Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Toward a Futuristic Emerald Isle

Chris Casey

Binh-Son Hua
December 29, 2024

Adapted from a template created by Prof. Michael Brady

Submitted in partial fulfilment of the requirements for the degree of
Computer Engineering

Declaration

I hereby declare that this CG Project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

Signed: Chris Casey

Date: 27/12/24

Abstract

In this project OpenGL 3.3 and C++ were used to simulate an infinite scene with a theme: "Towards and Futuristic Emerald Isle." Features from all four labs were required to be implemented. They included: geometry rendering, texture mapping, lighting and shadows and finally animation.

Some other considerations for the scene were that the frame rate must not fall below 15 FPS, allow user control of camera and implement an advanced feature.

The deliverables for this project were:

- Source code and Git Repository

- Final report in LaTeX (here)

- Progress report showing screenshots (included here)

- MP4 video showing the rendering of the scene.

1 | Introduction

1.1 Overview of Scene

1.1.1 Background

"In keeping with the theme of a futuristic emerald isle, I created a dystopian world where humanity has been forced underground to escape rising temperatures due to global warming. Individuals are reminded of the impending doom as they look up to see a black hole, actively swallowing planets! In the scene we can see how humanity has been forced to innovate, taking over stalactite structures and using them as buildings. Materials used to build these was recently discovered and is ultra high-tec!"

1.1.2 Implementations

This paragraph denotes what was achieved in the final source code. Several complex features were required to be implemented as detailed above. Below is a comprehensive list of what I achieved.]

Several complex features were required to be implemented as detailed above. Below is a comprehensive list of what I achieved.

A point light source and lighting were attempted but incorrectly implemented and had to be scrapped. Animation was also attempted but couldn't be included in the scene. The MP4 video and screenshots show these implementations/ attempts.

The frame rate is steadily above 15 FPS as shown in MP4.] I successfully rendered and textured buildings, roads and other objects for my scene. A custom SkyBox was put together and implemented correctly using online free use images and Adobe Editor. Camera control was included to allow user interaction with the scene.

A point light source and lighting were attempted but incorrectly implemented and had to be scrapped. Animation was also attempted but couldn't be included in the scene. The MP4 video and screenshots show these implementations/ attempts.

The frame rate is steadily above 15 FPS as shown in MP4.

1.1.3 Challenges

This paragraph denotes what I struggled to include in the final source code. I began working on an infinite plane but struggled to complete a chunking method. Upon defining a plane struct as well as reshaping my shaders to clamp the eye in place, I saw strange results. The camera was being reset, but any input lead to 'shaking'. This was omitted as a result.

I found lighting and shadows particularly difficult to implement. See below for my best attempt. I was able to define the logic for lighting and shadow creation in my shaders, however I could not integrate this with my texture mapping. Debugging attempts included binding the textures at earlier points in the runtime, ensuring correct UV Buffer, texCoords passed between shaders, etc. Unfortunately I could not figure this out and lighting had to be omitted. The code is still available on the Git repository.

Animation also had to be omitted for similar reasons. I was seeing positive results (skeleton runs as expected, no errors), but when included in the render loop, the rest of the scene disappeared. I believe this is due to some reassigning of OGL parameters.

]Unfortunately not all of the requirements for the project were met. Below is a

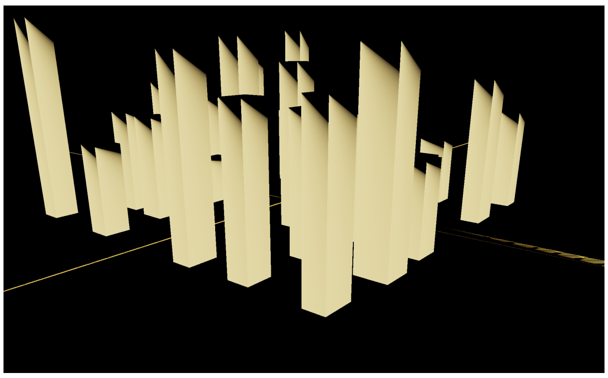


Figure 1.1: Attempting lighting and shadows

comprehensive list of what I struggled to do.

