# Take a Virtual Hike

Design Document

Team 19
Client: Mat Wymore
Advisor: Mat Wymore
Team Members: Trevor Nemes (Team Leader), Tyler Hassfield, Opeyemi Abass, Aashutosh Mallik, Akhilesh Ratnakumar, Zian Li
Email: sdmay21-19@iastate.edu
Website: https://sdmay21-19.sd.ece.iastate.edu/

Revised: 9/29/2020

# Executive Summary

## Development Standards & Practices Used

- Development
    - Each team member will focus on researching a specific element of the project.
    - Team will utilize new information to generate a plan in regards to building and integrating a VR Environment modularly.
    - The development process will follow a Waterfall life cycle.
- Communication
    - Weekly Zoom meetings to cover progress, have open discussion, and create a plan for the following week.
    - Private chat for quick, informal communication.
    - Documentation of meetings
- Coding Style
    - Consistent naming conventions, descriptive commenting, and regular testing.
    - Have modular design and implement a design pattern.
- Digital design
    - Use of Perlin Noise Algorithm for generating visuals.
    - Use OpenGL graphic API
    - Implement procedural generation to have an ever expanding world

## Summary of Requirements

- Explorable 3D world (first person perspective)
- Nature-themed world
- Full-scale environment, including life sized mountains and trees (1-to-1 scale)
- Use procedural generation when possible
- Basic collision detection
- Fly-through mode and walk-through mode
- No noticeable delay/choppiness while exploring the world (30 fps)
- Core code should not be platform specific
- Core code should be modular and extensible
- Include audio and/or a soundscape

## Applicable Courses from Iowa State University Curriculum

- CprE: 185, 288, 308, 388
- ComS: 227, 228, 309, 311, 327, 329, 339, 352, 363

## New Skills/Knowledge acquired that was not taught in courses

- OpenGL
- Procedural Generation Algorithms
- Creating and working with soundscapes

# Table of Contents

## List of figures/tables/symbols/definitions (This should be the similar to the project plan)

# 1 Introduction

## 1.1 Acknowledgement

We would like to thank and acknowledge Professor Mathew Wymore for his time, effort, and support during the development of this project.

## 1.2 Problem and Project Statement

Problem Statement:

The COVID-19 pandemic is mentally and emotionally stressful, and has limited opportunities for activities such as vacationing and connecting with nature. This project aims to create an explorable, full-scale three-dimensional virtual nature environment for the purpose of relaxation and stress relief. Planned unique features include a 1:1 scale with reality (how big should that virtual mountain REALLY be?) and elements of procedural generation. Due to these features, the direct use of a low-level graphics APIs, OpenGL, is expected, and various memory and CPU optimizations may need to be employed to achieve satisfactory performance. Programming languages may include C++ or others.

Project Statement:

To get this project done, we will be using the OpenGL graphics API to design the world. We will design all aspects of the world on a 1-to-1 scale with real life, such as trees, mountains, clouds, etc. We will create an algorithm using procedural generation which will help generate all aspects of the world randomly so that everyone's world will be unique every time they play. We plan on implementing a fly-through mode and a walk-through mode so that the users can choose either one to explore the world. We expect our application to run at 30 fps with no visible choppiness or lag. Lastly, if we have ample time, we will implement a nature themed soundscape to the application as well.

## 1.3 Operational Environment.

The virtual reality system will rely on a computer or console to be executed. The machine used to power the application must have a GPU and Processor powerful enough to support a virtual environment. This system will likely operate indoors and be sheltered from any harmful physical conditions.

## 1.4 Requirements

**Functional Requirements:**

- The User shall be able to load into a virtual Environment upon startup
- The User shall be able to move around and explore the world freely

- The User will have the ability to choose between a fly-through and walk-through mode
- The environment must be 1:1 scale with reality
- The Game must implement a soundscape for the environment

**Nonfunctional Requirements:**

- The rendering of the environment must be aesthetically pleasing
- The movement/usability while exploring environment must be simple and effective
- The application must run on a reasonably priced and attainable computer.
- The Game must contain elements of procedural generation

## 1.5 Intended Users and Uses

Our user base has the potential to be very large. Anyone who is trying to explore a virtual world for the purpose of relaxation as well as anyone who wants to explore an environment that is 1:1 scale is our user. Our platform is going to be used as a form of a stress relief tool that helps people escape the "'normal" world and presents them with a virtual world that relaxes their mind.

## 1.6 Assumptions and Limitations

The product will be a web application.

It will be a single user application

To navigate, it will have six buttons: up, down, left, right, move forward, move backwards.

The limitations:

The hike will be limited to a single location.

It will be a first-person view.

We only have two semesters to work on the project

## 1.7 Expected End Product and Deliverables

Our deliverables consist of two products, The first one being the platform itself and the second being an instruction manual. We plan on releasing the first prototype at the end of first semester and the final product at the end of second semester.

