# **INFO6020 – Graphics 2**

**Midterm Exam – March 1st, 2017**

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 book” exam. You have access to anything you book or internet resource you’d like
* The questions are ***NOT*** of equal weight. The exam has **five** **(5)** questions and **six (6)** pages (there’s lots of “parts” to each question, though, but I tried to clarify what I’m looking for in each)

* **CLEARLY** indicate which answer goes to which question. My suggestion is that you place each answer in its own folder, named “Question\_01”, “Question\_02” and so on (or something equally clear). 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.   
    
  (However, if the questions naturally “build” on each other, and there’s no “uncomment this and that” nonsense between questions, then I’m OK with them being combined)
* Do ***NOT*** do some clever “*oh, you just have to comment/uncomment this block of code*” nonsense. However, if the questions ***CLEARLY AND OBVIOUSLY*** build on each other, you may combine them (like if one question places objects, then the next one moves objects around with the keys) – even so, **MAKE IT 100% CLEAR** to me what questions the solution is attempting to answer.
* Place any written (“essay” or short answer) answers into a Word, RTF, or text file. Again, *clearly* indicate which question you are answering.
* If you are combining answers (which is likely), please indicate this with a “readme” file or some note (*not* buried in the source code somewhere).
* For applications: if it doesn’t build and run, *it’s like you didn’t answer it*. I’ll correct trivial, obvious problems (like you clearly missed a semicolon, etc.), but you need to be sure that it compiles and/or runs.
* You have until **11:59 PM** on **Wednesday, March 1st** to submit all your files to the appropriate drop box on Fanshawe Online.   
    
  **NOTE:** Although this may “look and feel” like a project, it isn’t, it’s an **exam**, so there is **no concept of “late marks**”; if you don’t submit your files by 11:59 PM, you don’t get any marks at all. *Don’t Be Late submitting.*

(Also be **SURE** that you are actually submitting the correct files)

* Your solution may **not** contain any third party libraries (like boost) or any "C++11" features (auto, ranged for, initializer lists, etc.). If it has either of these things, the question(s) will not be marked (because it won't build). There has been some stress about this, so here’s a “meeting you ½ way”: if it builds in Visual Studio 2012, then you’re OK.
* When ready to submit, PLEASE for the love of all that is good on this Earth, 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 “ncb” (intellisense) file – this will save a tremendous amount of space and shorten your upload time.
* **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.
* Unless otherwise indicated, all these solutions assume that you are creating/using a C++ project using Visual Studio 2008, 2010, 2012, 2013, or 2015 using the OpenGL 4.x API (with freeGLUT, GLEW, and GLM). I’ll be compiling opening them in VS2015.
* I’m not at school on Wednesdays (no scheduled classes, anyway), but you can reach me through e-mail ([mfeeney@fanshawec.ca](mailto:mfeeney@fanshawec.ca)) or by calling the school.

## 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:

* Curiosity Landing Site (the terrain, from the NASA 3D model site)
* The Curiosity Rover (again, from the NASA 3D model site. This was 3D printer model that I – tried do – clean up)
* A “flame” model (which is pretty bad, but it’s something that sort of looks like a flame)
* A Sky Crane Model (which isn’t the actual one from the mission, but it looks pretty cool, and has “nozzles” for the jets)

## The Questions:

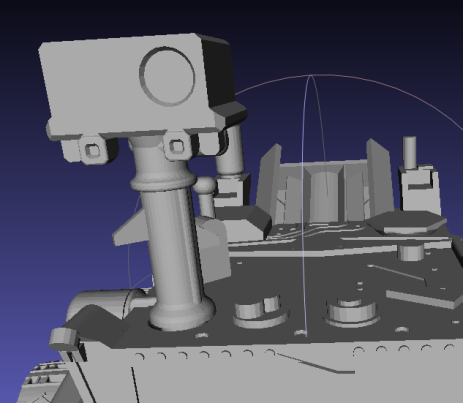
**“Seven Minutes of Terror.”**

In 2012, NASA JPL successfully landed the Curiosity rover on Mars. Because the lander was so large, they had to attempt a new, and fairly risky landing technique called a “sky crane”. To explain the process, they released a video called “Seven Minutes of Terror”: <https://www.youtube.com/watch?v=Ki_Af_o9Q9s>

It’s exciting stuff.   
  
