# **INFO-3111 “C++ Graphics” Mid-term Exam – Summer 2020**

Monday, July 27th, 2020

Instructor: Michael Feeney

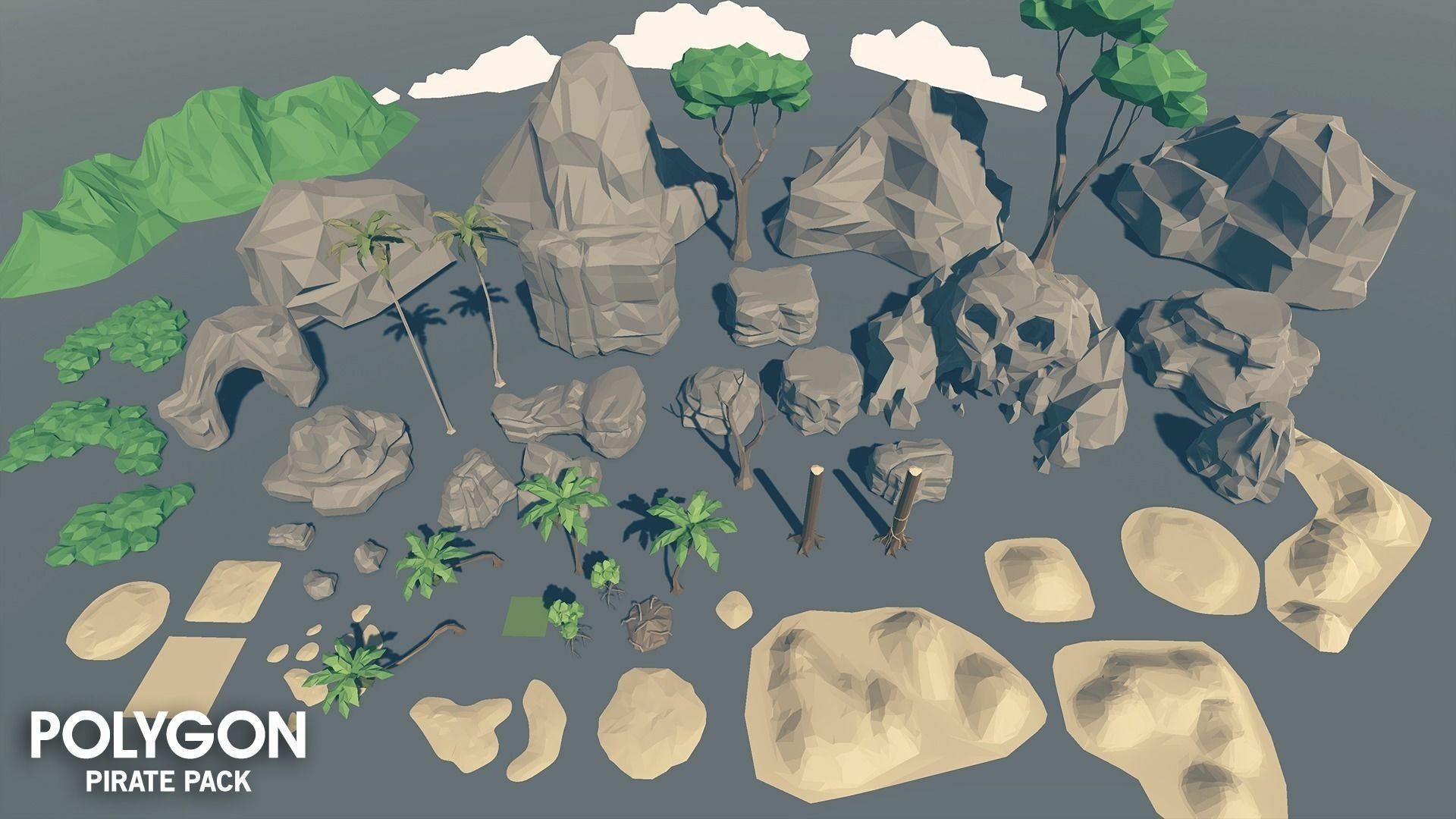
## The exam format:

* 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>
* The questions are ***NOT*** of equal weight. There are six (6) pages with six (6) questions
* 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.
* Place any written 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 **8:00 PM** on **Monday, July 27th** to submit all your files to 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 6:00 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)

* You can reach me through e-mail ([mfeeney@fanshawec.ca](mailto:mfeeney@fanshawec.ca)).
* There is also a **Pirate\_Rocks\_and\_Trees.7z** file you will need. It’s available on FOL with the mid-term.

