# **INFO3111 “C++ Graphics”**

**Midterm Exam #1, Friday, July 22nd, 2022**

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 proceed accordingly.*

* The questions are ***NOT*** of equal weight. The exam has **four** **(4)** questions and **six (6)** 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 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. The except 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. 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.
* **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 through 2019 using the OpenGL 4.x API (with GLFW, glad, and GLM).
* Your solution may ***not*** contain any third party libraries (like boost) or the following C++11 (or later) elements:   
  + Smart pointers
  + auto
  + Lambda function
  + Initializer List (other that the C/C++98 int array)
  + Assignment of member variables inside the declaration of a class

To be clear: if you have ***any*** of these elements, you will likely 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. However, if it sort of looks like you did it by accident/under the “stress of the exam”, then it’s possible I’ll ask you to change and resubmit it; it’s the situations where you seem to be well aware of what you’re doing, and are using these, anyway – those students will get zero. Imagine if I was your boss, and this was a requirement, and you ignored it... (I’d warn you, then fire you).

* 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

## 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).

There are five (5) models you can use:

* at-at.ply
* Isoshphere\_InvertedNormals\_xyz\_n\_rgba\_uv.ply
* snow\_terrain.ply
* Snowspeeder.ply
* TearDropBullet\_inverted\_normals\_xyz\_n\_rgba\_uv.ply

## The Questions:



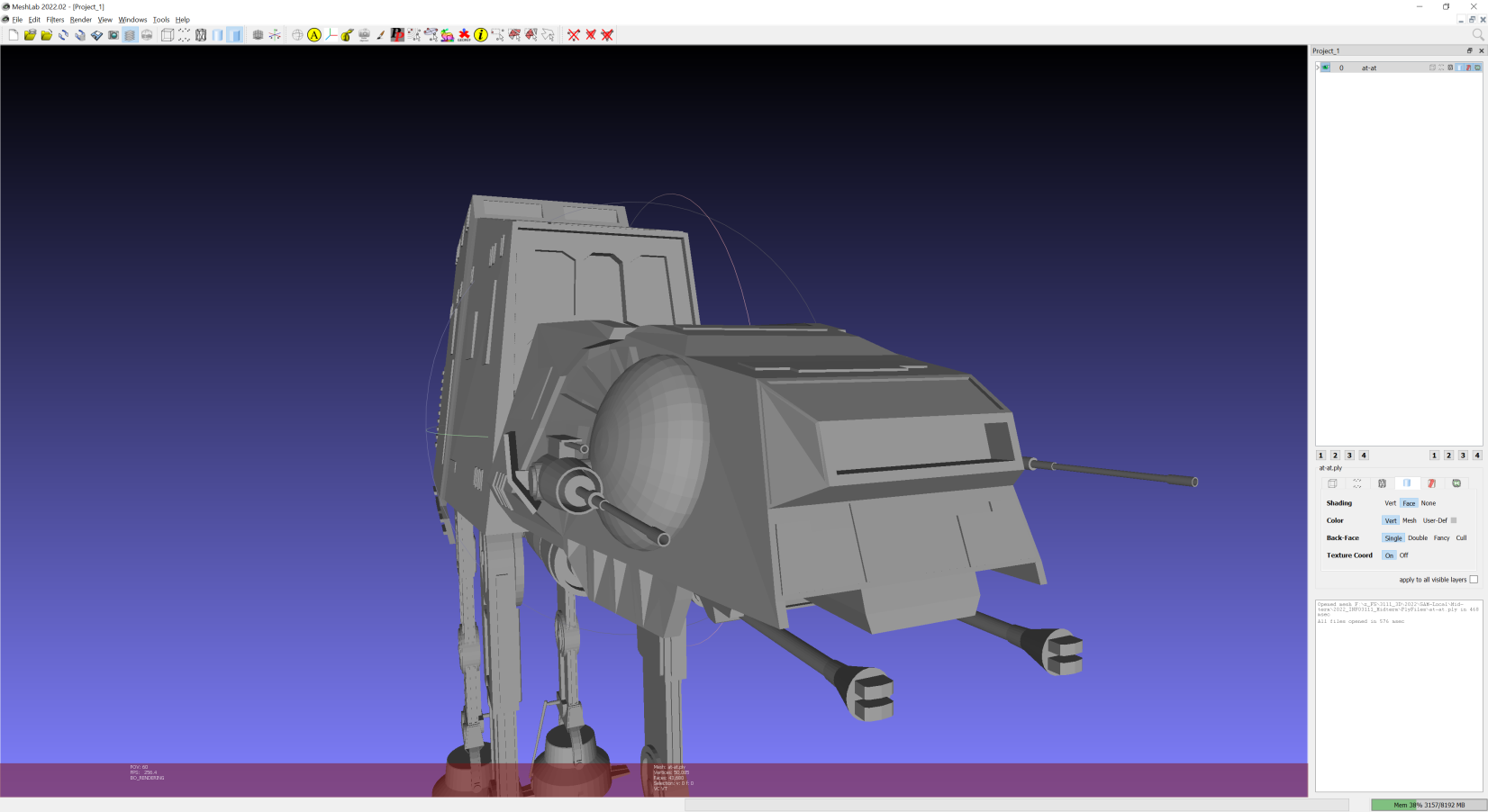
(From: <https://cdn.vox-cdn.com/thumbor/bgaFTjV7tU65h8wC15YO8onjzZk=/217x89:1172x626/1600x900/cdn.vox-cdn.com/uploads/chorus_image/image/56334983/at_at_2.0.jpg>)



