# **INFO3111 “C++ Graphics”, Final Exam, Thursday, June 20th, 2024**

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 or if anything “pops up” on your computer, I see a browser, etc. I’d suggest turning off your wi-fi and putting your phone in your pocket/backpack/purse/whatever.***

* The questions are ***NOT*** of equal weight and don’t add 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", will be marked to a higher standard.
* The exam has **six (6)** questions and **nine (9)** pages.
* **CLEARLY** indicate which answer goes to which question.
  + If you feel that the questions “build on each other” (which they do) feel free to submit a single solution/project.)
  + If you feel it’s clearer to have the submissions separate, I’d suggest placing separate questions in separate projects and folders.
  + If I can’t make heads or tails of what question is what, I probably won’t even mark it.
* 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 run the code as submitted.
* 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).
* Submit the **entire solution(s) and project(s)** *not* just the source code files.
* Please, *please*, ***please*** delete any files you don’t need as well as any temporary files Visual Studio creates (“.vs”, the debug and release folders, etc.). You won’t lose marks for submitting these, but it’s very annoying and it’s something you should know how to do. **But**, give me the **ENTIRE SOLUTION**, not just the source files.
* **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 was a pretty forgiving on the mid-term and project, I ***won’t*** *be* for this final; by now, you should know what’s expected, and how to submit your *entire* solution so that I can simply download, un-compress, build and run it.

Make sure the files aren’t pointing to some random location on *your* hard drive, for example. If that happens, I likely just won’t mark it.

## NOTE: Unless otherwise indicated, you should be displaying the ply files that were included with the exam (in the Simple\_Space\_Interiors\_SourceFiles.7z and Simple\_Space\_Interiors\_SourceFiles.zip folder – they contain identical files).

## The Questions:

You are to create the “hangar” of a space station, using the models from the “Simple Space Interiors” asset pack (the converted ply models are in the compressed files): <https://assetstore.unity.com/packages/3d/environments/sci-fi/simple-space-interiors-cartoon-assets-87964>

You will use the **ModelsToUseINFO3111SummerFinal2024.exe** program to determine these.

The models use three (3) textures: SpaceInteriors\_Emmision, SpaceInteriors\_Texture, and WorldMap. Most of the models use the SpaceInteriors\_Texture file.

**The “hangar”:**



**Some notes about the models:**

* They are aligned in one corner of the model. This allows you to “snap” them together more easily (MeshLab “Render”, then “Show Axis” will show this.)
* They are designed to fit into 5x5x5 blocks, or multiples of that (MeshLab “Render”, then “Show Box Corners” will show this). For example:
  + SM\_Env\_Transition\_Door\_Curved\_01 is 5x5x2.5 in size
  + The SM\_Env\_Construction\_Block models are all 5x5x5 in size
* They assume “Y” is “up” as we’ve been using.

You **MUST** apply the appropriate textures to the models or risk losing at least 50% of the marks.

**Some notes about the WhatModelsShouldIUseINFO3111SummerFinal.exe program:**

* This isn’t some work of amazing art, so there’s no error checking in it because, seriously?
* Now, if you enter correct info, it will work, but if you enter stupid nonsense, then it likely won’t.
* It will generate a file called **INFO-3111\_2024\_log\_XXX.log**, where “XXX” is a number based on your student number and certain information from your system (like the systeminfo generates) and state information (date, time, MAC address, etc.). The square serial number thing is part of this, too – it’s unique to your student number, on your system, at that time.   
  Note this does **not** include any personal information – it’s just hardware, OS, and location stuff.
* This is the same output that you see on the screen.

**You MUST include this log file with your submission.**

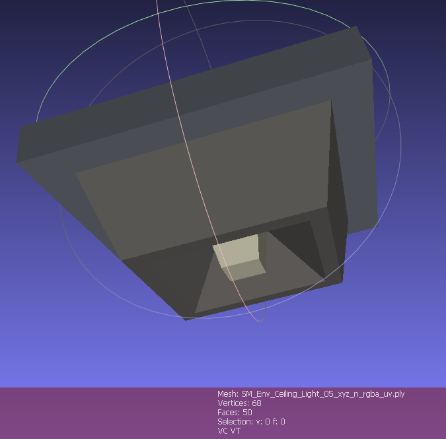
**If you do NOT include this log file, your exam will NOT be marked (and you will get a mark of zero).** If you screw up the files, then regenerate

Create the main hangar/lab:

1. (100 marks) Make the floor, ceiling, and walls.