I began working on an infinite plane but struggled to complete a chunking method. Upon defining a plane struct as well as reshaping my shaders to clamp the eye in place, I saw strange results. The camera was being reset, but any input lead to 'shaking'. This was omitted as a result.

I found lighting and shadows particularly difficult to implement. See below for my best attempt. I was able to define the logic for lighting and shadow creation in my shaders, however I could not integrate this with my texture mapping. Debugging attempts included binding the textures at earlier points in the runtime, ensuring correct UV Buffer, texCoords passed between shaders, etc. Unfortunately I could not figure this out and lighting had to be omitted. The code is still available on the Git repository.

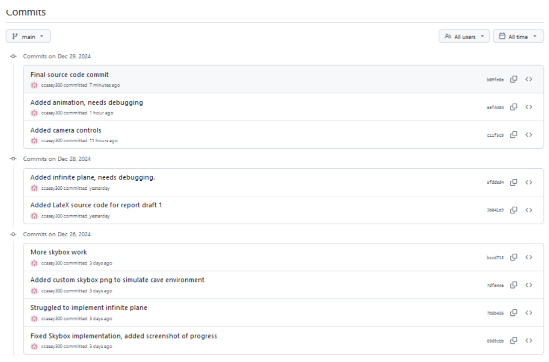
Animation also had to be omitted for similar reasons. I was seeing positive results (skeleton runs as expected, no errors), but when included in the render loop, the rest of the scene disappeared. I believe this is due to some reassigning of OGL parameters.

]

2 Progress Report

GitHub was used for version control and mapping progress. Screenshots were taken along the way which show the scene through various stages of development.

A screenshot of the git history is shown
[https://github.com/ccasey300/CG$_{p}roject.git$] :



Commits

Commits on Dec 29, 2024

Final source code commit
ccasey300 committed 7 minutes ago
b09fe0a

Added animation, needs debugging
ccasey300 committed 1 hour ago
aaf4484

Added camera controls
ccasey300 committed 11 hours ago
c11f5c9

Commits on Dec 28, 2024

Added infinite plane, needs debugging.
ccasey300 committed yesterday
5fdd8d4

Added LateX source code for report draft 1
ccasey300 committed yesterday
3b041e9

Commits on Dec 26, 2024

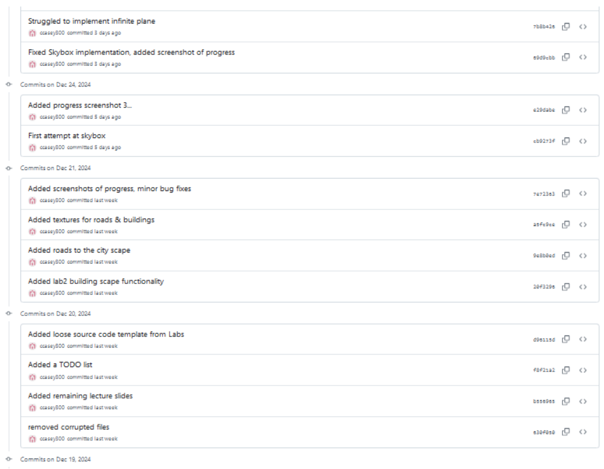
More skybox work
ccasey300 committed 3 days ago
bccd715

Added custom skybox png to simulate cave environment
ccasey300 committed 3 days ago
7dfe44a

Struggled to implement infinite plane
ccasey300 committed 3 days ago
7b8b426

Fixed Skybox implementation, added screenshot of progress
ccasey300 committed 3 days ago
69d9cbb

Figure 2.1: This screenshot shows the Git repository history (1/2).



