

INFO3111 “C++ Graphics”

Midterm Exam, Friday, May 26th, 2023

Instructor: Michael Feeney

The exam format:

- You may use any resources you feel are necessary to complete the exam, but you are to answer the questions **on your own**. I will be looking for plagiarism (i.e. copying) very carefully. There is no possible way that the specific code to answer these questions, or the output to the screen, would be very similar to the look of another student's code. Remember, this is a test and there are very clear policies about cheating on tests.
 - <http://www.fanshawec.ca/admissions/registrars-office/policies/cheating-policy>
 - http://www.fanshawec.ca/sites/default/files/assets/Ombuds/cheating_flowchart.pdf
- It is an “open computer” exam. You have access to any written materials and whatever is on your computer (including code from class or that you've written/modified), but you do **NOT** have access to the internet (with the exception of uploading your solutions to FOL at the end) – to be clear:

If I see anything remotely “internet” or “network-y”, even if this is an accident on your part, then I will assume you are cheating and will:

- 1) *Ask you to pack up and leave the classroom.*
- 2) *File an academic offence.*
- 3) *Give you a grade of zero on the exam (and possibly more if this is not your 1st offence)*

This includes using, or even looking at, your phone.

- The questions are **NOT** of equal weight, nor does the exam, necessarily add up to 100%. Bonus questions are simply added to the total amount - as if they are "regular" questions - but bear in mind that as they are "bonus", they will be marked to a higher standard.
- The exam has **five (5)** questions and **ten (10)** pages.
- The answers may be one, or a combination of, the following:
 - Short answer (in your own words)
 - Snippets of code
 - Complete running solutions
- **CLEARLY** indicate which answer goes to which question. My *suggestion* is that you place each answer (entire solution) in its own folder, named “Question_01”, “Question_02” and so on (or something equally clear) - specifically, copy *the entire solution folder* (with *everything* in it), to another folder, then rename that new folder. Another option is to create a Visual Studio solution and add a number of projects – one per question – to it. If I can't make heads or tails of what question is what, I probably won't even mark it. The exception to this is if the questions clearly “build” upon each other. i.e. there doesn't have to be *any* change of the code/project to show the results of the question.
- Do **NOT** do some clever “*oh, you just have to comment/uncomment this block of code*” nonsense - if you are expecting me to *edit* your source code: I'm not going to do that, and will simply run the code as submitted.
- If the questions clearly and/or obviously build on each other (which many do), you may combine them, but **MAKE IT 100% CLEAR** to me what questions the solution is attempting to answer. If you feel I need to know something (key mappings, that you edited the 3D models, etc.) please indicate this with a “**readme**” file or some note **in the root folder** (*not* buried in the source code/project folders somewhere).

- **If the solution does not build (and run), I will not mark it** (so you will receive zero on questions that can't be built and/or won't run). When I say "run", I'm not speaking about some, random, unforeseen bug, but rather something that you should have obviously dealt with, like memory exceptions, etc.
- While I *might* use the Debug build while marking, your submissions **must build and run in Release build**.
- Unless otherwise indicated, all these solutions assume that you are creating/using a C++ project using Visual Studio 2022 using the OpenGL 4.5 API (with GLFW, glad, and GLM), and build in **64-bit Release**.
- Your solution may **not** contain any third-party libraries (like boost), **smart pointers**, or the **"auto"** keyword.
To be clear: if you have **any** of these elements, you will receive a mark of zero (0) for that question. NO exceptions: you should be aware of what's currently used in industry (99% C++98/2003), and what just happens to be in the newer standards.
- When ready to submit, **please** delete all the "extra" Visual Studio files before zipping it up (remember this is C++, so all I really need is the .h and .cpp files, right?), like the "Debug" and "Release" folders with the "obj" files, as well as the intellisense file (the ".vs" file in the root).

NOTE: Unless otherwise indicated, you should be displaying the ply files that were included with the exam (in the PLYFiles.7z and PLYFiles.zip folder – they are the same files, just different archives).

