# **INFO3111 “C++ Graphics”, MAKE-UP Midterm, Tuesday, June 10th, 2023**

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 **seven (7)** questions and **six (6)** 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, for all that is good and sane in the world, 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.
* **If the solution does not build (and run), I will not mark it** (so y*ou 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 *might* use the Debug build while marking, your submissions **must** build and run in **Release** build.
* Unless otherwise indicated, all these solutions assume that you are creating/using a C++ project using Visual Studio 2022 using the OpenGL 4.x API (with GLFW, glad, and GLM), and build in **64-bit** **Release** using the default setting in Visual Studio (C++ 14).
* Your solution may ***not*** contain any third-party libraries (like boost), **smart pointers,** or the “**auto**” keyword.

To be clear: if you have ***any*** of these elements, you will 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.

**Some notes about the models:**

* Most of the models are same as the ones as the original mid-term: **Dungeon\_models.7z** (from project #1), and **Additional\_Mid-term\_Models.7z**.
* You will also need the **Terrain\_to\_Island\_Converter*\_2*.7z** file.

***If you have not downloaded these files from github or FOL, DO THIS NOW and GET OFF THE NETWORK – if I see you on github or FOL and you are not uploading your exam, I will assume you are cheating – see page 1.***

***That also goes for help or any other webpage that I see. Close your browser(s).***

*You are going to make a scary cave in the side of an island mountain.*

## The Questions:

***NOT*** the default!

1. (5 marks) Making the island:

Using MeshLab, generate a SINGLE island, in the following way:

* Open MeshLab (without opening a model). This will open it with an empty “project”
* Choose “Filters”, then “Create New Mesh Layer”, then “Fractal Terrain”
* In the “Fractal Terrain” dialog box, choose “**Hybrid multifractal terrain**” (“Algorithm” dropbox.)
* A white mountain with a blue background

  Description automatically generated with medium confidenceChange the “Max Height” to **0.5**.
* ***Use the first 4 numbers of your STUDENT NUMBER for the “Seed” value (the default is 2.0).*** Note: It seems that really large numbers (like your entire 8 digit student number, make the terrain ‘blocky’)

With a “Seed” value of 2.0, you will get this 🡪

MeshLab assumes that “up” is “z”, so we need to adjust this. Turn on the “axis” drawing by choosing “Render”, “Show Axis” to make this clear (if you want).

A close-up of a mountain

Description automatically generated

* Choose “Filters”, “Normals, Curvature, and Orientation”, then “Transform: Rotate”.
* Type in “-90” in the “Rotation Angle”, leaving the “Rotation on:” set to “X axis”, and click “Apply”, which will get you something like this 🡪   
    
  *(Note that you might have to rotate the window so the Y axis (the green line) is facing upward, which isn’t the default I think...)*
* A 3d model of a mountain

  Description automatically generatedSave this model with **JUST xyz** and NOT in binary form (“File”, “Export Mesh As…”, uncheck the “Binary encoding”, and choose OK.

A screenshot of a computer

Description automatically generated

🡨 Like this, with **nothing** checked.

* Either build the “**Terrain\_to\_Island\_Converter\_2**” project (or use the **Terrain\_to\_Island\_2.exe** file in the x64/Release folder) and convert the mesh you made into an island.   
    
  This takes the model file name as an command line parameter and generate an “island” mesh (like the one on the right 🡪) called “output.ply”.   
    
  You can also drag the mesh file onto the exe file.
* You will use this model (output.ply) as the island cave is in.

|  |
| --- |
| ***This is a different tool than the original mid-term:***   * ***The island is WAY bigger (proportional to the 500x500 dungeon tiles)*** * ***It has UV coordinates.*** * ***Do NOT use the original island model generator!*** |

1. (5 marks) Generate your “cave” layout:  
   * Run the INFO3111\_S2024\_IslandCaveMaker.exe file. This is similar to the tool you used to generate your “dungeon” for project #1.

XXXXXXXXXXXXXXXXX

X.......XXXXXXXXX

X.@...@....XXXXXX

X.......XX.XXXXXX

X.......XX.XX....

X.......XX.XX.XXX

X.@...@.XX....XXX

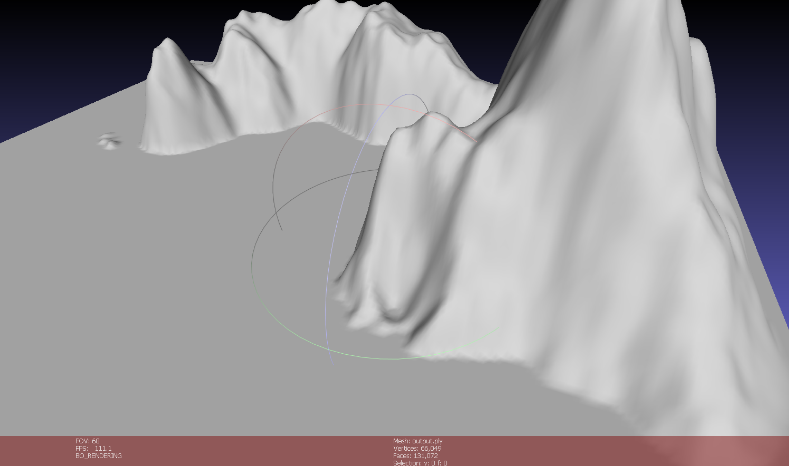
X.......XXXXXXXXX

XXXXXXXXXXXXXXXXX

* + It will generate a series of “caves” that look like this 🡪
  + Like project #1, the “.” characters indicate a “path” through the mountain.
  + There is an “entrance” to the cave on the right side (green arrow).
  + Every cave leads to a large “room” that has four “crystals” in it.  
    (indicated by the “@” symbols)
  + Be sure to submit the .log file that is generated!

