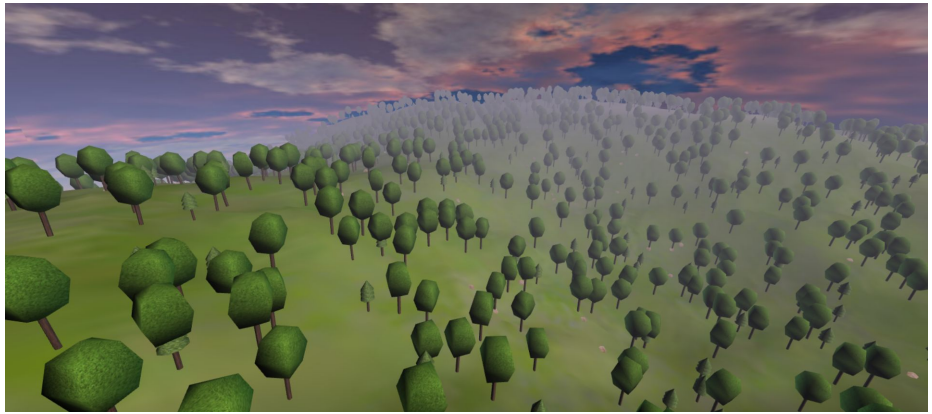
Take a Virtual Hike

Team: sdmay21-19
Advisor/Client: Mathew Wymore
Members: Trevor Nemes, Tyler Hassfield, Opeyemi Abass, Aashu Mallik, Akhilesh Ratnakumar, & Zian Li

Acknowledgements

We would like to take a second to thank and acknowledge our client and faculty member Professor Mathew Wymore for his time, effort, and support during the development of this project. All of his insight and guidance was heavily appreciated!

Project Overview



- Purpose: Create an explorable, full-scale 3D virtual nature environment in order to immerse the user can immerse themselves in relaxation and stress relief in the time of COVID-19
- Utilized the low-level graphics API, Three.js, to create and display our virtual world on a web browser.
- Our application will be a first-person movement based explorable 3D rendered world.

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 environment must include collision detection
- 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 (secondary requirement)

Non-Functional 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 application must contain elements of procedural generation

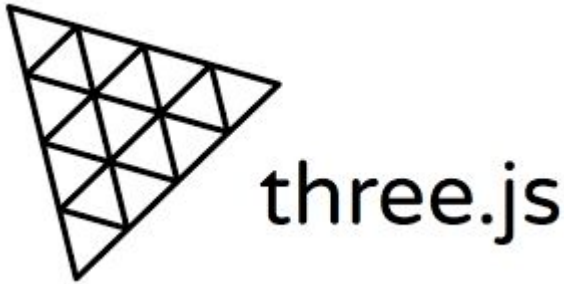
Engineering Standards

- 1362-1998 - IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps)
  - Provides a uniform scheme for preparing and presenting a concept of operations document
- IEEE 1448a-1996 - Standard for Information Technology - Software Life Cycle Processes
  - Establishes a common framework for software life cycle processes, with well-defined terminology, that can be referenced by the software industry
- IEEE 12207-1996 - ISO/IEC International Standard - Information Technology - Software Life Cycle Processes
  - Provides a common framework for developing and managing software. This standard provides industry a basis for software practices that would be usable for both national and international business

Engineering Constraints

- Use of THREE.js limits the programming language to JavaScript
- Achieve the Engineering requirement of running on a reasonably priced and attainable computer
- Time constraint: 2 semesters to get project done
    - Shortened second semester

Three.js



What is Three.js?

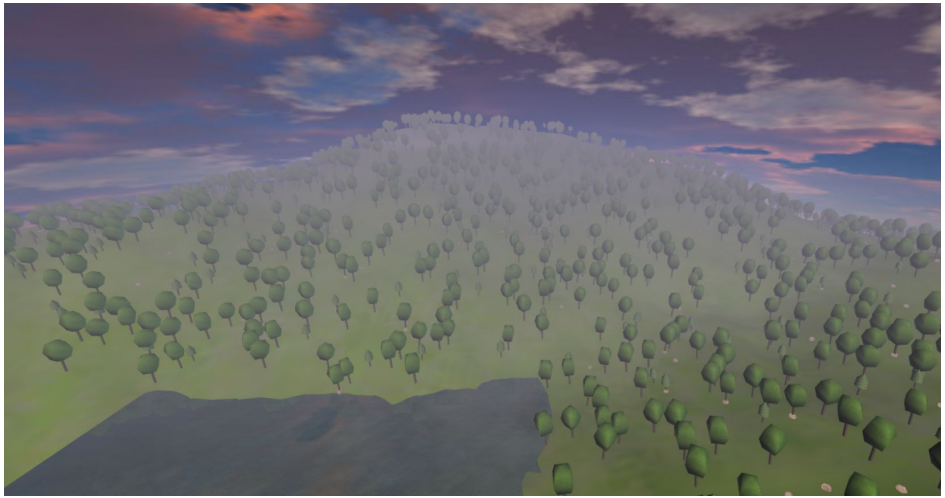
Three.js is a cross-browser JavaScript library and application programming interface used to create and display animated 3D computer graphics in a web browser using WebGL.