Some notes about the models:

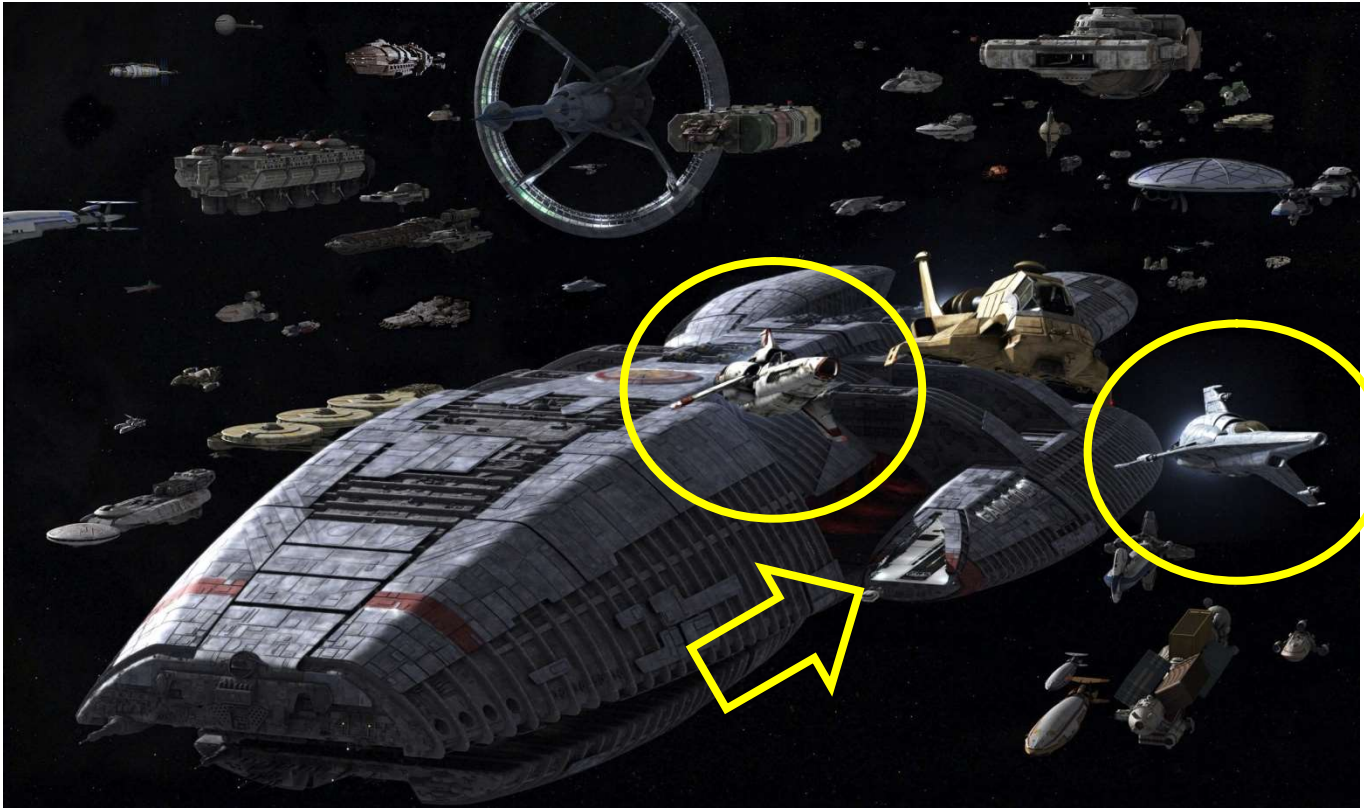
- The models are of **VERY** different resolutions. Depending on your system, the larger ones will take **MUCH** longer to load, particularly with the Debug build.
- However, the "matching" models (example: all the "Galactica...") are the same overall size (i.e. the "bounding box" or "extents" of the various models are the same). They are also (sort of) the same shape, though the size reduction mangles the models somewhat.

My suggestion is the USE THE LOWEST RES MODELS you can handle loading (time wise) and when everything is working like you want it, replace with the highest resolution and run the solution with the Release build.

I will be marking with the high-resolution models. If your submission shows the lower resolution models, you will lose marks. Same with

The Questions:

“Fleeing from the Cylon tyranny, the last battlestar, Galactica, leads a rag-tag fugitive fleet on a lonely quest...a shining planet known as Earth.”



Just in case you have no idea what “Battlestar Galactica” is, here’s a brief summary:

Battlestar Galactica was originally a 1978 TV show, attempting to duplicate the "buzz" of Star Wars (which released in 1977). In 2004 a "re-imagined" version hit the Syfy network (trivia note: apparently "re-imagined" was 1st used for Battlestar Galactica).

You should watch it. It's awesome. Lots of ships, lots of interpersonal drama. Edward James Olmos, Mary McDonnell, James Callis, and the until then, virtually unknown Katee Sackhoff and Tricia Helfer - what's there not to love?

In the future (or is it the distant past??), humans are all over the galaxy. They make robot servants - called "Cylons" - who revolt and there's war, then a tenuous peace.

(Fun fact: in the original 1978 version, an alien race made the Cylons, not the humans.)