* This should *mimic* the 1st image in the exam (page 3), in the overall shape and size, ***not***what’s placed inside it (that’s done in a later question).
* It is 5 “blocks” (**SM\_Env\_Construction\_Wall** model short axis) high.   
  These models are 5x10 in size (5 units high and 10 units “long”).
* The floor is 8 x 12 “floor panels” (**SM\_Env\_Floor\_01** model) in size.  
  These floor panels are 5x5.
* Make all four walls, including the wall that’s “behind” the camera in the image.
* In the picture, notice that one of wall panels has a yellow “06” number on it.   
  + This is the SM\_Env\_Construction\_Wall\_02 model.
  + The yellow number is clearly visible in the SpaceInteriors\_Texture texture.
  + Make copies of this texture and edit two (2) of the panels to show your eight (8) digit student number, adding “leading” zeros (0) if your student number has fewer than eight (8) digits.
  + Four (4) numbers should appear on each of the two (2) panels.   
    (for example, say your student number was “**263**7876”, then one panel will have “0263” and the next one would have “7876”).
  + These panels should be at the same height as in the picture, on the 3rd panel from the floor.

1. (200 marks) Add ceiling lights.

* Place an appropriate number of ceiling lights in the room, using the **SM\_Env\_Ceiling\_Light\_05\_xyz\_n\_rgba\_uv.ply** model.   
  + You will need at least **six (6)** of these.
  + Each one of these will be associated with a spotlight.
  + The spotlight will be located at the model location, and will face downward (i.e. pointing directly down to the floor).
  + **Five (5)** of the lights should make a **gentle penumbra** on the floor, but **one (1)** of the lights should have a **hard edge** to the spotlight.
  + **Add one more dim general light** (either a directional light or a large, dim point light) so that the rest of the scene is somewhat visible. i.e. The spotlights should look like the source of the lights, and you should see the “circle” of light on the floor, but there should be enough ambient light spill (from this light) to see everything else, just dimly.

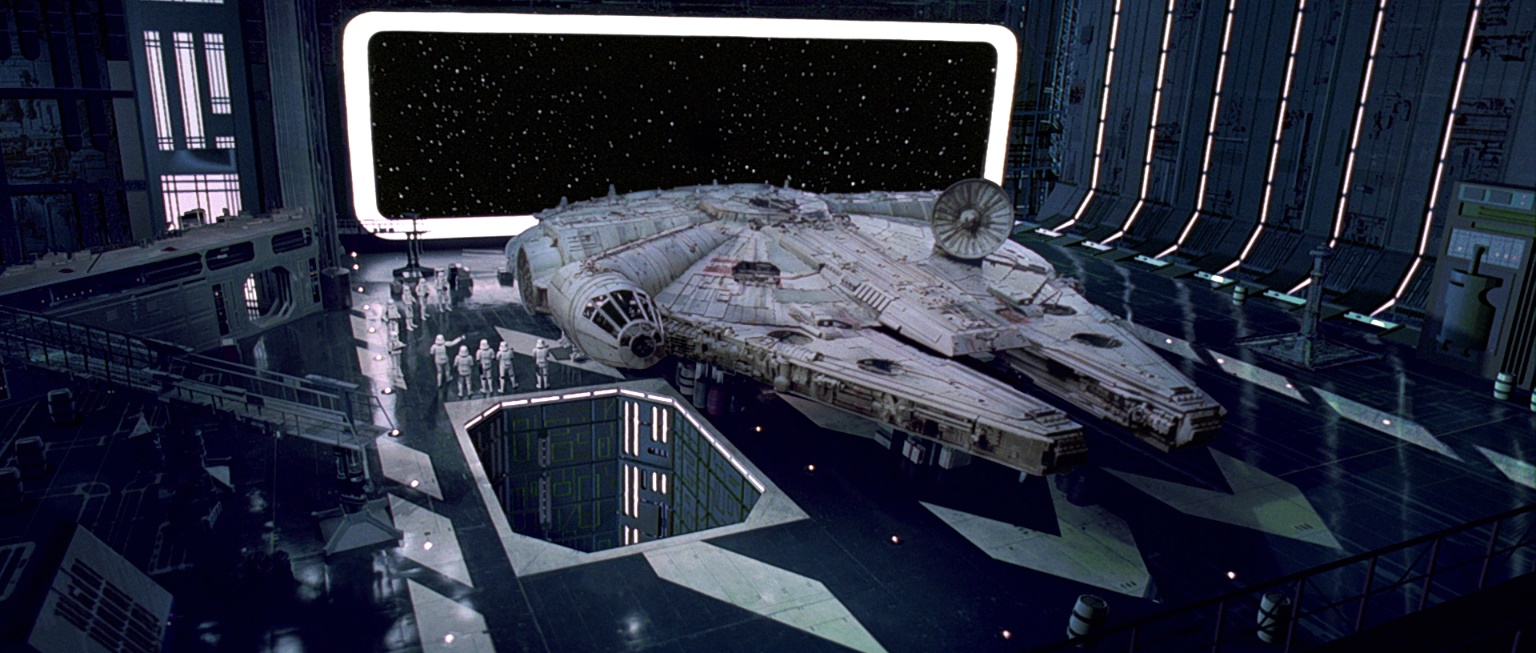
1. (100 marks) Add the props listed by **ModelsToUseINFO3111SummerFinal2024.exe**:

* The items should be “appropriately” placed, meaning that they are deliberately placed in a “sensible” location. So, lockers and server racks would go against the walls, phones on desk, chairs on the floor, etc.   
    
