# **INFO6028 – Graphics 1 - Mid-term Exam – Fall 2014**

December 9th, 2014

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>
* The questions are ***NOT*** of equal weight.
* There are four (4) pages with ten (10) questions (*INCLUDING* bonuses)
* 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 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.
* Place any written 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 **Tuesday, December 9th** 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)

* You can reach me through e-mail ([mfeeney@fanshawec.ca](mailto:mfeeney@fanshawec.ca)) or by calling the school.

## Questions:

Unless otherwise stated, assume *you have to create an application that demonstrates the following.* In other words, assume each question starts with “Create an application that...”

1. (20 marks) Replace the “space” skybox texture(s) with another “daytime” skybox. You can \*not\* use another “space” skybox, nor one that is generated from the “Spacescape” program.   
     
   What I suggest:
   * Type “skybox” into Google and use one of the ones it returns, or...
   * Go outside with your phone and take six pictures (up, down, left, right, front, back) and use those.

Some notes:

* + It will likely look “ugly”, expecially if you had to “chop it up” from a skybox you got from Google. Don’t worry about that – your artist will have to stress about it looking good, you will only have to worry about it getting into your game.
  + Remember: this /isn’t/ a project, so if two (or more) of you have the same skybox, it better be clear that it’s not exactly the same.
  + **Give me the link to the skybox you used.**

1. (10 mark **bonus**) Incorporate both your new skybox and the “old” space skybox (or get a different “night time” skybox). Demonstrate a smooth transition between the two using user controls (keys, mouse, etc.), or with a timer.
2. (2.5 marks each, “essay/short answer” style) Questions:   
   * Why did we move the skybox with the camera? What happens if you don’t (i.e. do you “always” have to do this? Why or why not?)?
   * In class we used a sphere to represent the skybox. But it’s a “box” (“cube”)!! What’s the “deal” with that? In other words, how can we even get that to work at all – why don’t we /have/ to use a cube when using a skybox?
   * If you just use the “regular” sphere (or cube) as a skybox, you can see the texture when the object is small, but not when the object is large (i.e. when we’re using it as an actual skybox, far larger than the camera). What’s going on with that and how can you solve the problem (and why – don’t just say “oh, we did this and that”, I want to know /why/ we did “this and that”)?
   * In the way we used the skybox, there was a potential issue with the perspective transformation (the one in the “reshape” callback). What was this potential problem and how do we deal with it?

Engine

Landing bay:

1. (20 marks) Demonstrate “discard” transparency. You \*can’t\* use the screen door example we used in class.
2. (5 marks, “essay/short answer” style) The GLSL “discard” command can be used as an optimization. Explain why.
3. (40 marks) Using a single, “flat” model (like the seafloor or something like that), ***clearly*** show the use of 15 lights in a single shader (we used 10 in class).

1. (5 marks, “essay/short answer” style) Why can performance suffer as we have more and more lights? And how much does it suffer, a little bit (like 2x the lights == ½ the performance), or a lot (as we add more lights, the performance degrades more “quickly”)? Why is this?
2. (20 marks) Draw 10 objects of varying (say “random”) levels of transparency, but in a way that you can “see through” them all. Make sure that you have some user controls (or some timed animation) that alters the scene so we can also see them “separately” (i.e. “side by side”/whatever as opposed to “in front” of each other).  
     
   For example, consider this image:   
   (from: http://www1.pictures.zimbio.com/gi/International+Art+Fair+Contemporary+Objects+qtnXQLmCMygl.jpg)



This image shows them “beside” each other (they aren’t blocking each other), but it I looked /from/ the left or right, I would be looking through them all.

1. (5 marks, “essay/short answer” style) What did you do to make the transparency in Question 8 “look good”?
2. (20 marks **bonus**) We’ve been using an “ubershader” in class. That is, a single shader that has a switch/if statement and/or a bunch of Boolean/integer variables to allow us to select or turn off/on various aspects of the shader.   
     
   The shader is selected by the call glUseProgram(), passing the “name” (integer value) of the shader. If you’re using my (very scary, “good enough for Rock-n-Roll”) CGLShaderManger, this is encapsulated in the “UseShaderProgram” method (it’s not magic, it’s calling glUseProgram() inside – shhhh, don’t tell anyone!).  
     
   Spit out the (some) of the functionality of the “ubershader” into another shader.  
     
   In other words, you should have two (2) shaders, instead of one.   
     
   Some suggestions:
   * I would suggest keeping all the uniforms, vertex layout, etc. exactly the same.
   * In fact, I would further suggest that you keep the same vertex shader, but create two fragment shaders – i.e. your “two” shader programs are using the same vertex shader code.
   * Pick something simple, like making the 2nd shader draw the just debug colour or something

Create a scene where you clearly demonstrate the use of these two shaders. In other words, you should be drawing some objects with one shader, and some (at least one) object(s) with the other shader.

(That’s it for the exam).