StormSynth Draft Proposal

September 17, 2024

1 StormSynth: AI-Driven Storm Simulation and Visualization

- **Date:** September 17, 2024
- **Team Members:** Owen Kroeger
- **Professor:** Dr. Artzi
- **Project Jira:** https://oskroeger.atlassian.net/jira/software/projects/SCRUM/boards/1
- **Confluence:** https://oskroeger.atlassian.net/wiki/home
- **GitHub Project:** https://github.com/oskroeger/StormSynth

1.1 Project Overview

The goal of this project is to create a 3D visualization of storm development that allows users to explore the birth, lifecycle, and dissipation of storms based on input environmental parameters. This project will leverage neural networks to recreate storm behaviors using existing simulation data and will be implemented in Unreal Engine to provide a dynamic, interactive user experience.

1.2 Project Objectives

1. **Conceptualization and Design**:
   - Develop a detailed understanding of storm dynamics and lifecycle stages.
   - Utilize existing storm simulation data and train neural networks to replicate these phenomena within Unreal Engine.
   - Design a user interface that allows users to adjust environmental parameters and observe the resulting changes in storm behavior.
2. **Problem or Challenge**:
   - **Weather Visualization**: Take real-world theoretical weather principles, and provide a real-time dynamic animation of a storm lifecycle with UI parameter modification capabilities.
3. **Implementation**:
   - **Theoretical Foundations**: Apply concepts from meteorology, neural networks, and 3D graphics.
   - **Architectural Design**: Structure the project into modules for data integration, neural network training, visualization, and user interaction.
   - **Working Code**: Develop code in C++ and Blueprints within Unreal Engine to control storm animations and user interactions, and Python for data preprocessing and neural network training.

- **Testing and Performance Assessment**: Conduct testing to validate the accuracy of storm simulations against real-world data and optimize performance for smooth real-time rendering.

4. **Team Collaboration**:
   - Ideally, a team of 2–3 will work on different modules, such as data preprocessing, neural network development, and UI/UX design, but may be a solo project depending on the interest level of other peers.
   - If the project complexity increases, it can be split into subprojects focusing on specific storm types.

1.3 Integration of Core Computer Science Areas

- **Artificial Intelligence and Machine Learning**: Training neural networks to understand and generate storm patterns based on input data.
- **Software Engineering**: Developing a strong architectural design that integrates AI, data visualization, and user interaction.
- **Computer Graphics**: Using Unreal Engine's advanced rendering and particle systems to create realistic storm visuals.
- **Human-Computer Interaction**: Designing an intuitive user interface that allows easy manipulation of parameters and exploration / experimentation with storm dynamics.

1.4 Technical Details

1. **Programming Languages and Tools**:
   - **C++ and Blueprints**: For implementing core functionalities and interactive features within Unreal Engine.
   - **Python**: For data preprocessing, training neural networks, and interfacing with Unreal Engine.
   - **Unreal Engine**: To create semi-realistic 3D visualizations and handle real-time interactions.
   - **TensorFlow/PyTorch**: For developing and training neural network models based on storm simulation data.
2. **User Interface (UI) Components**:
   - **Administration Mechanism**: Allows users to manage storm parameters such as humidity, wind speed, and temperature.
   - **Interaction Mechanism**: Users can interact with the storm model by adjusting parameters via sliders or input fields and navigating through the 3D environment.
   - **Output Mechanism**: The system generates visual outputs in the form of dynamic 3D storm models that evolve over time, reflecting changes in the input parameters.

1.5 Social and Professional Responsibilities

1. **Historical Context**: Reference the evolution of weather modeling and visualization, building on years of meteorological research.
2. **Ethical Considerations**: Ensure accurate and responsible representation of storm data to avoid misinformation, especially since the tool could be used in educational settings.
3. **Legal and Security Issues**: Comply with data usage agreements for any third-party simulation data and ensure the software is secure, especially if deployed in an educational context.

1.6 Local and Global Impact

1. **Individual Impact**: Provides an educational tool for individuals to better understand storm dynamics, potentially aiding in severe weather education.
2. **Organizational Impact**: Useful for educational institutions and meteorological organizations as a training tool.
3. **Societal Impact**: Contributes to broader weather literacy and awareness, with potential benefits in areas like meteorological education and public safety.
4. **Business and Economic Context**: The project could serve as a prototype for commercial applications in weather education, VR experiences, or professional meteorological training.
5. **Environmental Context**: By assising in the understanding of storm behavior, the project could indirectly contribute to more effective environmental management and severe weather awareness / response.

1.7 Conclusion

This project aims to deliver a semi-realistic and interactive 3D storm visualization experience that combines key areas of computer science, integrates theoretical and practical components, and addresses various possible educational and professional applications.

1.8 Next Steps

1. Finalize team (roles and responsibilities if team expands beyond 1).
2. Develop initial UI prototypes and outline core functionality requirements.
3. Begin data collection and preprocessing for basic neural network training.
4. Set up Unreal Engine project framework and integrate initial neural network outputs.
5. Synthesize simple animations based on initial neural network outputs.