

# INFO-3111 “C++ Graphics” Final Exam, 2023 (Friday, June 16<sup>th</sup>, 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](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 **six (6)** questions and **nine (9)** pages. The questions involve submitting a working Visual Studio solution.

- The questions build on each other, to make a complete scene. However, you may decide that a different camera angle will better show the scene, so:
  - You are to submit a single solution (not multiple solutions/projects)
  - Use keyboard controls to best show the appropriate question, so pressing “1” will place the camera to best show question 1, “2” to show question 2, etc.
  - NOTE: you do not *need* to do this, but it's very likely that it's to your advantage.

**Do not** comment out your code, or expect me to alter it in any way at all. It should be ready to run as is.

- 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.

- 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 (part of knowing what you’re doing is knowing – or figuring out – what files are needed and what aren’t.), like the “Debug” and “Release” folders with the “obj” and log files, as well as the hidden “.vs” (intellisense) folder – this will save a tremendous amount of space and shorten your upload time..
  - **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.

If you’re missing some common stuff that git doesn’t like, like the lib files, it’s no big deal (I can unzip those), but if you’ve adjusted to the project/solution – like placing the includes/libs in a different folder – *and* it doesn’t build because you’ve forgotten to submit these, that’s pretty annoying...

- **No “auto” or the boost library** (or any other 3<sup>rd</sup> party library we haven’t used).  
If you use either, I will not mark your submission and you will receive a mark of zero.  
No exceptions.  
This is a trivial restriction – it takes almost no effort to *not* use these items, and there is no justification for “needing” to use either in INFO-3111 (or any other C++ development for that matter, but I digress).

PLEASE make sure you give me your entire solution, so that I can simply download, un-compress, build and run it.

Having said that PLEASE remove the things you don’t need, for example:

- I don’t need *the entire github repository* including the lectures, etc.
- I don’t need the “.vs” folder (which can be gigabytes in size)
- I don’t need *every single model you’ve ever seen*, or downloaded, or whatever.  
Do I need the “Galactica” model? The “fractal terrain”? The “hockey player”?  
Seriously? Delete these massive things, please.

### ***The Models and textures:***

- *The vast majority of the models are the ones in the “Dungeon\_models” folder that’s been on github for a few weeks. In other words, if you’ve pulled from git in the last week or so, you’ve got those models.*
- *The textures are under the “textures” folder inside the “Dungeon\_models” folder.*
- *There is an additional “Dungeon\_models (additional)” compressed file that has some additional models you’ll need. It’s only the models that are missing from git, so if you unzip this folder and overwrite the “Dungeon\_models”, it will only add the missing model files.*
- *The 7z and zip files have the same content.*

## The Questions:

You are to create a small dungeon made up a several large open spaces and smaller corridors.

You will use the assets taken from the Synty Studios “POLYGON - Dungeon Realms”

(<https://syntystore.com/products/polygon-dungeon-pack>).

Many of the “environment” assets can be combined like LEGO bricks into any shape you’d like. They are all proportional sizes. In the picture below, you can see there’s a bunch of different “floor” models as well as many “wall” models.



When combined, you can make rooms like the one below. Also note that the small corridor is also made of floor and wall models:



To get the layout of your dungeon, run the **WhatModelsShouldIUseINFO3111SummerFinal2023.exe** program. This will generate five (5) variations of dungeons and you can pick the one that you'd like. **Note that you need to submit your log file**, too.

(this log file is the same output as on screen and is just a text file).

#### **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.)
- The textures that correspond to the model are shown in the ply file, for example, the SM\_Env\_Floor\_Grate\_01.ply file has the following:

```
comment TextureFile Dungeons_2_Texture_01_A.png
```

This means that you can use the Dungeons\_2\_Texture\_01\_A.png file as the appropriate texture.

#### **Some notes about the WhatModelsShouldIUseINFO3111SummerFinal2023.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\_log\_XXX.log, where “XXX” is a **number that's generated based on your** student number (but it's not your student number – it's to catch sneaky cheaters). This is the same output that you see on the screen.