Why use Three.js?

- It is cross-browser which means it is cross platform. This is one of our project goals, to make the project run on all platforms.
- It ia a mature framework
- OpenGL is no longer supported on macOs

Procedural Generation

- Procedural generation allows for an endless continus world and reduced memory requirement.
- Procedurally generate meshes, terrain, and textures
- Perlin noise used for textures and terrain generation with different dimensions

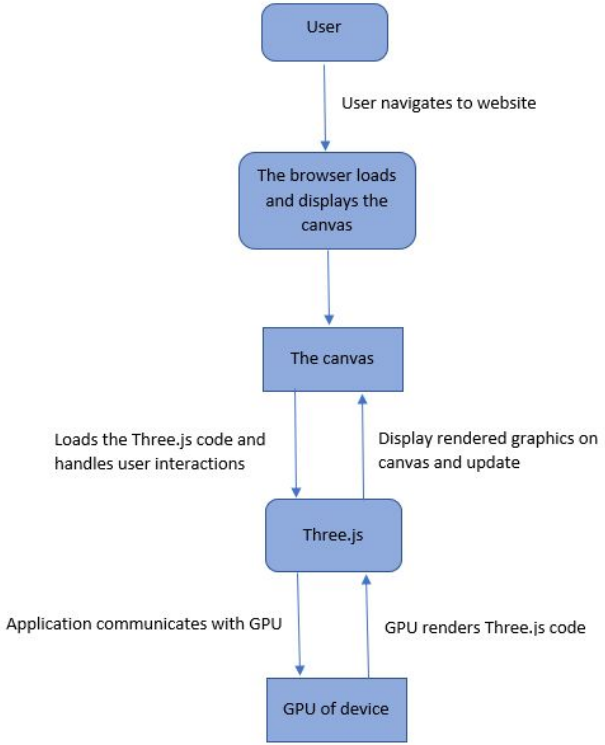


Implementation

- Trevor Nemes - Procedurally generated trees algorithm
- Tyler Hassfield - First person movement
- Opeyemi Abass - Procedurally generated bushes and rocks algorithm
- Aashu Mallik - Water generation and soundscape
- Akhilesh Ratnakumar - Terrain generation
- Zian Li - Worked on the sky and lighting

Application Design Diagram

1. Navigate to website
2. Browser loads and display the canvas
3. The canvas loads the three.js code and handles user interaction
4. The three.js code communicate with the computer gpu
5. The gpu renders the three.js code
6. The canvas displays the rendered graphics



Challenges Faced

- Integrating the first-person walking controls with the terrain
- Coordinating tasks between teammates while being remote
- Lack of experience
- Matching the objects on terrain and water levels with terrain scaling

Final Results

- Working model of the game where users can walk around and explore the environment.
- Project can run on all web browsers
- Terrain generates different heights
- Trees, rocks and water planes are generated on the terrain
- Achieve immersion for the user
- Made the game user friendly

