) to manage your code using a repository control tool. We recommend a tool called *g*it and a free code repository hosting service called *b*itbucket (google bitbucket). This will make your life much easier in how to manage the code among different programmers as well as backing up your work.

The final project has to be submitted using the online system by the posted deadline. This should be the version that you use to demo your project.

The submission has to be ONE zip file containing all the source tree. The code has to be reasonably documented and has to contain a good README file that, among other, describes what is necessary to compile the project on another machine. In addition to the source code you need the following:

1. A (maximum) 4 pages report that briefly describes the project, the motivation for the project, acknowledges any external resource that you use, describes what is necessary to compile the project on another machine, explains any design and algorithmic decisions done (i.e. what algorithms did you use), a discussion of what worked, what was difficult, and a description of how the load was balanced between members of the team. (Note that in general it is ok that multiple members worked on a given task).

2. A web page encapsulated in a folder called web that showcases your project. The purpose for this web -page is to upload it to the course hall of fame so it should be like a nice advertisement for your project. So it should not be verbose, but it should be flashy and crisp and with lots of pictures. You have a lot of freedom on this. All links used in this web-page should be relative so I can just upload it by copying the folder to the University hosted web-site.

3. Please e-mail the instructor if you would like to give him the permission to use your web page to show in class or publish online on the course web page.

4. This is optional, a short video showcasing your project. For instance, for games you can have some video capture of someone playing the game. If you do this, you can add it as an objective.

Teams

You are responsible to create your own teams based on the rules above. Once the team is formed please send an e-mail to the instructor with the names, IDs

and e-mails of the team members, the designated team-captain as well as a team name. Internal problems within the team should be brought to the attention of the instructor. No abusive behaviour of any kind will be tolerated between team members.

Evaluation

The evaluation of the project is done almost exclusively during the demo so best is to think about the demo as an oral examination. The evaluation follows the following marking scheme (out of 40 marks)

5 marks. The completion of the objectives (0.5 points per objective). You have to showcase them one by one during the demo. All members of the team will get the same marks for this category.

5 marks Report, README and web-page

10 marks. Presentation, demo and Q&A. See the demo section for details. Team members can get different marks for this category.

5 marks. Polish, difficulty, robustness, efficiency. While the completion of the objective list is the qualitative evaluation component of the project, these 5 marks are the qualitative evaluation of your project. For instance, even if you completed your objectives ,if they are not polished polished and well exeuted (i.e. bugs or crashes or too slow) you will lose marks here. Conversely, even if you did not manage to finish all objectives but they were difficult and the prokect is ver well polished you will get a high grade here. Also if you go beyond the boundary of the course with your features and you did research on your own you will score additional points here. Team members will get the same marks for this category.

5 marks. Subjective. These are purely subjective marks that we will award based on the overall feel of the project. Team members can get different marks for this category.

Demo

The deadline for the project and the demo session as well as the location of te demo session will be discussed and decided in class. Refreshments will be provided.

The demo must be given by all members of the team otherwise all members of the team will get a grade of zero for the project. The only exceptions are medical and a dr. note is mandatory. You should try to attend the entire demo session, not only your demo slot, it is nice to support and have the support your fellow students.

The demo can be done either using your own machine or on the machines in H837. If your own machine is NOT a laptop, please come earlier to set it up.

Note that in general we will not be looking at the code, unless we suspect cheating. The evaluation is done almost exclusively during the demo and by looking at your report therefore it is very important to make a good presentation and to be convincing when showcasing your project.

There will be roughly 11-12 teams and the time for each demo is roughly 15 minutes. During this relatively short time you have 8 minutes to showcase your project (presentation): run it, go over the objectives, etc. This part has to be split between the members of the team so it has to be very well coordinated. Rehearse it beforehand.

The rest of the demo I will ask questions regarding the project. The presentation and Q&A part represent a large part of your grade so think careful how you want to present it given the short time.

We look at the clarity and conciseness of the presentation as well as to show the features of your project in the best possible light and highlight the main challenges of each objective,.

IMPORTANT: Please bring a hardcopy version of the final proposal at the demo.

Risk Management

Any large project, especially done in teams, is subject to certain risks that if unchecked can lead to project catastrophies like not having anything really working well during the demo or even worse project not compiling. Despite your very hard work will lead to a grade of zero for the project for the entire team.

Assessing these risks and preparing for contingencies is a very important exercise that you will often have to do in your careers and if done properly it will help you improve your performance.

For you project, you should pay attention to the following factors:

Coding skills. In my experience the programming level of the students in any given class has a huge variance and I cannot stress enough the importance of choosing a project appropriate to your programming level. Select a set of objectives that combine simple problems that you are confident you can finish with more difficult ones that you are not sure exactly how to do.

Organisation and planning. Start early, have milestones and hard deadline that you set internally to your team. Allocate enough time for debugging and assume that some objective will take much longer than expected. You should freeze your code at least 24h before the deadline and only make small changes to the code that you make sure will not introduce unexpected behaviour.

Teamwork. Meet often to discuss and merge the code. Communicate and help each other. Assign roles and authorities in your team. Create a harmonious and productive team environment. Get to know each other's strength and weaknesses.

Code management and backup. Code sharing in team development can be tricky so take precautions. Use a good source control software and do not e-mail back and forth code. This is a recipe for disaster. Test your own code very well before merging with the code from other teammates. Break down the problem into smaller steps and do not move to a new step until the precious step works perfectly. Do not forget to back-up frequently.

Cheating

Cheating is the most serious offence in the academic world. Therefore cheating carries severe penalties ranging from getting a negative grade for a homework or project to failing the course to getting expelled from the University.

To avoid this situation please read carefully the cheating policy of Concordia University, always acknowledge explicitly any help, code or resource you got from a class-mate, friend or the internet and when in doubt, please ask the instructor.