Struggled to implement infinite plane
ccasey300 committed 3 days ago
7b8b42e

Fixed Skybox implementation, added screenshot of progress
ccasey300 committed 3 days ago
e9d9cbb

Commits on Dec 24, 2024

Added progress screenshot 3...
ccasey300 committed 5 days ago
e29dabe

First attempt at skybox
ccasey300 committed 5 days ago
cb9273f

Commits on Dec 21, 2024

Added screenshots of progress, minor bug fixes
ccasey300 committed last week
7e72363

Added textures for roads & buildings
ccasey300 committed last week
a5fc9ce

Added roads to the city scape
ccasey300 committed last week
9e8b0ed

Added lab2 building scape functionality
ccasey300 committed last week
20f3296

Commits on Dec 20, 2024

Added loose source code template from Labs
ccasey300 committed last week
d96115d

Added a TODO list
ccasey300 committed last week
f8f21a2

Added remaining lecture slides
ccasey300 committed last week
b556965

removed corrupted files
ccasey300 committed last week
e30f0s0

Commits on Dec 19, 2024

Figure 2.2: This screenshot shows the Git repository history (2/2).

Progress: Description of Screenshots

Screenshot 1: Geometry Rendering

Screenshot 2: Texturing and SkyBox

Screenshot 3: Custom SkyBox

Screenshot 4: Lighting and Shadows attempt

Screenshot 5: Animation attempt

Screenshot 6: Final Render

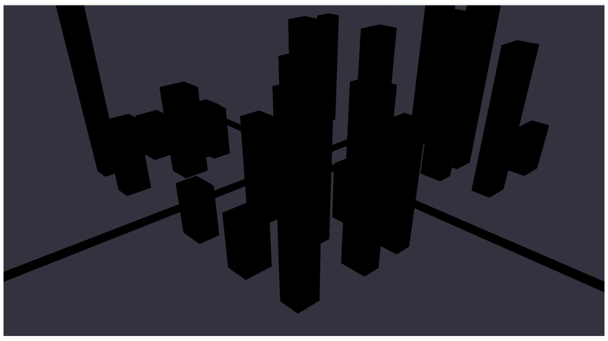


Figure 2.3: This screenshot shows successful rendering of buildings and roads in the scene.

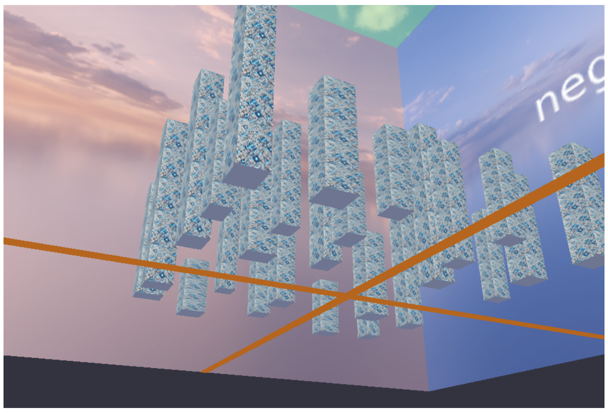


Figure 2.4: This screenshot shows texturing of objects (buildings and roads), and implementation of skybox debug from lab 2.

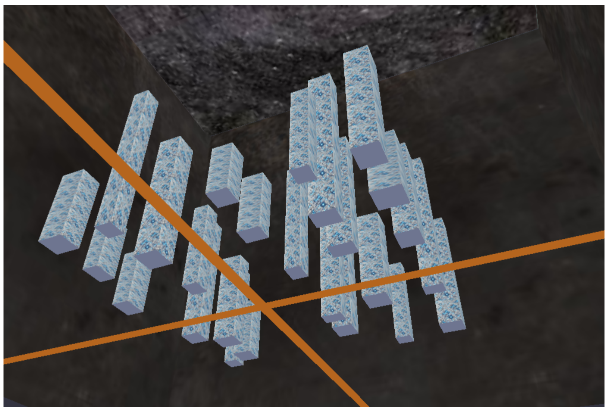


Figure 2.5: This screenshot shows a custom "cave themed" SkyBox. This was created using online resources like Adobe Image Editor and free-use google images.

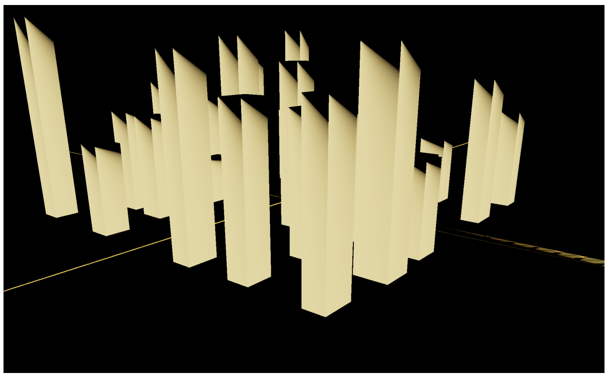


Figure 2.6: My attempt at lighting and shadows, progress can be seen but ultimately could not integrate with textures.