**You MUST include these log files with your submission.**

**If you do NOT include these log files, your exam will NOT be marked (and you will get a mark of zero).** If you screw up the files, then regenerate them – the values should be the same and is based on your student number.

As an example, assume that your log output on the right →

Note: you will use *your own* dungeon layout **\*NOT\*** this one.

1. (50 marks) Create a floor and outside wall.

- The “X” are walls or pillars and the “.” are open areas (i.e. they are just floor)
- Note that the wall and floor models are not the same thickness. In other words, the “X” characters around the outside are much thinner.
- In this question:
  - Make a large open space with at least six (6) variation of the “floor” models (in the “Floors” folder).
  - Place walls around the outside using at least six (6) variations of the “wall” models (in the “Walls” folder).
- Note that there’s one or two entrances to the room (a break in the “X”s around the outside. In the example, you can see this at the bottom and top.
- Add enough evenly lit lighting to see everything. This should be “bright enough” but not “too bright” – it’s an underground dudgeon, right?
- Apply the appropriate textures to match the models.
- There should be at least one (likely two or more) larger “open” areas in the dungeon with no walls or pillars. In the image above, the red square is in such an area.

```
XXXXXXXXXXXXX.XXXXXXX
X...X.X...X...X
XXX.X.X...XX.XXXXXX
X..X.X.....X
X.XX.....XXXX.X.X
X.....X...X
X.....X...X
X.....X...X
X.XX...X.....X
X.....X...X.X
X...XXX...XX...X.X
X.....X.....X
XX..X.....XXX
X..X.....X
X..X.....X.X
X.X.X.....X.X
X.X.X.....XXXXXX.X
X.....X
XXXXXXXXX.XXXXXXXXXXX
```

Make the floor in this area *noticeably* sunken (i.e. lower than the rest of the floor. In a later question, this area will become a shallow “pool” filled with water. It should rectangular be *at least* 3x3 squares in size. If you feel that you need “sides” to your pool, then use more “wall” tiles.

Note that the “walls” can also be made like the pillars, like four (4) walls wherever there’s an “X” will give you walls. i.e. you don’t *have* to make the walls thinner than the pillars.

However, if you want something fancier, see the “bonus” questions later in the exam.

2. (100 marks) Fill in the rest of the walls inside the room/dungeon.

- The “X” inside can represent walls or pillars/columns.
- If there’s a few “X”s in a row, then it’s a wall.
- If there’s an “X” by itself (i.e. not touching another “X”), then it’s a pillar or column.
  - Note that if an “X” touches another “X” *at a diagonal* then it’s not connected.  
i.e. it’s not part of the wall or it’s an adjacent pillar.  
There aren’t any “diagonal” walls.
- Walls can be placed right on the floor (or slightly penetrating if that looks better).
- Pillars/Columns are made of four (4) wall models in a square, the size of a floor model, making a square shape, with the “backs” of the wall models facing towards the inside.
- Apply the appropriate textures to match the models.

3. (150 marks) Place several lit “torches” in the scene:

- Choose one type of “torch” (light) model in the “Torches” folder.
- Place six (6) of these torch models spread out evenly in the scene, attached to the walls, near the top. The idea is that they are torches with fire or candles, attached near the top of the wall.
- Place small, bright red-orange lights where the fire/candle of the torches would be. These should light up the area nearby (like a small candle or flame would).
- Make the torches “flicker” by slightly perturbing the linear attenuation each.

You do this by first setting up the “regular” (or non-flickering) attenuation you’d like.

Then, for each frame, adding a small random number to this “base” attenuation of the light and passing *that* (i.e. total) to the shader. Don’t update the original “regular” value, though. The idea is that you’re *passing* this slightly adjusted value each frame, not *saving* it.