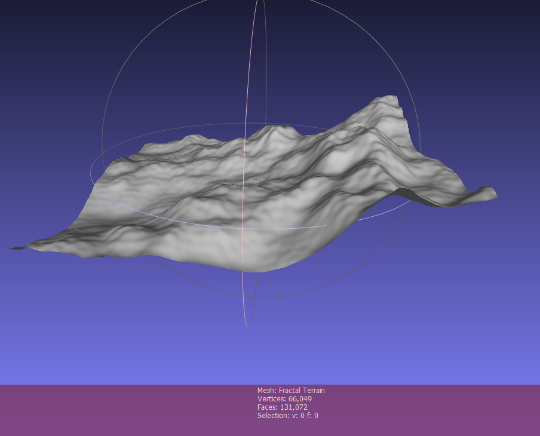
## Questions:



You are going to create a more detailed pirate island, using a combination of the meshlab and some of the models in the “Polygon Pirate Pack” (seen above). Here’s the link to the entire collection: <https://www.cgtrader.com/3d-models/exterior/historic/polygon-pirate-pack>

1. (20 marks) Using MeshLab, generate a SINGLE island, in the following way:

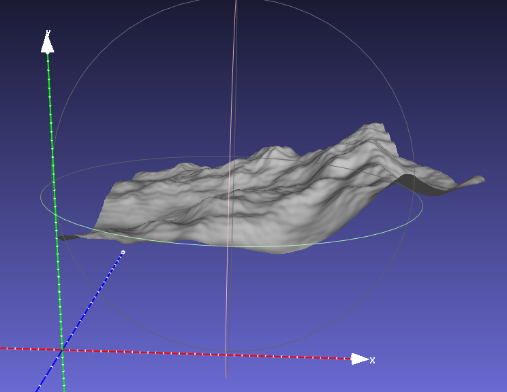
* Open MeshLab (without opening a model). This will open it with an empty “project”
* Choose “Filters”, then “Create New Mesh Layer”, then “Fractal Terrain”
* In the “Fractal Terrain” dialog box, choose “**Hybrid multifractal terrain**” (“Algorithm” dropbox.)
* Change the “Max Height” to **0.5**.
* Pick a “Seed” value (the default is 2.0)
* ***… using the following method:***
  + Get the UTF-8 “decimal code point” for each letter of your *full* name. You can get these numbers here: <http://www.ltg.ed.ac.uk/~richard/utf-8.cgi?input=M&mode=char>.
  + Add all these numbers up. Take the first four (4) numbers of the final result as your seed value. For example: **Michael Feeney** gives:   
    77+105+99+104+97+101+108+ (“Michael”)  
    70+101+101+110+101+121 (“Feeney”)  
    = 1295 🡪 *so my seed would be “1295”*



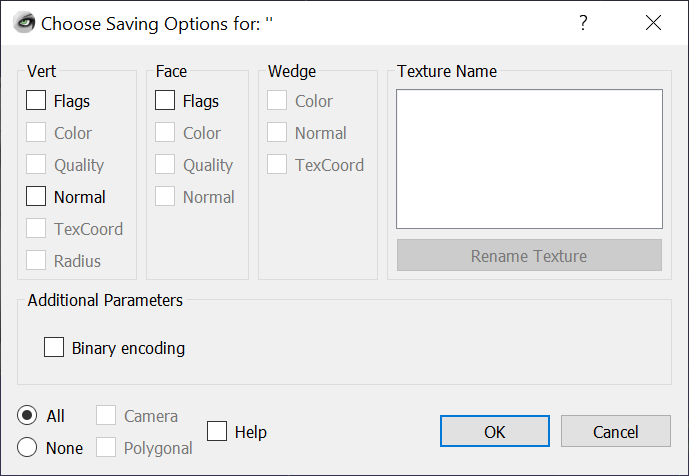
* + (If your *full* name doesn’t generate four (4) digits, then assume the 1st digit is a zero (0).)

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

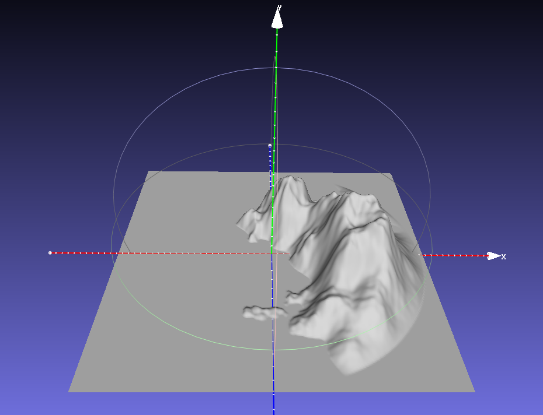
The Island Converter assumes “up” is “z”, so we need to adjust this. Turn on the “axis” drawing by choosing “Render”, “Show Axis” to make this clear (if you want).



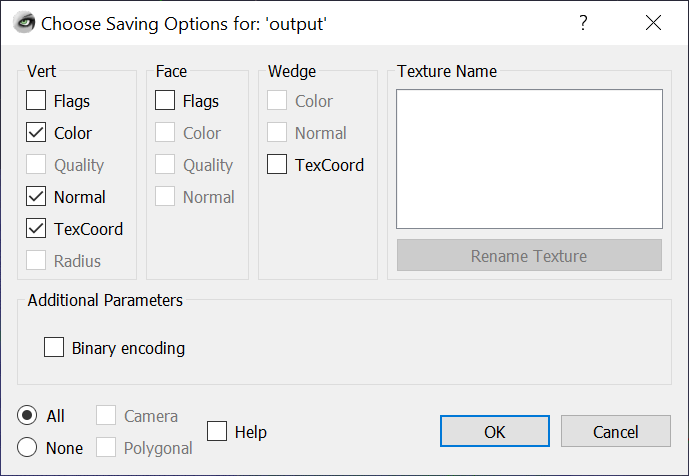
* Choose “Filters”, “Normals, Curvature, and Orientation”, then “Transform: Rotate”.
* Type in “-90” in the “Rotation Angle”, leaving the “Rotation on:” set to “X axis”, and click “Apply”, which will get you this 🡪   
    
  (Note: Do **NOT** change the origin to the centre; you’ll do that later.)