During the war, enormous "space aircraft carriers" (basically) called "Battlestars" are made. They are several kilometers long (really), with thousands of humans crew members living on board. They also have hundreds of single person space fighters called "Vipers".

In the picture (above), the Battlestar Galactica is the giant ship (if you are unclear... seriously?). The picture shows two version of the Viper, an older one (in left yellow circle) and a newer one (in the right yellow circle).

You can see one of the huge landing bays Galactica (yellow arrow). The Vipers (and other ships) land on these runways sort of things inside the bays. Note that these landing bays bays are HUGE relative to the Vipers (the right picture):



The following models are in the "PLY Files" 7z/zip file :

- Battlestar Galactica (several resolutions)
- "new" version of the Viper
- Cylon Raider
- "old" version of the Viper (several resolutions)

They are in various formats (xyz only, xyz & normal, and xyz & normal & uvs)

If you end up using larger models – and have a larger “scene” – you will have to change the “near” and “far” clipping plane on the projection matrix.

It's this line (line 428 of theMain.cpp with the latest git commit):

```
mProjection= glm::perspective( 0.6f, ratio, 0.1f, 1000.0f );
```

- The 3rd number (0.1f) is the "near" plane. Any vertices *closer* to the camera than this (in world space) is “clipped” (not drawn).
- The 4th number (1000.0f) is the “far” plane. Any vertices *farther* from the camera than this (in world space) is “clipped” (not drawn)
- Our original “triangle or death” was really tiny (fit into a 1x1x1 cube). The “apartment building” model in the current scene fits is 48x85x68 in size, so everything in the scene is much closer than the 1,000 unit “far” plane.
- However, the “old viper” model is 100x100x200 (approximately). If you keep the model at this scale and move away, it'll almost certainly be more than 1,000 units – and the viper will disappear in the distance.

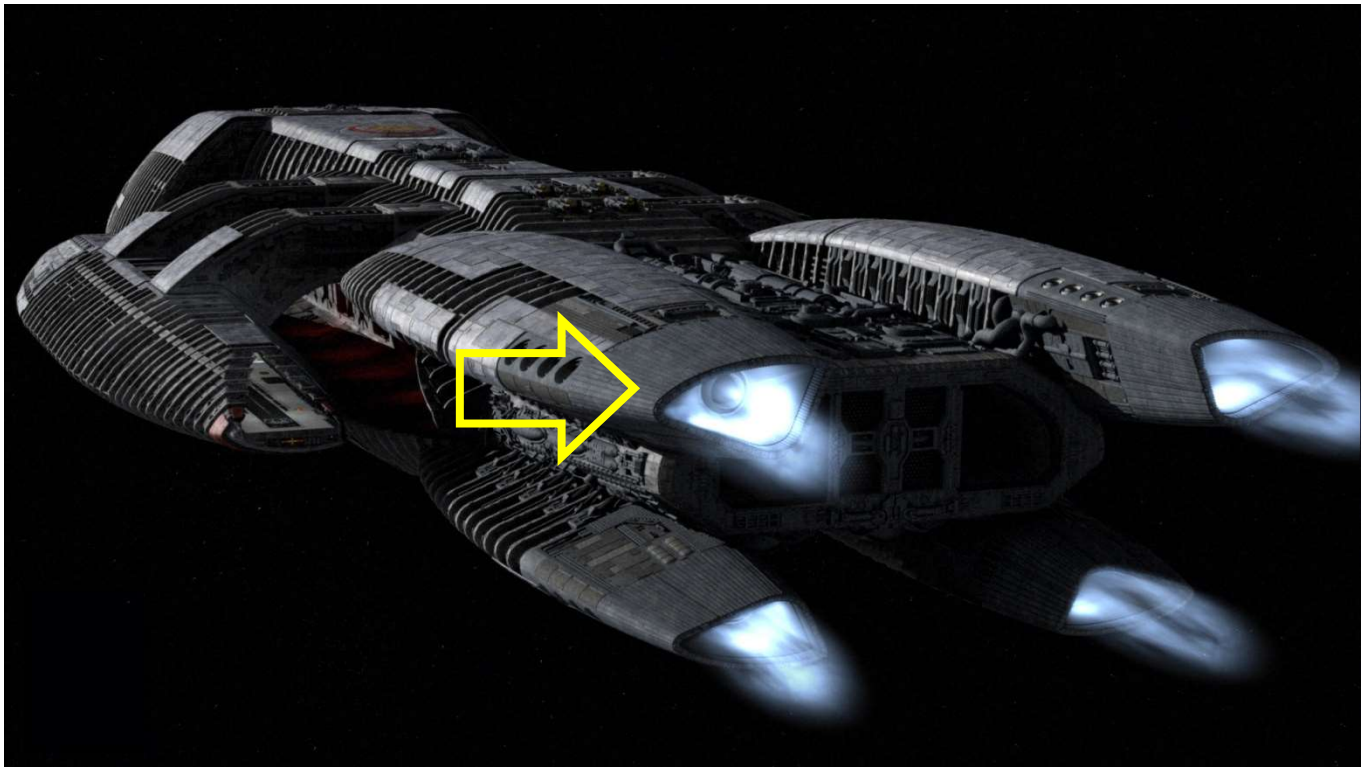
General rule: make the “near” plane as LARGE as possible (this is the *closest* thing to the camera), and the “far” plane as SMALL as possible (this is the farthest thing from the camera).

