**Computer Graphics Final Project Directions**

**Due Dates:**

**Everything is due at 11:59 pm unless otherwise noted.**

**See Blackboard “Syllabus and Schedule” for upcoming due dates. You will only see these in your Blackboard calendar after you have added yourself to a group.**

**Final Project Choice 1 – Group Game Engine and Demo Level**

**Objective:** Create a real-time 3D game engine with OpenGL

Three to four person teams: *for example,*

1 – Graphics programmer – textures, interactivity, game mechanics (upgrades, life etc)

2 – Graphics programmer camera movement and 3D transformations, ray casting for bullets

3 – Graphics programmer - Project Manager, textures, collision detection

4 – Graphics programmer – lights, game AI, level design

There are other jobs but the point is that EVERYONE will be responsible for a significant portion of the graphics programming

Technical requirements for an OpenGL game:

* Complex 3D environment with imported models
* 3D transformations and camera movement
* Multiple Lights
* Textures
* User Interactivity
* A playable game (1 level of a game)
  + This requirement is based on the proposed game
    - Examples of games people have done before:
      * 1st Person Shooter/Action
      * Tower Defense
      * Racing
    - This could include collision detection (feel free to use sphere collisions), physics, artificial intelligence etc

 Extra Credit:

* Effects from the topics section of the Final Project choice 2 options.
* Sound FX/ music
* Character animation
* Particle systems and effects

You can download the following from the web to increase quality (MAKE SURE TO CITE YOUR SOURCES IN THE REPORT):

* 3D Models (e.g. <http://www.turbosquid.com>)
* Software: physics/graphics libraries (*although not recommended – ask me beforehand!!*)
* Textures
* Sounds/Music

Grads: each person must include 1 additional graphics effect from the topic list in ‘Final Project Choice 2’ (or you can propose something to me if there is something else you are particularly interested in)

**Grade Breakdown**: 20% of your final grade.

* 10% final presentation
* 60% Implementation
* 5% webpage and report
* 25% of project grade will be a prototype showing at least 25% progress

**Proposal:** (1 page document):

       Schedule

       Plan broken up into 4 tiers of completeness (which will roughly correspond to grade letters)

* Roles and proposed tasks for each person

       Risky Areas/Backup Plan

       Design document (describe game play, interaction, and graphics)

**Code, deliverables, etc:**

Post link to zip in blackboard.

**Final Presentation Video**

       Presentation Info:

1.     Maximum 10 minutes preferably less (suggest walk up w/ laptop tested and ready to go)

2.     Show web page

3.     Run game for a few minutes

4.     While playing discuss

1.     the game

2.     the technologies

**Group Report (1-2pages. Single spaced, 12 pt Arial font)**

            Describe your game, and include technologies incorporated. Discuss what works and what does not work, based on what you originally proposed and any other features you have implemented. Include screenshots or have them on the webpage. Make sure to include a link to the webpage.

**Webpage**

This is not worth much so don’t spend time making a really fancy webpage. THIS IS NOT A WEB DESIGN CLASS! (Doing a webpage in Word is just fine) I just want you to have something that you can show off to people. This needs to be live on the web. Include a brief description of the game and some screenshots.

**Individual Report**

            Discuss what ***each person in your group*** worked on.

            Assign a grade to each group member (i.e., A,B,C,D,E)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Final Project Choice 2 – Individual Tech Demo**

**Objective:** You must implement and enhance/improve one graphics effect that was not already in a previous project and give a short presentation that teaches everyone how it works and what you did to enhance/improve the algorithm/technique/effect. Some examples of enhancement could be: increasing frames per second, fixing edge cases, combining it with another complementary algorithm for an overall better look, making the final the graphics look qualitatively better while still being interactive.

**Directions: NOTE: ALL CODE MUST BE OPENGL 3.0+ COMPLIANT!** E.g., the things that need render to texture should use a Frame Buffer Object. However, if you anticipate hardware issues with your implementation approach, let me know in advance. Many of the following have example code associated with it, which you are free to use and modify, but keep in mind that some of it may not be up to date or use a different shader language (e.g., Cg). You may need to do additional searching on your own to get things to work.