We plan on delivering our product as a web platform. The platform will have two account types, "user" and "admin". The platform is going to have a 1:1 scale environment where the objects are created algorithmically using a procedural engine. The platform is going to be accessible to the user using a web portal. The user will have the ability to access the front end of the platform except. The admin will have access to the entire platform as well as the procedural engine which will allow the admin to tweak the parameters and type of the objects being created by it for the environment.

We also plan on delivering a manual to the client that will elaborate on the technologies used to create our platform. We also plan on including an instruction section in the manual as well as a walk through of the platform so that the client and the user is familiar with the product.

# 2 Project Plan

## 2.1 Task Decomposition

Tasks:

- Design Document - Chapter 1
- Design Document - Chapter 2
- Design Document - Chapter 3
- Design Document - Chapter 4
- Design Document - Chapter 5
- Design Document - Chapter 6
- Finalize Design Document
- Learning OpenGL and C++ (Research and Tutorials)
- Create and start coding the base of the application
- Create the procedural generation algorithm that will generate the world
    - Have the algorithm create all aspects of the world at a 1:1 scale
- Work on user movement within the world
    - Create a walkthrough mode and flying mode
- Work on collision detection
- Create a nature themed soundscape for the world

## 2.2 Risks And Risk Management/Mitigation

- Learning OpenGL and C++ (Research and Tutorials)

  This will be the first time working with OpenGL for every one in the team and a few of us are a bit familiar with C++. As such we have the risk of dealing with a steep learning curve and its hurdles to overcome. The estimated probability for this risk is 0.3.

- Create and start coding the base of the application
- Create the procedural generation algorithm that will generate the world
- Work on user movement within the world
- Work on collision detection
- Create a nature themed soundscape for the world

The risk we might face in these areas stems from our lack of experience with algorithms regarding procedural generation. The estimated probability for all these risks is 0.6. For our risk mitigation plan, we have decided to follow the ‘watch and monitor’ risk mitigation strategy which involves monitoring the project for risks and consequences and identifying any changes or shortcomings that can affect the impact of the risk. For example, if the teammates that are working on tasks such as developing a procedural generation algorithm run into any problems or are stuck at writing the algorithm, the rest of our teammates will try to step in and help them solve the problem.

Other potential risks might include,

- Bugs in code that are time consuming to solve, estimated probability for this risk is 0.4.
- Scheduling conflicts with teammates, estimated probability for this risk is 0.1.

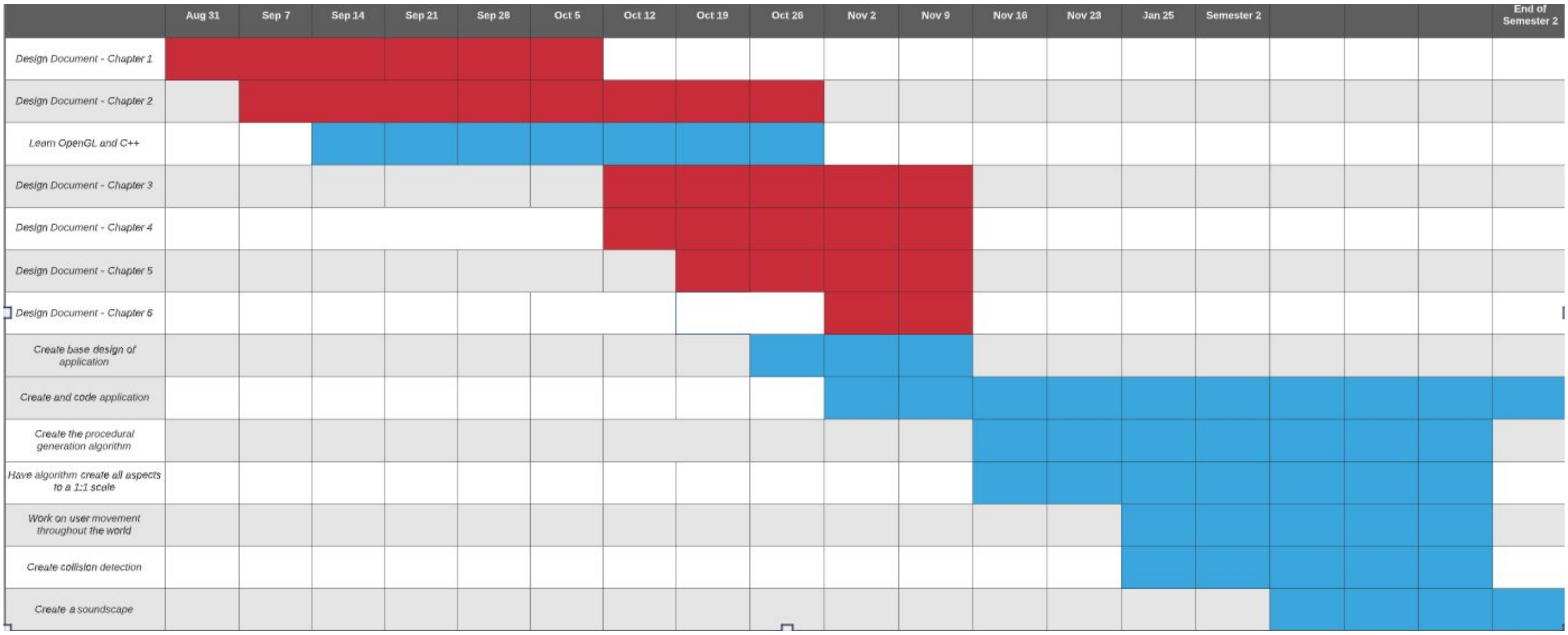
### 2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Our team has decided to set the milestones based on the project schedule. Our first milestone is to have the first chapter of the design document polished by the end of the first week of October 2020 which is week 7. Another milestone is to be done with the second chapter of the design document by week 10, other chapters of the design document subsequently following the project schedule. Our team has planned to be versed in OpenGL as well as C++ by the beginning of the first week of November by watching and doing tutorials. We plan on distributing roles to our team members by the end of the second week of November 2020. Our team plans on starting to learn about procedural generation beginning the first week of November. Another key milestone would be for our team to start implementing our learnings and start working on the prototype of the project beginning the second week of November 2020. Overall we plan on being done with the prototype by the end of December 2020 and the entire project by the end of May 2021.