If you don't change the scale of the models, I'd suggest:

- A near plane of 1.0 or even 10.0f.
- A far plane of 10,000 or even 100,000.0f

```
mProjection= glm::perspective( 0.6f, ratio, 1.0f, 100000.0f );
```


1. (20 marks) Place the Galactica in the scene, along with a couple Vipers, in a "dramatic" shot like on page 3.
 - Place the camera in a dramatic location, far away and above, looking "down" on the Galactica (like in the picture).
 - Place two (2) Vipers (one "new" and one "old") like the picture, nearer to the camera and "flying up and off" at an angle (like the picture)
 - Newer Vipers (the curvier ones) are more dark grey - colour them as such
 - Older Vipers (the less curvy ones) are much lighter *almost* white coloured - colour them like that.
 - The Galactica is dark grey.
 - Note that the scene is HUGE, so you'll have to place at least a few (likely three) lights in the scene, and manipulate the distance and attenuation so everything is more-or-less evenly lit. i.e. I *don't* want some giant glaring "bright" spot on one (or more) of the models.
2. (40 marks) Add four light-blue/white lights to show the "plasma" (or whatever) engine exhaust (or whatever) that's coming out of the Galactica:



- Place the camera near the rear of the Galactica (like the picture above)
- Add four lights, inside the "engine" area (like indented parts). Note you'll need to control the attenuation and/or the "cut off distance" to make the lights both bright AND limited to insides of the engines. I don't care about a little light "spill", but I want the lights on the *inside* of the engines and not washing all over the ship
- You still need to see the Galactica, so you'll need to keep the same "general" lighting that you had in Q1 (it doesn't have to be *exactly* the same lights, just that it's a general area light, like in the picture)
- If you use the same solution as Q1, use the "9" key to place the camera at this location and angle, and the "0" key to put it back where it was in Q1.

3. (40 marks) If you watch the "Battlestar Galactica - Big Battle Compilation (720p)", you'll see Vipers launching from, and landing on, the Galactica, as well as the "Cylon Raiders" (the evil Frisbee/robot looking ships with the scary back-and-forth red "eye" things) leaving their "BaseStar" and attacking. There's also a lot of things blowing up. And yelling... So.Much.Yelling.

An "offline" copy of this video is on FOL, called: "**Battlestar Galactica - Big Battle Compilation (720p).mp4**".

Do **NOT** go to YouTube to watch this as you aren't supposed to be on the internet, right?

Also **MUTE YOUR LAPTOP** (unless you have headphones, I guess)

You are to recreate the scene from the pilot episode (the clip, which *isn't* from YouTube) where the Cylons take on an entire wing of "newer" Vipers and "hack" them or something, making them useless. Then they destroy them all.

This is the "**Cylons take out the newer Vipers by hacking.mp4**" clip, around 3:17 in. (**DON'T** watch this on YouTube).

The "older" Vipers are immune to this hacking, so some of them get launched to "kick Cylon butt". Unfortunately the people in the "new" Vipers don't have a good day.

- Place five (5) "new" Vipers flying towards the Cylons. They should be "pretty far" away from the Galactica (you can see in the YouTube clip they are quite far away, like 50 km maybe?). They should be in some kind of sensible "formation" (i.e. like the video - facing the Cylons, flying away from the Galactica, near each other, etc.). These need to be the "newer" Raiders (so make them the right colour - dark grey)
- Place two (2) Cylon "Raiders", in formation, flying towards the Vipers. They should be pretty far away from the Vipers.

NOTE: I realize that you only have a limited amount of "space" in your "scene", since you have to have *both* the Vipers and Cylons in the same screen ("shot"), so when I say "pretty far", I mean they shouldn't be "right beside" each other, but I also don't want them to be micro-tiny, either.

