|  |  |
| --- | --- |
| Program: | CPA3 |
| Course: | INFO3111 – C++ Graphics – Summer 2023 |
| Professors: | Michael Feeney |
| Project # 1: | Basic 3D scene of loaded, transformed models |
| Weight: | 15% of your final mark |
| Due Date: | Thursday, June 1st, 2023, before the start of class (@12:00 PM)  (Note: This is due after the mid-term) |

Description and Purpose

You are to create a 64 bit Win32 (Windows 7/8/10) + OpenGL 4.x application that displays a “sensible” static (or dynamic) 3D scene using external models that have been translated in the scene using transformations, and either coloured or lit using a basic GLSL shader. You will have to be able to interactively manipulate (i.e. “move”) the camera to view the scene.



**You are going to make an underwater scene involving a shipwreck, rocks, fish, etc.**

The plyFiles.7z file has a large number of models to chose from, divided into sections:

* Water, seafloor, etc.: A “seafloor” model, as well as a “ocean wave” model
* Docks, boats, etc.: Contains a couple “ship wreck” models and some dock models
* Fish: Well... has fish, jellyfish, sharks, and even a whale!
* Plants: Anything “plant like” (we’ll pretend they are all underwater plants)
* Debris/”stuff” to put on the sea floor: Things like barrels, skulls, bottles, etc.

Unless otherwise stated, the models are “solid fill” (i.e. not wireframe) with the z/depth buffer should be on.

Any “debug” items (like the wireframe spheres used to place the lights) should be off by default (when the application is started).

Some of the questions/requirements build on each other (like question 1 & 2). If you choose to combine them, please do the following:

* Bind the number keys to switch between camera locations (if needed).
* CLEARLY indicate which questions you are answering.   
  For example: “Press number key 2 to see the answer to questions 1 and 2”
* You can also make a “readme” file that is SEPEARATE from the solution.   
  i.e. I will \*NOT\* try and hunt down comments buried in the source code.   
  (If your “readme” directs me to certain parts of the code, that is likely OK)
* I will *NOT* change your code. I should run AS IS.   
  Do NOT indicate something like “Comment out lines X to Y to see question 3” or something like that.   
  I will NOT read stuff like that.

**If it doesn’t run, I will assume it can’t run, and you will get a mark of zero.**

**It MUST build and run in RELEASE mode (64 bits).**

(I *might* do something minor like try to unzip the libraries files that github stripped out or something along those lines, but I should be able to download it, build it, and run it, without incident.)

Warning level 3 is fine. In fact, I’ll almost certainly completely ignore any “warnings”.

**No “boost” library or “auto” keyword. Use of these will give you a mark of zero, no exceptions.**

You can only use code that was provided for you during course, or that *you* wrote during this course.

Use of any other code is not allowed. This includes, but is not limited to:

* chatGPT code.
* Code you “borrowed” from some other site.
* Some game engine.
* Etc.

Here’s the thing: I know the code *\*I\** wrote and if I’m suspicious that you had didn’t actually write (i.e. understand) the code you submitted, I’ll simply ask you about it face-to-face. Like why you did this or that, how you might do it differently, ask how you might have changed stuff. If you just “got this code somewhere”, it’ll be pretty obvious, based on your responses.

Details:

1. (10 marks) Using the “Seafloor 2” model and one of the “Ocean” or “Ocean waves” files, create the basics of where your scene will be:  
   * The sea floor should be light brown or brown-grey coloured, like sand.
   * The ocean represents the waves and should be blue and wireframe.
   * Place the camera between the two models (like you are a scuba diver or swimmer who is “underwater).
   * The models should be scaled so that fills most of the screen (sort of like the image above).  
     This means that they should be large and or large relative to where the camera is (or both).
2. (20 marks) Place a shipwreck in the centre of the sea floor.   
   * Note that there is a “shipwreck” model (two parts of them) and an entire ship.   
     Choose whichever one you’d like, but it has to be all of the ship (so if you use the “shipwreck”, you’ll have to place both parts).
   * Place these in the centre of the scene, somewhat “buried” into the bottom of the sea floor.   
     It should also be at a reasonable scale, taking up a “reasonable” proportion of the scene.   
     (like it shouldn’t be tiny, or be even close to the same size as the sea floor model, but it should be the main “focus” of the scene – sort of like the image above).
   * If you choose to move the camera from question 1, then make sure you can see the entire ship.   
       