This large flat area is the “water” part of the model.

1. (50 marks) Choose a location for your “cave” and place the entrance.



* + The idea is that you can enter the cave from the water or from the beach, so it is right at the waters surface (or very close).   
      
    i.e. it’s not half way up the mountain or something.
  + The circle here indicates a reasonable location.
  + This “version 2” island generator makes much, much larger islands that are proportional to the “dungeon” models.
  + Use the “SM\_Env\_Dwarf\_Wall\_Archway\_01.ply” model to mark the entrance to the cave.   
    Place this model near the surface of the water (or on the beach) partially into the side of the mountain. Note that this will overlap the geometry of the island model, but don’t worry about this.
  + Remember that the rest of the cave will “attach” to this entrance, so be sure to take into account the orientation of the cave.
  + Choose a reasonable colour for this cave. I’m assuming it’s made of stone or stone blocks, so pick some grey “stone” or “dirt” colour.

1. (200 marks) Place your cave in the mountain:   
   * Like Project #1, your “cave” is made up of rectangular floor and wall tiles.

XXXXXXXXXXXXXXXXX

X.......XXXXXXXXX

X.@...@....XXXXXX

X.......XX.XXXXXX

X.......XX.XX....

X.......XX.XX.XXX

X.@...@.XX....XXX

X.......XXXXXXXXX

XXXXXXXXXXXXXXXXX

* + You do *not* have to fill out all the “X” locations in the cave (but you can if you’d like); I’m only concerned that it looks like a cave *when you place the camera inside the cave.*
  + Choose a single floor and wall tile style for the “corridor” that leads to the large room. i.e. all the tiles should be the same style.
  + Unlike project #1, you are placing a ceiling on this cave.   
    You do this by inverting the floor tiles and placing them at the roof/ceiling.
  + I need enough light to see the cave coordor, but it should be “dim” (it’s a cave, right?)

1. (100 marks) Place the large room behind your cave.

* Choose *different* floor and wall tiles from the corridor (question 4).
* The room is 7 x 7 tiles in size. i.e. there’s 7 x 7 floor tiles surrounded by wall tiles.
* I don’t care how “thick” the walls are – I’m only interested in how the room looks when inside it.
* This room also has a ceiling.
* The room can be any “height” you’d like (1 wall tile, 2 wall tiles, 5 wall tiles – don’t care).

1. (300 marks) Place your magical glowing crystals:   
   * Pace each of the four (4) types of “large” crystals, one at each “@” location in the map.   
     (SM\_Env\_Crystals\_Cluster\_Large\_01.ply, etc.)
   * The crystals should be “glowing” and have a high specular value:
   * Choose a bright colour for the crystal colour (to simulate an emmisive light).
   * Place a small, bright light inside the crystal of (more or less) the same colour as the crystal.   
     The idea is that they crystals are glowing with a dim light, lighting up the room near them.
   * It’s important that the colours of the crystals “match” the colours of the crystals light.
   * It’s also important that the lights aren’t “too” bright – it’s a cave with gently glowing crystals illuminating the room, not the surface of the sun or something.
   * I need to *clearly* see each of the crystals and the colours, like that they are gently glowing near the crystals (so the floor and/or wall nearby is lit up) and that they each have their own coloured light inside them.
   * Make sure the rest of the room has a very dim light (like the corridor coming into the room).
2. (200 marks) Place the camera at various locations to show off your masterpiece:
   * F1 should place the camera facing towards the entrance to your cave, somewhere on the water nearby. Imagine you are on the water approaching the cave, like about 5-10 “car lengths” (100 feet or 30 meters). Like far enough that you can clearly see the cave, the side of the mountain, etc. but not “right at” the entrance of the cave.  
       
     It's day time, so make sure there’s enough “sunlight” in the scene to see.

XXXXXXXXXXXXXXXXX

X.......XXXXXXXXX

X.@...@....XXXXXX

X.......XX.XXXXXX

X.......XX.XX....

X.......XX.XX.XXX

X.@...@.XX....XXX

X.......XXXXXXXXX

XXXXXXXXXXXXXXXXX

* + F2 places the camera inside the corridor at the 1st turn. Imagine you are inside the first part of the corridor, have just noticed the turn, and are looking down the next corridor to see if something is there.  
    (The blue arrow is where you are and where you’re looking)  
    You should clearly see that this is a corner.
  + F3 places the camera in the last straight stretch of corridor, like you’ve just turned the last turn and see that there’s a large room full of magical crystals!
  + F4 places the camera above the entire cave, showing the corridor and the glowing crystals.   
      
    To do this, you will also have to make your ceiling invisible.   
      
    You can either hope that the back-face culling doesn’t draw the ceiling tiles, or you can se these tiles to “inviable” (not being drawn).   
      
    Note that they have to “reappear” when looking from another location (pressing one of the other F keys)

*That’s it.*