

COMP 477/6311

Animation for Computer Games

Projects

1 Demonstration

- Project demonstrations will be in H-401 on Friday, 19 April 2013, starting at 2 p.m.
- Demonstrations are *not* mandatory. If you don't want others to see your project, or you are too shy to present it, then don't. I will grade your project on the basis of your report, as described below. However, you have much to gain by presenting, and nothing to lose.
- There is no particular order: whichever team is ready can present. If you have particular time requests, and inform me in advance, I will endeavour to respect them.
- There is no particular duration for a presentation, but around 10 minutes is appropriate.

I have 28 project proposals. Assuming that some teams will decide not to demonstrate, we might have about 20 demos. At 10 minutes per demo, we will need 3 hours 20 minutes. So don't go on for too long!
- Your presentation should consist of a brief introduction, highlighting the significant features of your project, followed by a demonstration if possible.

Use slides if you feel you need to, but bear in mind that slides take time to set up and present. Slides might be useful to show technical points (e.g., math or code) or diagrams. If all you have is bullet points with text, slides are probably not worth the effort.

It is not necessary for every member of the team to speak, but the team member chosen to present the project should introduce team members and outline their contributions.

After the presentation, we will have a question-and-answer period if time permits.
- The classroom has the standard AITS setup, with a computer running Windows. There is a USB socket that you can use to download your project. The software environment and graphics card are both fairly limited: .NET programs probably won't run unless you provide the .NET environment, and OpenGL extensions are probably not supported.
- There is also a VGA connection to the projector, enabling you to run your demo from your laptop.

I have ordered an HDMI adapter and it has been promised for Friday: I hope it will be available.

2 Submission

- Reports are due by **midnight on Monday, 22 April 2013.**

- I will grant short extensions supported by a solid reason. However, I do not have a lot of time for grading, especially as many of you are "potential graduates", for whom grades must be submitted in good time. Please try to submit as soon as you can.
- The weekend between demonstrations and submission is intended to give you time to polish your project in minor ways on the basis of feedback you receive or your own feelings about it. And to finish your report, of course.
- Submit *source code*, *executable*, and your *project report*. The source code should be your own, not code for libraries, etc., that you have used in the project.
- Provide me with sufficient resources to compile your project if you can.
If you have used public domain software (e.g., OpenGL/GLUT), it should be feasible for me to compile. If you have used a proprietary system such as Unity, it probably won't be.
- The *preferred form of submission* is via EAS as Project 1. Alternatives:
 - everything on a USB stick or CD;
 - hard-copy of report, software on a USB stick or CD.

Please put physical objects (USB stick, CD, paper, etc.) into my mail box in EV 3.177 (near reception) or my assignment box in EV 3.251.

3 Report

- The goal of your report is to convince me that your project deserves a good grade.
- I do not require formal documentation (requirements, design, use cases, test plans, etc.).
This kind of documentation is appropriate in other contexts (e.g., large teams working in industry) but is not necessary for a class project created by a small team.
- A good report will answer most of the following questions. (Not necessarily all of them, because some questions might not be appropriate for your project.)
 - What makes the project interesting and relevant to COMP 477/6311?
 - What were the key challenges of the project?
 - What were the biggest problems you encountered while working on it?
("Problems" means *technical* problems, not running out of coffee or something.)
 - Did you anticipate these problems, or did they come as a surprise?
 - Did you try something that didn't work, and have to look for an alternative approach?
 - How closely does your final project match your original goal?
 - What did you learn by doing this project?
- If your team has more than one member, the report should include an individual statement by each member, describing the members's individual contribution to the project and collaboration within the team.

- Include a User Manual or design the program to be *completely* self-explanatory.
In previous years, I have seen impressive demonstrations and then been unable to reproduce the effects on my own computer. This is very frustrating!
- Please acknowledge sources of code or documentation that you used. For example:

"We used the NeHe Tutorial on Particle Systems¹ and the FireBlade particle engine²."

4 Grading Criteria

- You will get credit for:
 - Interesting and effective uses of animation.
 - Development and refinement of algorithms.
 - Good visual effects.
 - Novelty.
 - Well-crafted code.
 - Robustness (i.e., I can't crash your program).
 - Documentation (comments in code and Report as described above).
 - Either ease of use or clear instructions (i.e., marks lost if I cannot make it work).
- I do not give a fixed breakdown of marks (10% for good use of libraries, etc.) because each project has its own particular merits. Allocating a fixed amount for library usage would penalize a project that did not use libraries.
- Each member of a team gets the same marks.
I will make an exception to this rule only if I receive a statement, signed by all members of the team, specifying unequal marking. For example:

"We, the members of team *BlowItUp*, unanimously agree that Jane should get 75% and that Tarzan should get 25% of the marks. Signed [Jane] [Tarzan]."
- The marks are not related in a simple way to the superficial appearance of the project. A project that has rather crude graphics but is coded in basic OpenGL might get more marks than a project which looks fancy but is built on a large framework such as Unity.
- The most common reasons for deducting marks are:
 - Using software without acknowledgment;
 - Submitting programs that I cannot figure out how to run.

P.G.
17 April 2013

¹http://nehe.gamedev.net/article/particle_systems/15008/

²<http://sourceforge.net/projects/fireblade/>