For the project metrics, The application is going to be a 1:1 scale model of the Earth’s environment. The platform will have procedural generation implemented that will generate objects for the environment dynamically using an algorithm which will be developed by our team. The platform will have the option for the user to be able to choose between walkthrough mode as well as flying mode. The application will also be equipped with collision detection that will detect any kind of contact as well as collision between the user and the procedurally generated objects.

For the evaluation criteria, We plan on evaluating the work done by using many types of tests for example; Unit testing, System testing, Stress testing, etc. We plan on measuring the progress of the project by using Trello. Since we haven’t begun developing our project platform/application, we have not settled on the final metrics and the test cases yet.

## 2.4 Project Timeline/Schedule



| | Aug 31 | Sep 7 | Sep 14 | Sep 21 | Sep 28 | Oct 5 | Oct 12 | Oct 19 | Oct 26 | Nov 2 | Nov 9 | Nov 16 | Nov 23 | Jan 25 | Semester 2 | | | | End of Semester 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Design Document - Chapter 1* | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | | | | | | | | | | | | | |
| *Design Document - Chapter 2* | | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | | | | | | | | | | |
| *Learn OpenGL and C++* | | | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | | | | | | | | | | |
| *Design Document - Chapter 3* | | | | | | | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | | | | | | | | |
| *Design Document - Chapter 4* | | | | | | | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | | | | | | | | |
| *Design Document - Chapter 5* | | | | | | | | 🟥 | 🟥 | 🟥 | 🟥 | | | | | | | | |
| *Design Document - Chapter 6* | | | | | | | | | | 🟥 | 🟥 | | | | | | | | |
| *Create base design of application* | | | | | | | | | 🟦 | 🟦 | 🟦 | | | | | | | | |
| *Create and code application* | | | | | | | | | | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 |
| *Create the procedural generation algorithm* | | | | | | | | | | | | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | |
| *Have algorithm create all aspects to a 1:1 scale* | | | | | | | | | | | | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | |
| *Work on user movement throughout the world* | | | | | | | | | | | | | | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | |
| *Create collision detection* | | | | | | | | | | | | | | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | |
| *Create a soundscape* | | | | | | | | | | | | | | | | 🟦 | 🟦 | 🟦 | 🟦 |

The following information represents all of the tasks with their corresponding due dates, which is also shown in the Gantt chart above.

- Design Document - Chapter 1: October 4
- Design Document - Chapter 2: October 25
- Design Document - Chapter 3: November 15
- Design Document - Chapter 4: November 15
- Design Document - Chapter 5: November 15
- Design Document - Chapter 6: November 15
- Finalize Design Document: November 15
- Learning OpenGL and C++ (Research and Tutorials): November 2
- Create and start coding the base of the application: November 16

The rest of the tasks will be started within the first semester but will not be finished until the second semester, and they will all be worked on simultaneously.

- Create the procedural generation algorithm that will generate the world: End of second semester
  - Have the algorithm create all aspects of the world at a 1:1 scale: End of second semester
- Work on user movement within the world: End of second semester
  - Create a walkthrough mode and flying mode: End of second semester
- Work on collision detection: End of second semester

We will only end up creating a soundscape if we feel like we have ample time, our client said this is not a full on requirement and to only do it if we have time at the end of the second semester.