     Note that the image on the previous page is just a *very* general idea (I was looking for a picture with a ship, some fish, etc.)
   * The ship should be brown, i.e. “wood” coloured.
3. (60 marks) Add a “lot” of fish.   
   * Using whatever “fish” models you’d like, add:  
     + At least one whale.
     + At least one shark.
     + At least 5 other “individual” fish (i.e. swimming around alone or in pairs).
     + At least 2 “schools” of fish (“fish cluster” model).
     + The amazing “Santa Octopus” sitting on the sea floor.
     + At least 3 (three) jellyfish.
   * These should be set to a relative scale and orientation.   
       
     For instance, whales are large (like as large as the ship), a shark is much smaller, but pretty big, while jellyfish and a school of tiny fish should be, well, pretty tiny.
   * Choose whatever colours you’d like, but they should be somewhat “appropriate”. Like the shark should be “grey”, the whale grey or blue, but the fish are tropical so could be bright colours.   
     (Google “tropical fish” to get a sense of what’s typical for fish colours)
4. (40 marks) Place some rocks or coral on the seafloor:

* In the “crap to put on the sea floor”, there’s three (3) rock models (and other stuff – for question 5).
* Place 20 rocks around the sea floor using these three (3) models.   
  + Like the ship, they should be placed like they are slightly “buried” into the seafloor (like the seafloor is sand).
  + Even though there’s only 3 models, you can change the scale and orientation to give quite a bit of variation. In other words, it shouldn’t look like you’ve just used the same model over and over again – at a glance, it should give the impression that there’s more than just three types of models.
  + Colour the rocks as grey or brown (you know, “rock colours”), but give them *slight* variations of colours – like slight variations of grey/brown/whatever.
* Place one “arch” rock (SM\_Env\_Rock\_Arch\_01). Make this a unique “rock” colour” (i.e. slightly different from the other rocks, but still grey or brown or whatever). This should be larger than the other rocks.
* Place one “skull” rock (SM\_Env\_Rock\_Skull\_01). Colour it like the above (unique, but still “rock coloured”).  
  This should also be larger than the other rocks.

1. (60 marks) Time for some gardening!

* There are 12 “plant” models in the “plants” folder.   
    
  Place 20 “plant” models of at least six (6) types.
* Colours these using reasonable “plant” colours you’d see underwater, so mainly variations of green, but perhaps some browns, yellows, etc. (Google “underwater plants” to give some ideas).
* Like the rocks, the size should be appropriate, they should be placed so that they look like they are “growing” out of the sea floor, and they should each be unique – like each plant should be a slight variation of the other plant colours (like slightly different green colours, etc.)
* Place a few of the plants should be on the rocks, too.

1. (60 marks) Place enough lighting to see what’s going on.

* You may use any number or type of lights you’d like.
* I want the scene to be visible, but not “washed out” (i.e. the light is far too bright) or too dark (it’s daytime).
* It should somewhat mimic sunlight coming in from above the water surface.
* Assume the water is clear and not too deep, so the sunlight is reaching all they way to the seafloor.

1. (60 marks) Place a bunch of debris on the seafloor, likely things that were inside the ship. There are many models in the “crap to put on the sea floor”: cups, cannons, a cup - even an anchor.

* Place 10 (ten) of these items in reasonable places on the seafloor and/or on the rocks and/or on the ship. Spread these around, so not all on the ship or all in one location, but as if they were scattered when she ship sank and fish and waves have moved the stuff around over time.
* Note that some of the models are a collection of smaller items (like the “SM\_Prop\_Stack” models).   
    
  These are considered as \*1\* model though – i.e. a stack of 3 barrels *isn’t* part of the 10 items.
* Colour these in whatever appropriate colours you’d like.

1. (60 marks) Using the keyboard, set up variations of camera locations:

* Pressing the “1” key will move the camera into the location from question 1. That is: underwater, with the “waves” model as wireframe, “looking at” the ship wreck.
* Pressing the “2” key will move the camera above the scene, looking “down”. Imagine this would be from a drone or helicopter above the scene rather than someone underwater.   
  + This should b well above the “waves” model, which you can see through since it’s wireframe.
  + The camera should be high enough that you can see the entire scene (more or less).
* Pressing the “3” key should place the camera so that it’s underwater and looking “through” the arch way of the “arch” rock; imagine that you were a scuba diver, and you were behind the arch, but looking through the space at the scene. It’s reasonable that you can’t see *all* the scene, but you the camera should be places so that you can see at least a number of interesting things.
* Pressing the “4” key should place the camera somewhere just above the deck of the shipwreck (imagine you were standing on it, or swimming right above), looking at the shark. This could be in any direction – it depends on where you placed the shark, right? (If the shark isn’t visible, like it’s behind something, please move the shark)
* Pressing the “5” key should be like “4” above but looking at the whale. Like viewing the shark, if the whale is hiding behind something, please move the whale somewhere else so you can see it.   
    