Figure 2.7: My attempt at including animation, could not resolve OGL parameter logic in time

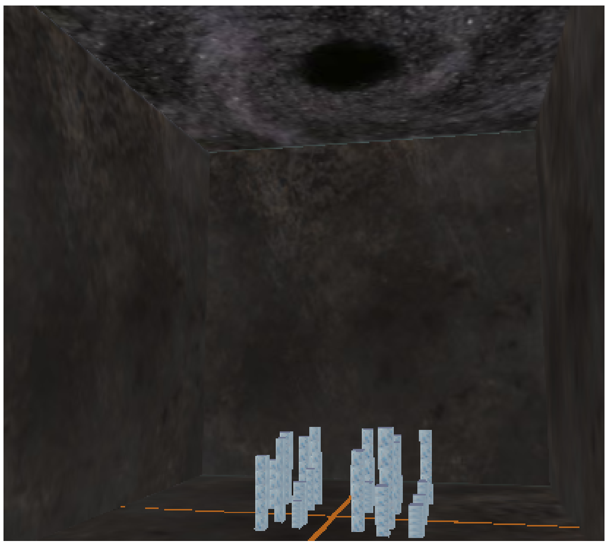


Figure 2.8: Final render of scene

3 | Discussion: Quality + Robustness

I primarily worked using a combination of different lab templates. I primarily used lab 2 and lab 4. However, this approach came with its limitations. Primarily, a lack of object-oriented code as well as being more application specific than I wanted. This lead to lengthy attempts and difficulty debugging.

As a result, though the code is (maybe) easily understood upon visual inspection, additional elements required a significant amount of work in order to integrate with the existing scene. In this way, I was too dependent on the templates given and should have deviated my approach earlier.

There is some redundant code, several similar structs were defined in order to resolve some texturing issues I faced. If I had more time in the future I would take the existing functionality and try to leverage OOP.

The code includes error-handling much in the same way we did in class. File loading, shader implementation and null pointers are managed appropriately. All cleanup is done after rendering as required. Performance was consistently good, however was never properly challenged. The option to include more animation, more lighting, etc. is there. The scene consistently runs well over 15 FPS on my laptop.

4 | Conclusions and Future Work

If I was to revisit this project, there are quite a few improvements I would make. The biggest being previously mentioned OOP.

I would focus on technical implementations at an early and revise my attempts to debug. Inclusion of modular programming would likely make the project's completion far more effective. Rather than building all of the objects in the scene, implementing a skybox, etc., I would focus on one object and ensure I can get lighting and shadows to work. .cpp became too cluttered and I was pigeon-holed.

I struggled to implement an infinite scene also. I tried using a chunking method, clamping the camera and actively rendering the environment based on user input but wasn't successful.

Lighting and animation I found particularly difficult to integrate into my existing template. I believe my shader logic for shadows was at least partially incorrect, (see screenshot). I was disappointed to not be able to include animation succinctly in my scene, though the MP4 video shows my attempt. We can see the skeleton running as required in the previous lab 4. Debugging attempts included focusing on OpenGL parameters redefinitions and resetting these. Debugging attempts were unsuccessful and could not get A) the city scape with skybox and textures and B) skeleton running on screen.

Overall, I struggled most with combining different techniques in my code. This was a symptom of poor decisions early on. We can see that some valid attempts were made (lighting, animation), but were not ready in time for the deadline.

Acknowledgements

AI models used for debugging and creating interesting objects:

GPT 3.0

GitHub CoPilot (GPT 3.0 also I believe)

Resources used:

OpenGL Tutorials http://www.opengl-tutorial.org/

Infinite plane attempt https://www.youtube.com/@OGLDEV

Lecture notes, labs, etc. Binh-Son Hua

Contents