  If they are “stupidly” or randomly placed, then you won’t get marks for this.
* You all have the same last four items (Shittle, Lander, Simulator, and Rocket), which are quite large. You can place them where you’d like – they *don’t* have to place them at the same location as in the picture if you don’t want to.  
    
  Note: You might have duplicates (like you might get another rocket, etc. from the list of items to be placed). In that case, you’ll need to place the duplicates, too.

1. (100 marks) Place some semi-transparent “beakers” (glass containers):

* Add one of the “bigger” (SM\_Prop\_Desk\_01 and SM\_Prop\_Desk\_03) desks to the scene, somewhere away from the walls and at an angle.
* Place four (4) SM\_Prop\_Beaker\_01 models on the desk and make them semi-transparent.
* These should appear transparent regardless of the placement of the camera.

1. (200 marks): Open the hangar doors.



Stars

* + Pick one of the walls to be a giant door that opens to space. In the picture above (from Star Wars), there isn’t a physical door, but we can imagine it’s “open” and you can see out into space – see the stars, etc.
  + There are two (2) ways you can do this:  
    - Like a giant barn door, where the entire wall moves as one unit.
    - (BONUS: +100 marks) Like real hanger bay doors on Earth, where they sort of “collapse” into each other, like in the “Telescopic Hangar Door.mkv” video.
  + Choose one key to “open” the door(s) and one to “close” them:  
    - The wall should *slowly* move along, revealing the space outside. This should take about 5-10 seconds, but the time isn’t *too* critical – however, it shouldn’t slide across in a split second or something (the doors are *way* bigger than the shuttle, so they must be pretty heavy, right?)
    - You do this by adding a small offset to the doors each frame.
    - You can either test that the door has moved to (or past) a certain point, or you can use some kind of time (you could even just increment a counter each frame and make note of the count it is when the doors have moved “enough” and make note of that).
    - You can either allow the “opposite” key to “interrupt” the movement, or “lock” them until the door has completed its motion. In other words, if I press the “open” key, and while the door is moving, I press the “close” key, you can either start moving the door the other way or ignore the keystroke until the door is completely open.
    - NOTE: I do **NOT** want to *hold down* the key to do this. It should be a single press to start the entire open/shut process. Also, the door(s) need to stop at the correct location, not fly into space or whatever.

1. (150 marks): Show “space” outside.   
   * Place a whole bunch of ***very*** tiny little white spheres outside. See the arrow on page 7 pointing to the stars outside the giant door. They are so small you can hardly tell they are spheres, right? So if I see a bunch of large balls, that’s not what I’m looking for.   
       
     Pro tip: You can make a function that just randomly places a while lot of tiny little spheres outside the hangar. Like run a loop that adds, say, 100,000 “stars” by picking a location, and inside the loop, check to see if this location is “too close” the origin (i.e. inside the hanger is definitely too close, right?). If the location is too close, *don’t* add it to the scene.   
     Every once in a while it’ll pick a “bad” location (too close), but with enough iterations through the loop, there should be “enough” stars.   
       
     How many times through the loop?? (you ask) I dunno, pick a number and see what it looks like. If it’s not enough, increase the number of loops.   
       
     How many stars?? (you ask) A comparable number so it looks like the picture on page 7.

**BONUSES:**

* + (+5%) Add the space skybox we used in class (which has nebulae and stuff as well)
  + (+5%) The space station is spinning, so slowly “rotate” the stars (and the skybox if you did that as well), mimicking this effect.
  + (+5%) Add two (2) of the “SM\_Env\_Ceiling\_Light\_04\_xyz\_n\_rgba\_uv.ply” models (a red and green indicator light) to the walls beside the door. While the doors are opening and closing, gently “pulsate” the brightness of the lights.
  + (+5%) Update these red and green lights so that they mimic rotating warning lights instead. You’d do this by placing spotlights at the same location and rotating them around. Think about old school police lights which rotated around.   
    (If you do this, the lights don’t have to slowly “pulsate” the lights – unless you want to.)

**That’s it.**