  (For these last two, it’s possible that the shark or whale are quite close to the shipwreck – that’s totally fine, as long as the “camera angle” shows **both** the ship and the shark/whale).

**BONUS (5% overall each):**

* “Move” the shark by updating its position each frame. You do this by adding some very small number to the position of the shark model, so its location updates, and it looks like it’s “moving”
* Update question 8, with the “4” key, where it’s looking at the shark, so that it *keeps following* the shark.

You will submit:

* **Your entire solution** (PLEASE remove the “extra” files from it, making it smaller), and compress it.
* **A video demonstrating your application.** This can be using OBS or zoom or FRAPs or even your cell phone camera but has to *clearly* show:  
  + You launching the application from visual studio
  + CLEARLY showing some identifying information, like a comment in the code with your student name and number
  + What you are doing (buttons being pressed, etc.) as it is happening.

Additional requirements:

* While you may freely “borrow” mine (or anyone other) code ***but*** your code should be “sufficiently” different from mine in terms of the output on screen. See the "plagiarism" test, later in this document, for more details.
* Further, you *cannot* simply use an existing game engine (or part of a game engine), even if it's a "from source" engine (i.e. you have the entire source) to complete this assignment; it should be either completely new of significantly modified. This includes, but is *not* limited to: Unity, Unreal, Cry, Anarchy, XNA, Cocos, Ogre, the framework from the OpenGL text, etc.   
    
  Related to plagiarism/cheating and not doing much work:   
  + If you simply submitted the in class code, then you invested zero time, so you did no work, so you get a mark of zero (in that case, it is a clear case plagiarism/cheating, and an academic offence would also be submitted).
  + If you took the in class code and made some trivial changes - like replaced the teapot model with the rabbit model, slightly repositioned them, and maybe changed their colour - you might not have actually "cheated", but you did essentially no work: "How long would it take me - your instructor - to make those changes?" If it's something that would take 10 minutes, you won't get many parks for that
  + It has to be something that a random "typical" person would say "yes, that's significantly different" in order to "be different".
  + If you code does not even compile, I will not mark it. Since it can't run, you would get a mark of zero.
  + If you code does not build (i.e. linker error) and run (i.e. no crazy run-time crash that is unexpected), I may investigate this further, but only if there is some simple problem and/or *very* slight and/or *very* obvious (and easy to fix) configuration error or last minute typo.
  + **\*\***You can **\*not\*** submit something just clearly based on the mid-term (*including the mid-term itself*).

Project Corrections

If any corrections or changes are necessary they will be posted to the course web site and you will be notified of any changes in class. It is your responsibility to check the site periodically for changes to the project. Additional resources relating to the project may also be posted.

80/8-year old “squinty eye” plagiarism test:

I have very little tolerance for plagiarism, but some students might be unclear about what it is.

Basically, it’s submitting somebody else’s work as your own.

There is sometimes some confusion over this because you could argue nothing is actually “unique” (see: <http://everythingisaremix.info/> for a fascinating overview of this).

The whole point of assignments/tests/projects in this course (or any course, really) is to try to see if you are actually able to ***do*** the coding that’s asked of you. In other words: How competent are you? Handing me someone else’s code and/or making a trivial change isn’t good enough.

Also, it’s illegal:

* <http://www.plagiarism.org/ask-the-experts/faq/>
* <http://definitions.uslegal.com/p/plagiarism/>
* <http://en.wikipedia.org/wiki/Plagiarism>
* <https://www.legalzoom.com/articles/plagiarism-what-is-it-exactly>

In other words, I’m not going to be drawn into a giant debate over how “different” your code is from mine or anyone else’s, if any sensible person (including me) would conclude that the code/application is pretty much the same thing, then it is. It is up to my discretion to decide this.

* While you may freely “borrow” mine (or anyone other) code ***but*** your code should be “sufficiently” different from mine (you might want to replace the word “sufficiently” with “significantly”).
* In other words, you *cannot* simply use an existing game engine (or part of a game engine) to complete this assignment; it should be either completely new of **significantly** modified.
* How will I determine this?
  + If I showed your application and/or your source code to either a pragmatic 80-year-old mother, or a typical 8-year-old, or even some random person walking down the hallway (i.e. a non-expert), and they looked at it, tilted their heads, squinted their eyes, and said “you know, they look the same,” then they ***are*** the same.
  + Another test would: How much time it would take for a "competent programmer" (me, for example) to make the changes you are submitting? The point here is that I don’t “care” if you tell me “But it took me *weeks* to make the changes!” Fine, but if I can make those same changes in 10 minutes, then not a lot of work has been done (certainly **not** sufficient work for someone who is trying to convince me they are proficient).