What We Learned

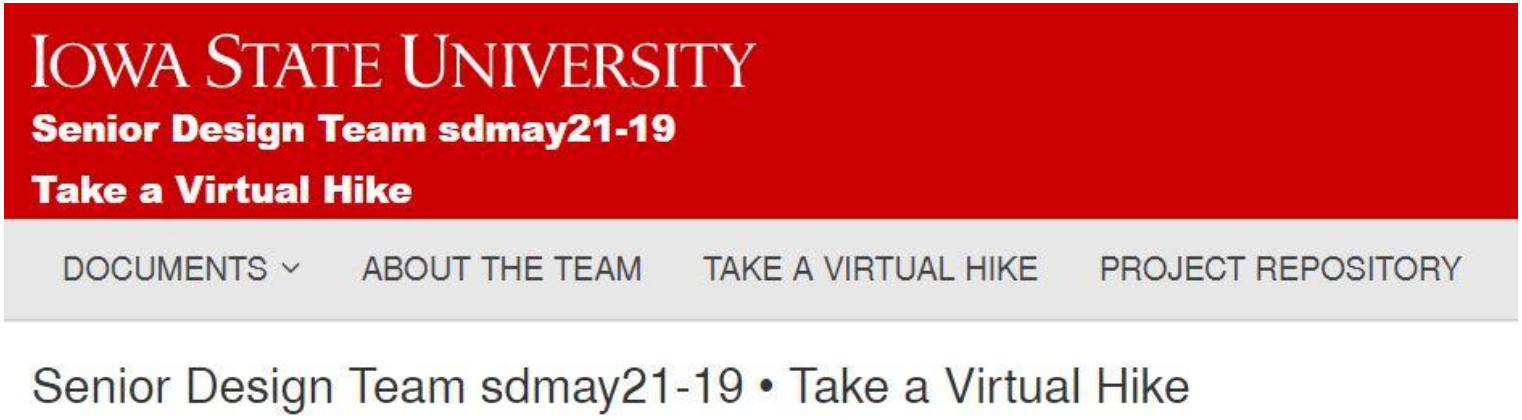
- We learned to procedurally generate objects using the Perlin Noise Algorithm
- We learned to create and render scalable 3d virtual environments in a web browser using Three.js and WebGL
- We learned to implement collision detection into our application
- We learned to implement a soundscape to make the virtual environment seem realistic
- We learned to implement first person view controls for the user to navigate the environment
- We learned to use Git for version control and issue tracking
- We learned to work in an agile environment as a team

How to Access

Access our application with the following link:

Team website: http://sdmay21-19.sd.ece.iastate.edu/

- From the home page, clicking the “Take a Virtual Hike” tab will load the application



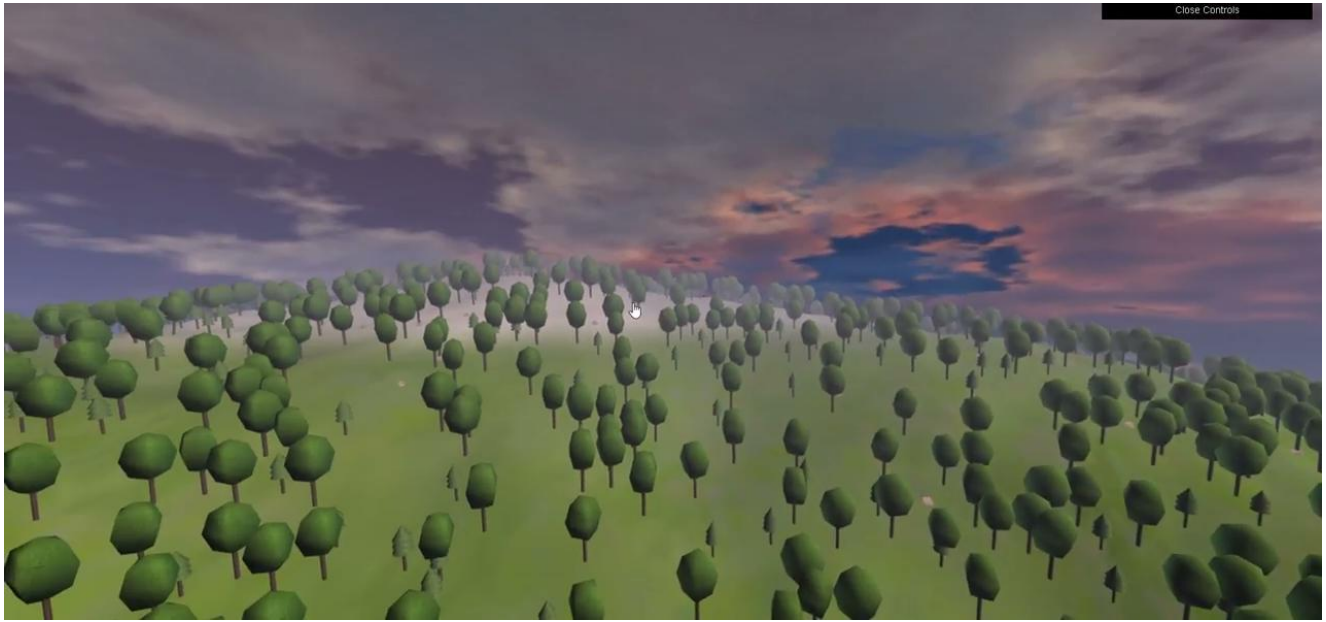
How to Play

The application allows the user to navigate the virtual environment with the following controls:

- Mouse - Look around
- W/up arrow - move forward
- A/left arrow - move left
- S/down arrow - move backwards
- D/right arrow - move right
- Space bar - Jump
- F - toggle flying mode
  - Shift - move down
  - Space bar - move up
- Esc - releases mouse from screen lock and returns to normal use.

Demo

https://youtu.be/5b8P6iKovUQ



Thanks for listening!

Any Questions?