* Save this model with **JUST xyz** and NOT in binary form (“File”, “Export Mesh As…”, uncheck the “Binary encoding”, and choose OK. In other words, **UN**check everything (this is the format the terrain to island converter needs).



* Download and compile the “Terrain\_to\_Island\_Converter” project and convert the mesh you made into an island. This takes the model file name as a command line input and generate an “island” mesh (like the one on the right 🡪) called “output.ply”.   
    
  Note: you can also drag your input ply file onto the executable (exe) file.
* Open this model into mesh lab, **generate normals** and **scale it by 2.0x**, and add **texture (UV) coordinates**:
  + Choose “Filters”, “Normals, Curvature, and Orientation”, then “Transform: Scale, Normalize ”; type in 2 in the “X Axis” text box;   
    Click “Apply”
  + Choose “Filters”, “Normals, Curvature, and Orientation”, then “Re-Compute Vertex Normals”
  + Choose “Filters,”, “Textures”, then “Parameterization: Flat Plane”  
    Choose “XZ” as the “Projection Plane”  
    Click “Apply”
  + Next, choose “Filters”, “Texutre”, “Convert PerWedge UV into PerVertex UV”  
    Click “Apply”



* Save your model, “File”, “Export Mesh As…”
  + Make sure you **UN**check “Binary”, and check “Color”, “Normal”, and “TexCoord” (under “Vert”).   
    *Everything else should be UNCHECKED*.
* You will use this model as the island your pirates live on.   
  The model should be:
  + 512 units wide in the XY axis, from -256.0 to +256
  + The “water” should be at 0.0 along the y axis.
  + The maximum height should be something like 150 (but will depend on the seed value)

1. (50 marks) Decorate the initial pirate island:

* Place the “skull” model at the highest point on the island.
* Choose 3 of the larger rock models, and place them on the next 3 highest points.   
    
  Note: You don’t have to programmatically do this; just look at the island and pick it with your amazing human eyes and brain!
* Place 3 of the “cave” (called “SM\_Env\_Rock\_Arch\_01”) models, somewhere at the base of the island:
  + 2 of them should be on the “water” and at the side of the island. In other words, you should be able to enter the cave from the water, and the cave should look like it’s leading into the side of the mountain.
  + 1 of them should be in the side of the mountain, but should be on one of the beaches. In other words, you would get into the “cave” by walking.
  + Place a small rock model on the beach, somewhere in front of the cave (see the last part of question 3 to get more details of the placement of this rock).
* There are several tree models (and 3 tree trunk models). Place 8 trees on the island, using at least 5 different kind of tree. Place these in “reasonable” locations, and they should be somewhat scattered around the island (i.e. they shouldn’t all be in the same place, or very near each other).
* Place a single point light in the scene, to simulate “daylight”. Note: you can NOT use a directional light (if you don’t know what this is, then don’t worry about it, but it was asked in one of the help sessions).
* Place the camera somewhere where you can see the entire island.

1. (50 marks) It’s night time!

* Change the light from question 2, so that it’s very dim, simulating “night time”.
* Place 2 small orange point lights in the “eyes” of the “skull” cave. They should only illuminate the inside of the eye sockets
* Place 1 small orange point light in one of the “cave” (arch) models. It should look like there is a fire glowing inside the cave.
* Assign the “N” key to set the lights to this “mode” (“nighttime”).  
  Assign the “D” key to set the lights back to “daytime” – as they are in question 2.

1. (30 marks) Place the camera at good locations to see your masterpiece. Use keys 1-3

* 1 – places the camera in a location to look at the skull, from the front
* 2 – places the camera behind the small rock, in front of the cave on the beach, looking towards the cave
* 3 – place the camera above the island, looking down

1. (**30** **BONUS** marks) Change the fragment shader to the “fragShowVertexColours.glsl” shader.

* This shader is identical to the in-class one, with the exception of a “uniform bool bUseVertexColours;” added. Setting this to “true” will use the vertex colours from the file, rather than the “one colour per object” colour. This way you can see the colours of the island.

1. (**30 BONUS marks**) Make the lights in the skull “flicker”:  
   * Choose an attenuation that is the “dimmest” that the lights will be.
   * Choose an attenuation that is the “brightest” that the lights will be.
   * In the main game loop, pick two random number between these values, and set the illumination of the two lights.
   * At each frame, the illumination value will change, making it look like the lights are “flickering”

**That’s it!**