Then do something to improve or enhance it. All of the algorithms presented below have short comings and tradeoffs. As you implement them, identify those short comings and trade offs. Then do something to fix/improve/enhance/make-look-better one of those. If you come up with an enhancement idea and aren’t sure about whether it is sufficient to meet this requirement, feel free to ask me.

**Topics:**

Choose from the following list (or you can propose something to me if there is something else you are particularly interested in):

* Cube Maps (<https://learnopengl.com/Advanced-OpenGL/Cubemaps> )
  + Environment Reflection, see also (<http://http.developer.nvidia.com/CgTutorial/cg_tutorial_chapter07.html>)
  + Environment Refraction
* Shadows
  + Shadow Maps <http://fabiensanglard.net/shadowmapping/index.php>), (<https://learnopengl.com/Advanced-Lighting/Shadows/Shadow-Mapping> )
* Stencil Shadow Volumes (<http://www.swiftless.com/tutorials/opengl/basic_shadows.html>)
* Advanced Collision Detection and Response
  + AABB (<http://www.gamasutra.com/view/feature/3426/when_two_hearts_collide_.php>)
  + OBB (<http://www.gamasutra.com/view/feature/3383/simple_intersection_tests_for_games.php>)
  + Physics Simulation (<http://gafferongames.com/game-physics/>) , (<http://www.gamasutra.com/view/feature/3032/exploring_spring_model.php>)
* Space Partitioning
  + Binary Space Partitioning Trees (<https://web.cs.wpi.edu/~matt/courses/cs563/talks/bsp/document.html> )
    - Octtrees (<http://www.cs.berkeley.edu/~demmel/cs267/lecture26/lecture26.html#link_3>)
* Particle Systems (<http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/> )
* Bloom (<https://learnopengl.com/Advanced-Lighting/Bloom> )
* Blur
  + Motion Blur (<http://ogldev.atspace.co.uk/www/tutorial41/tutorial41.html> )
  + Depth of Field (<http://developer.download.nvidia.com/SDK/9.5/Samples/DEMOS/Direct3D9/src/HLSL_DepthOfField/docs/HLSL_DepthOfField.pdf>)
* Procedural Generation
  + Terrain (<http://vterrain.org/>)
  + Geometry (e.g., Trees, plants, water, etc) (<http://www.cs.washington.edu/education/courses/cse557/07wi/projects/final/artifacts/kehrt-lerner/#note1>)
* Find a paper from ACM SIGGRAPH in the last 5 years and implement it.
  + <https://dl.acm.org/action/doSearch?AllField=graphics&SeriesKey=tog&startPage=&EpubDate=%5B20180105%20TO%2020230105%5D&queryID=19/5134110244>
  + Search for the paper through lib.utsa.edu
* Physically based rendering (<https://learnopengl.com/PBR/Theory> )
* Some cool shader effect that isn’t listed here
* Integrate a 3D User Interface
* Virtual Reality
* Augmented Reality
* *or propose something else that is relevant to computer graphics…*

Grads: Each person should incorporate two items from the above list (or you can propose something to me if there is something else you are particularly interested in – e.g., deferred rendering).

**Proposal:**(1 page document):

       Schedule

       Plan broken up into 4 tiers of completeness (which will roughly correspond to grade letters)

       Risky Areas/Backup Plan

**Code, deliverables, etc**

Post link on Blackboard

**Final Presentation Video.**

       Presentation Info:

1. Maximum 10 minutes preferably less
2. Run demo
3. Have a powerpoint presentation that explains the details of what you implemented so that other people in the class can learn how to do it – you will be graded on the clarity of your presentation and how well you demonstrate your knowledge of the topic

**Final Report**

            Describe your demo, focus on how the included technology works and how you implemented it. Discuss what works and what does not work, any other features you have implemented, and describe ways to improve it or add to it in the future. Include screenshots on the webpage only.

**Grade Breakdown**: 20% of your final grade.

10% project presentation

5% final report

40% Implementation

20% Enhancements to Implementation

25% of project grade will be a prototype showing at least 25% progress