Make sure this small random number is negative as well, so when you are “adding” this small value, the torch can get dimmer as well (adding this small negative number makes it slightly dimmer, while adding the small positive number makes it slightly brighter).

4. (100 marks): Place several semi-transparent crystals in the scene.

- Place at least five (5) of the “crystal” models (in the “Crystals” folder) throughout the scene. These should look like they have randomly been scattered or dropped or were just naturally there. Like they shouldn’t be dropped together or in a line, etc.
- Make the crystals semi-transparent (so you can see through them).
- The transparency should be “order independent”, in that I can move the camera around and see all the crystals through all the other crystals.
- Make the crystals very “shiny” (i.e. have a very high specular component)

5. (150 marks): Place a few plants that are growing in “holes” in the ceiling.

- Place three (3) “groupings” of plants in the room.
- Each “grouping” consists of a single tree with a few smaller plants around them.
- They should be scattered around the dungeon.
- To mimic the “holes in the ceiling” by placing spotlights shining down on these plant groupings.
- The spotlights should be large enough to light up all the plants in each grouping.
- Point the spotlights at various angles, so that the light on the floor should be elliptical in shape (rather than circular, which it would be if the spots were *directly* above).
- They should also be different elliptical shapes.
- You can do this by *not* placing the spot lights directly above the plants, but having the spot direction pointing at the plants.

6. (150 mark): Fill the pool with water.

- Use either the FractalTerrainFromMeshLab model, or generate your own (in MeshLab, “Filters”, “Create New Mesh Layer”, “Fractal Terrain”, choose the “Algorithm” you’d like from the drop down menu. Note that this will be really large, with “up” being the Z axis, and the origin at one of the corners (not in the centre).
- Apply a “water” or “pond” texture to this mesh (from the “Dungeon\_models (additional)” or if you have an appropriate texture on your computer, you can use that).
- Make the surface of the water semi-transparent so you can see through it to the bottom of the pool underneath. (don’t worry if this makes the lily pads semi-transparent, too – let’s assume these are magical lily pads or something...)



## BONUSES:

7. (100 mark **bonus**): Mimic the time of day changing.

- Gradually move the spot lights from one side of the mode to another.
- When they get to the one side, gradually dim them (mimicking evening).
- After this, make the overall lighting dimmer but *don't* reduce the torch lights. This will enhance the light from the torches.
- After a little while, place the spot lights on the *other* side other model, gradually increase the overall lighting, and start moving the spot lights again.

8. (150 mark **bonus**): Make the wall thin, not using a “bunch of columns”.

- Only use a four (4) sided column for “X” values that are separated from everything.
- Every other “X” – that’s a wall – use an appropriate two sided wall model (i.e. the wall models facing back to back). Make sure that the corners line up OK.
- If there’s a “T” or “X” intersection, you can intersect the models if you’d like.
- In other words, the walls in the map should only be “2 wall tiles”

9. (100 mark **bonus**): Make the water in the pool move.

- If you’ve used a lily pad texture, you’ll have to add a plain “water” texture as well.
- Add at least two water textures (these can be the same texture, but go into paint and rotate and scale the 2<sup>nd</sup> texture) and gradually adjust the offset to the texture coordinates over time, but by a different amount. i.e. each texture is having its UV coordinates updated, but differently.
- To get the final colour, sample and combine both these textures.  
Be careful not to over-saturate (i.e. make it “too bright white”) as adding the two will do this.

10. (100 mark **bonus**): It’s a drought.

- Mimic the dungeon going through a drought. You do this by:
  - Gradually lowering the water level until its under the floor of the pool.  
(i.e all the water eventually dries out)
  - If you’ve used the lily pad texture, leave it at the bottom of the pool – i.e. don’t make the texture go under the floor, so we can still see the lily pads.  
Use discard transparency to remove the “water” parts of the lily pad texture, though.
  - After the water is gone (after a few seconds), slowly change the textures on all the plants to a brown (or whatever colour you think dead plants are).  
This should take a few seconds as well.

**That’s it.**