- Create a nature themed soundscape for the world: End of second semester

## 2.5 Project Tracking Procedures

For this part, we plan to use four different types of web-based platforms. They each play an important role in keeping our project on track. They are:

1. Trello board: We use this to distribute tasks and keep track of who is doing what part of the project on a weekly basis.
2. Discord: This is where we do most of our communications, we use this for anything. This includes questions, meeting planning etc.
3. GitLab: This is for storing and maintaining our code for the project. It also helps us keep the code structured and safe. With this, we can also experiment and try new ideas because of the branch feature of GitLab.
4. WebX: This is what we use to have our weekly meetings. On this platform, we have live discussion and assign tasks to group members. Also, we also have discussions with our project advisor and clarify any lingering concerns that discord communication cannot handle.

Together, all these four platforms keep us organized, on track and synchronize with each other.

## 2.6 Personnel Effort Requirements

| Task | Number of Hours Required | Description |
|---|---|---|
| Research APIs/Frameworks needed to design project | ~5 hours per person | Every member of the team spent the first week researching the tools necessary to begin work on the project. These topics included a recommended IDE and graphics API. |
| Complete OpenGL tutorials | ~6 hrs per week<br>3 weeks | Each team member spent a minimum of 5 hours a week completing OpenGL tutorials. This was intended to introduce the group to the platforms we will be working with, as well as slowly develop an understanding of the steps needed to complete the project. |
| Create first OpenGL instance and render an image on headset | ~12 hrs per person<br>2 weeks | Every member will work on creating a working OpenGL program that displays on a VR headset |
| Design procedural generation algorithm | ~15 hrs per person<br>2 people<br>3 weeks | 2-3 team members work on deriving the procedural generation algorithm that will be used to create and implement textures/patterns |

### 2.7 Other Resource Requirements

- Microsoft Visual Studio
- OpenGL API

### 2.8 Financial Requirements

We do not need any financial resources for the project, we also do not have a budget.

## 3 Design

### 3.1 Previous Work And Literature

Include relevant background/literature review for the project

– If similar products exist in the market, describe what has already been done

– If you are following previous work, cite that and discuss the **advantages/shortcomings**

– Note that while you are not expected to "compete" with other existing products / research groups, you should be able to differentiate your project from what is available

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

### 3.2 Design Thinking

Detail any design thinking driven design "define" aspects that shape your design. Enumerate some of the other design choices that came up in your design thinking "ideate" phase.

### 3.3 Proposed Design

Include any/all possible methods of approach to solving the problem:

- Discuss what you have done so far – what have you tried/implemented/tested?
- Some discussion of how this design satisfies the **functional and non-functional requirements** of the project.
- If any **standards** are relevant to your project (e.g. IEEE standards, NIST standards) discuss the applicability of those standards here
- This design description should be in **sufficient detail** that another team of engineers can look through it and implement it.

### 3.4 Technology Considerations

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

### 3.5 Design Analysis

– Did your proposed design from 3.3 work? Why or why not?

– What are your observations, thoughts, and ideas to modify or iterate over the design?

### 3.6 Development Process

Discuss what development process you are following with a rationale for it – Waterfall, TDD, Agile. Note that this is not necessarily only for software projects. Development processes are applicable for all design projects.

### 3.7 Design Plan

Describe a design plan with respect to use-cases within the context of requirements, modules in your design (dependency/concurrency of modules through a module diagram, interfaces, architectural overview), module constraints tied to requirements.

## 4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or software.

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study or acceptance testing for functional and non-functional requirements).
2. Define/identify the individual items/units and interfaces to be tested.
3. Define, design, and develop the actual test cases.
4. Determine the anticipated test results for each test case
5. Perform the actual tests.
6. Evaluate the actual test results.
7. Make the necessary changes to the product being tested
8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you have determined.

### 4.1 Unit Testing

– Discuss any hardware/software units being tested in isolation

### 4.2 Interface Testing

– Discuss how the composition of two or more units (interfaces) are to be tested. Enumerate all the relevant interfaces in your design.

### 4.3 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

### 4.4 Results

– List and explain any and all results obtained so far during the testing phase

- Include failures and successes
- Explain what you learned and how you are planning to change the design iteratively as you progress with your project
- If you are including figures, please include captions and cite it in the text

## 5 Implementation

Describe any (preliminary) implementation plan for the next semester for your proposed design in 3.3.

## 6 Closing Material

### 6.1 Conclusion

Summarize the work you have done so far. Briefly reiterate your goals. Then, reiterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

### 6.2 References

List technical references and related work / market survey references. Do professional citation style (ex. IEEE).

### 6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar data that does not directly pertain to the problem but helps support it, include it here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc,. PCB testing issues etc., Software bugs etc.