You are to recreate the specific portion of this video where the “sky crane” starts does the actual lowering of the rover, then flies away, then what happens after the rover has landed (which isn’t in the video... but is HAPPENING RIGHT NOW ON MARS!! Think about that!!).



Note: you don’t have to “draw” the cables (though you can, if you like), but you will have to animate the “lowering” of the rover.

1. (10 marks) Place the sky crane and rover in a scene above the landing site. We are talking about the “touchdown” portion of the image above. Place the camera somewhere where you can see the lander, the crane, and (some of the) ground.
2. (30 marks) Complete the “Sky Crane-Rover Separation”. You need to show:
   * The “thrusters” being fired and running. Use the “flame” model to show the thrusters “on” (no “flame” model means they are off). Add “flickering” lights near/inside the “flames” to show thrust (i.e. quickly and randomly change the attenuation of the lights to give a “thrusting” effect”. They should be somewhat transparent.
   * You can assume that the cables are already “cut” (or whatever happens to them?), and the crane is flying away (it crashed a few miles away, apparently).
   * When the “F” key (or another key – just let me know) is pressed, the crane turns slightly and flies away (I’m not really concerned about this – I’m only interested in how the flames of the rockets look)
   * There should be a skybox with stars (or something – clouds are fine, I guess).
3. (30 marks): Duplicate the first image (or one of them, anyway) from the rover. You can find the one I mean here: <https://www.nasa.gov/images/content/674898main_pia16013-full_full.jpg>. Note:
   * It’s black and white
   * The camera is somewhere on the rover (from the “stalk” thing on the top of the rover, I’m assuming)
   * There’s stuff at the edge – that black stuff with the numbers on it.
4. (30 marks): Make a “selfie” just like the rover did!
   * Here is an article about it: <http://www.huffingtonpost.com/2014/06/30/how-curiosity-rover-took-selfie_n_5543343.html>
   * This is from another camera on the “arm” portion, but you don’t have to include that – you can just assume it’s there – but the point is that you are getting a “close up” image of the rover, from quite near the rover.
   * The colours are a little “off” (a little more “brown-orange” than “normal” – note that this is, apparently, what it really looks like, but I want you to *add* a “brown-orange” tint to the image)
   * The image is somewhat distorted, which you can emulate by using a very wide field of view combined with having the camera very close.
   * On the model, there’s a place for a single camera lens:   
     Place a sphere (or ½ sphere) in that location (for the actual lens). What it should look a “fish eye” sort of lens. Make this lens semi-transparent as well as also partially reflecting the skybox (so you can see some stars reflecting in it).



🡨 a “fish eye” lens. Notice that the lens sort of “bulges” out – this is what I’d like you to mimic when you place the sphere (or ½ sphere) in the “camera” location on the rover.

1. (20 marks): See what the rover sees
   * Place the “camera” (i.e. what you are rendering on the screen) where the camera on the rover is
   * Use keys to pan and tilt the view, from the rover’s camera
   * Add some basic reticle effect, something like this: <http://www.ultimatehuntingandfishing.com/onlinestore/wp-content/uploads/2014/10/LEICA-ER-2.5-10x42-Rifle-Scope-Plex-Reticle-Elevation-and-Windage-Target-Turrets-50030-4.jpg>
   * Change the location of this reticle based on the “tilt” of the camera (so if you tilt the camera up, the reticle would go “down”) – i.e. the centre of the reticle should be pointing at the horizon.   
     (You only have to tilt the camera up and down a little, like 15 degrees max)

**That’s it**