- Using a really tiny sphere, and a really "tiny" light (i.e. mess with the attenuation and cut-off), mimic the "eye" thing on one of the Cylons. I'm pretty flexible about this, but it should be clear that there's a little, bright "eye" light, but I don't want the entire Cylon Raider awash with red light, either.

4. (30 marks): The new Vipers are having a *really* bad day... mimic the slaughter of the poor Viper pilots, *after* the Cylon hacking thing has happened (around 3:59 in the YouTube clip), but just before they all blow up (around 4:20 and on).
- Change the orientation of the five (5) “new” vipers *across all axes* (x, y, and z) so they look sort of “random”, like they have “drifted”, like they are in the clip.
 - Add a bunch (at least five (5)) of “missiles” or “bullets” from the Cylons. You can use the “Tear Drop” shaped model for this to mimic this effect:
 - One of these models has the models facing “out”, like every one of the other models. The other one has the normal facing “in” – or have their normal “inverted”.
 - These “inverted” models are “inside out” – i.e. their normal are facing inwards. Remember that by default, only the “front” facing triangles are rendered. Also, the lighting only “lights” objects that face the camera. When these are drawn, you are really seeing the “back” of the model.

This mean if you place a small light at the same location as these “inverted” models will sort of look like they “glow”.
 - The regular ones (the ones that don’t have “inverted” in the title) can still be lit up, but if you place a light at the same location, the light will be “inside” the model. The problem is that since the normal face *away* from the light, the object isn’t lit. To illuminate these sorts of models, you’d have to place a number of lights around the model.
 - You can use whichever model you’d like, but I’m looking for this “glowing tracer bullet” sort of effect like you see in the movie clips.

5. **BONUS** (20 marks): Implement some rudimentary animation for this "hacking" scene:

- By pressing a button on the keyboard, you start this, and it runs continuously (i.e. I do *not* want to *hold* the button down for this to take place). You can do this by adding a boolean variable to your solution, then check for the keypress, changing its state. You then read this boolean variable in the main loop (the while), updating the state of the appropriate game objects.
- Make the eye move back and forth on the front of the Cylon Raider. I'm **not** looking for this to *actually* move in a the slightly circular fashion behind the "eye slits" like in the video (though that would be neat);

I'm looking for a small back and forth movement limited the width of the "face" of the Cylon Raider. This can be done with a really small, bright "point" light or a really narrow, bright spot light (shining on the "face" of the Cylon). Your choice.

- Make the Vipers gently rotate ("tumble") in space. I'm looking for the following:
 - They should move a random speeds across *all* axes, so x, y, and z
 - The "rotational speeds" should be *different* for each axes *and* with each Viper. In other words, any individual viper should have different rotational speed on its x, y, and z axis AND each Viper should have slightly different speeds.
 - They should be gentle, not like some crazy carnival ride speed (like in the video).
 - You do **not** need to accelerate to that speed (like in the video) - simply pick a speed and start the "animation"
- Change the location of the camera (or have multiple camera angles, chosen by various keyboard keys) to clearly show this happening.

From Stack Overflow, here's how you create a random float number:

(note that you *cannot* use the C++11 `<random>` header, since you aren't allowed to use C++11. Also, the C++11 `<random>` while "all that", still isn't actually bloody random... gheesh)



This will generate a number from 0.0 to 1.0, inclusive.

```
float r = static_cast <float> (rand()) / static_cast <float> (RAND_MAX);
```

This will generate a number from 0.0 to some arbitrary `float`, `x`:

```
float r2 = static_cast <float> (rand()) / (static_cast <float> (RAND_MAX/X));
```

This will generate a number from some arbitrary `LO` to some arbitrary `HI`:

```
float r3 = LO + static_cast <float> (rand()) / ( static_cast <float> (RAND_MAX/(HI-LO)));
```

Note that the `rand()` function will often not be sufficient if you need truly random numbers.

Before calling `rand()`, you must first "seed" the random number generator by calling `srand()`. This should be done once during your program's run -- not once every time you call `rand()`. This is often done like this:

```
srand (static_cast <unsigned> (time(0)));
```

In order to call `rand` or `srand` you must `#include <cstdlib>`.

In order to call `time`, you must `#include <ctime>`.

share improve this answer

edited Mar 5 at 9:28



Florin Mircea
476 ● 7 ● 12

answered Mar 26 '09 at 15:54



John Dibling
73.2k ● 17 ● 121 ● 246

(<https://stackoverflow.com/questions/686353/c-random-float-number-generation>)

That it. So say we all*.

(*Battlestar Galactica reference. Really, you should watch it.)