# **INFO6020 – Graphics 2**

**Midterm Exam (Make up version) – April 18th, 2017**

Instructor: Michael Feeney

## The exam format:

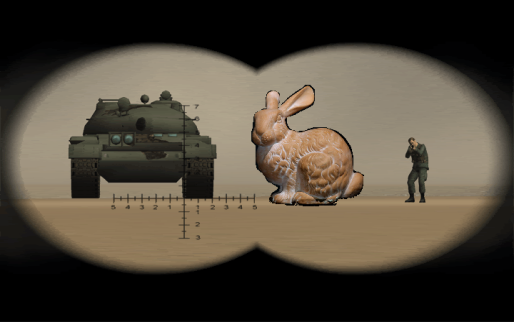
* You may use any resources you feel are necessary to complete the exam, but you are to answer the questions **on your own**. I will be looking for plagiarism (i.e. copying) very carefully. There is *no possible way* that the specific code to answer these questions, or the output to the screen, would 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 **four** **(4)** questions and **five (5)** 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 any 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.

1. (25 marks) You are on the hunt for bunnies, specifically Stanford bunnies.   
     
   Create a scene that has:  
   * a number (at least 20) of Stanford bunnies, of various sizes. These would be lit by at least two lights, and no textures.
   * “ground” (for the bunnies to sit on), that can be anything you’d like: a single ply file or multiple ply files, arranged together. The ground is lit by two lights, but also has a single texture (something grass and/or dirt like).
   * Some fences (to contain the bunnies). These can be lit if you’d like, but they have to use some sort of “discard transparency.” Note: You can **NOT** use the textures that we used in class for this; you must use another set of textures.
2. (50 marks) Like real bunnies, Stanford bunnies are easily frightened, so you can’t get too close. Alter your scene in question 1 so that you can “zoom in” on the bunnies.   
     
   Specifically:  
   * You need to turn the camera up, down, left, and right. You can use the mouse of the keyboard. The camera does not have to *actually move* in the scene.
   * Press the “z” key to “zoom in”.
   * Press the “x” key to “zoom out” (i.e. go back to the regular view)
   * When zoomed in (*most of the marks are for this part*):
     + You need to move the camera “closer” to the bunnies. Note that the camera has to be looking in the same direction. You can do this in a number of ways (including actually repositioning the camera), but the effect has to be that the objects you are seeing are “closer” than they are when “zoomed out”. Ideally, you change the field of view (in the projection transform) which causes an effect like this (note foreground and background of scene with respect to the actor):
       - <https://www.youtube.com/watch?v=svEPWBxpYjo>
       - <https://en.wikipedia.org/wiki/Dolly_zoom>   
           
         (Note that in the actual, physical movie effect, the camera is *moving* towards the object. I *don’t* want to see this *motion*, just the end results of the motion – if you look at something through a binocular or telescope, you have a “flattened” depth of field – *that’s* what I want)
     + There has to be a full-screen “reticule” effect that look like a typical TV/movie “binocular” or “telescope” sort of effect. Something like these:

1. (30 marks) Add one of the following particle effects to the bunny scene:

* A cosy fire that the bunnies can sit around, ***or***
* Rocket exhaust, which makes the plain bunnies Super Rocket Robot Bunnies!

Regarding marking here: Although it is up to you to decide how you want to implement the effect (point particles, different colours, imposters, 3D particles, etc.) the effect has to look “convincing,” in the sense that a “typical” person would look at it and immediately determine what you are showing.

So, if you do something like a red coloured point fountain in the centre of the screen, it won’t look nearly as good (aka “convincing”) as a system based on flame texture based 3D/imposters.   
  
I’m specifically wanting:

* “Flickering” of some kind, which includes changes in colour of the fire ***and*** changes in intensity of the light being shone on nearby objects, over time: <https://www.youtube.com/watch?v=9lh_becOt4Y> (note the light on the ground is slightly getting brighter and darker, randomly, over time)
* Some kind of transparency in the “fire particles”. You should be able to ‘see through’ the fire. Specifically, the “older” particles should be more see-through that the newer ones.

1. (50 marks) In game and science fiction, there is a concept of a “portal” which is like a door or window into another, alternative world. The idea is that this world “overlaps” our world, but you can’t see it. The game “portal”, the “Stargate” move and TV shows, and the book “The Subtle Knife” and others make this one of the central themes.



The image on the left is like a “Stargate” portal, where you walk through and go to another part of the galaxy. The one on the right is from the original Star Trek, where the portal allows you to go to a different time.

Note that you can look through the portal to see what’s going on.

You are to create two scenes of bunnies, one in the “daytime” and one in the “night time”.

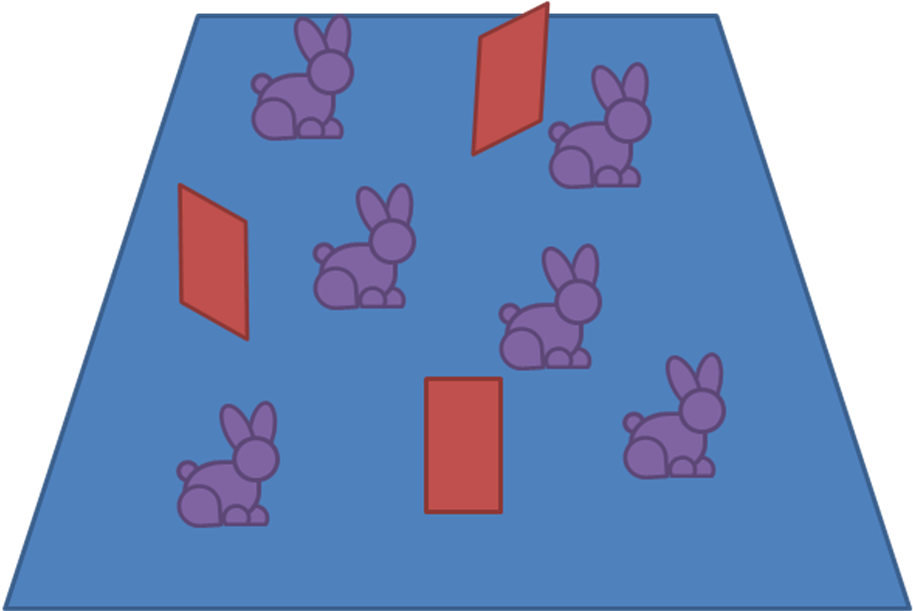
You can use the same scene you used in previous questions, or you can create a new scene.

Details:

* Like question 1, you need 20 bunnies, and a “ground” object of some sort
* You need to have TWO different scenes, one with a number of bright lights (the “daytime” one), and one with either very dim lights or only one, dim light (the “night time” one).
* The bunnies in the “daytime” scene have to be in *different* places from the “night time” scene.
* Place five quads or blocks (cubes) in random locations and angles (rotations around the Y or up-down axis) in the scene. These will be the “portals”.
* Using the stencil buffer, show the “night time” scene, but display the “day time” scene through the “portals”.
* You should be able to move around the scene with the mouse or keyboard.

The image below attempts to show this:

* Note that the portals are facing random directions.
* You would be able to see the “other” scene through these portals



That’s it.