In case you don’t know anything about Star Wars, here’s what you are seeing in the above image:

* The AT-ATs are the giant robot elephant things. They walk along, and can shoot LASERs (or whatever) out of the “guns” underneath (and to either side of) the “head” portion.   
  You can also see the red glowing “cockpit” in the front of the AT-ATs (this is where the pilots/drivers sit I’m assuming).
* The Snow Speeders are the other, flying ships (to the left). They have to “guns” located on either side of the cockpit.
* You can see the LASERs/bullets/whatever-you-want-to-call them being shot from either craft.

You are to re-create glorious OpenGL non-textured 3D scenes based mainly on this video (from the actual movie): <https://www.youtube.com/watch?v=s5-ccUfW9eI&ab_channel=StarWarsUK>

1. (50 marks) Create a scene like the image **at the top** page 3.   
   * Use the “terrain” model as the “ground” (it’s snow, so should be white)
   * Place five (five) AT-ATs at approximately the locations they are in the image. The feet of the AT-ATs should be touching the ground.
   * The camera should be near the ground, looking up at the AT-ATs, and far enough away to match the image.
   * Make sure you have them somewhere that has a “mountain” or “hill” behind them (the terrain is pretty bumpy, but you may have to scale the terrain to a larger size to give a decent effect).
   * There should be enough white light so that both the tops and the bottoms of the AT-ATs are visible (from sunlight above, and the reflection of the light, off the snow, below). You can do this by placing a bright light high above the entire scene, and a less bright light far below the ground (to light up the underside the AT-ATs). This 2nd “underground” light should be very far away, too, and not at bright.   
       
     You’re trying to get the effect in the top picture, where the underside is somewhat lit/visible, but not nearly as bright as the sunlight from above.
   * Change the clear colour of the background (it is default black, though we changed it to a “blue-ish” colour in the in-class example), but setting the glClearColor() to something “light grey” – sort of like a very cloudy/overcast sky (like the image has – i.e. the sky isn’t black or blue, but a grey colour, like it’s about to snow or rain or something.)
2. (50 marks) Make the cockpits of the AT-ATs glow with red lights:
   * Notice that the front window things (where the eyes might be) of the AT-ATs have a slight red glow. You’re trying to mimic this.
   * To do this, place one (or ideally two) very small bright red lights right in front of the “windscreen” area (indicated by the white arrow here 🡪
   * Make sure very little red light “spills” into the rest of the scene, particularly the snow. You might not get it “perfect” – like some might spill onto the little gun points sticking out, but it shouldn’t colour the AT-AT red.
   * You get ten (10) marks per AT-AT you illuminate.
3. (50 marks) Place the Snowspeeders in formation, flying towards the AT-ATs :
   * At around 48 seconds of the video (on Page 4), you can see the Snowspeeders flying over the trench in formation. From the video, it seems like they stay (more or less) in that formation as they fly towards the AT-ATs.
   * You are trying to mimic the shot at the 50 second mark:



* + In the image above, there are only four (4) Snowspeeders, but it looks like there’s about eight (8) of them. Place about eight (8) of them in this sort of formation.
  + You need to mimic the various angles the ships are in the scene. i.e. they aren’t flying level to the ground, but are twisting and going up and down all the time – you are trying to capture a moment in time similar to the image above.

1. (10 marks) Using the keys, change the location of the camera in the scene:  
   * Pressing the “1” key will move the camera to the position it is in question 1
   * Pressing the “2” key will move the camera as it is in question 2 (behind the snowspeeders, and far away from the AT-ATs).
   * Pressing the “3” key will move the camera to near the cockpit of the AT-AT, looking at the *farthest* Snow Speeder approaching in the distance.
   * Pressing the “4” key will move the camera high above the scene (imagine it’s a shot from a drone with a camera), looking down at the entire scene:
     + You should *not* be able to see the edges of the terrain (I’d suggest making the terrain really, really large so this can’t happen – and/or place the camera carefully and remember that I’ll be maximizing your window!
     + You should be able to see all the ships and AT-ATs.

That’s it. The exam is over.