Real-time Weather Rendering System

Requirement specification

BSc Thesis

The goal of this thesis is to research and implement a real-time, procedural, weather rendering system. The following document reveals the process of creating such a system from both a technical and mathematical perspective, considering different algorithms for techniques like noise generation and raymarching.

Field of Studies:	BSc in Computer Science
Specialization:	Computer perception and virtual reality
Author:	Matthias Thomann
Supervisor:	Prof. Urs Künzler
Date:	March 14, 2021
Version:	0.1

Contents

1 General

1.1 Purpose

This document serves the purpose of defining and clarifying the goals, which the thesis 'Realtime Weather Rendering System' is supposed to achieve. Furthermore, the requirement specification allows for a more accurate evaluation of the achievement of objectives and of the result itself.

1.2 Revision History

Version	**Date**	**Name**	**Comment**
0.1	February 27, 2021	Matthias Thomann	Initial draft
0.2	March 03, 2021	Matthias Thomann	Updated project schedule
0.3	March 11, 2021	Matthias Thomann	Determined requirements
0.4	March 14, 2021	Matthias Thomann	Added system overview diagram

2 Vision

3 Scope of Work

3.1 Initial Situation

In computer graphics, especially in games, an astonishingly large group of features are recurring across all programs and genres. With the most obvious ones being water surfaces, cloudscapes and fire effects, they are present in almost any game. Naturally, those features grew in complexity, customizability and computational demands over time.
One of the core mechanics for achieving realistic results is called a *volumetric shader*. A prototype such a shader has been created in a previous project and will be used as base.

3.1.1 Previous Work

In a previous project, the process of creating a volumetric shader has already been researched and implemented in a prototype. Thanks to its high flexibility, different cloudscapes could be rendered by the same shader.

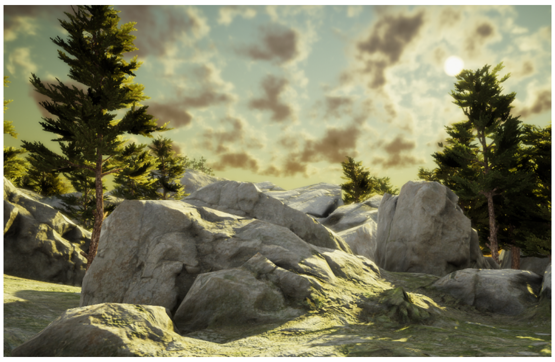


Figure 1: Result of the previous work's shader (Evening).

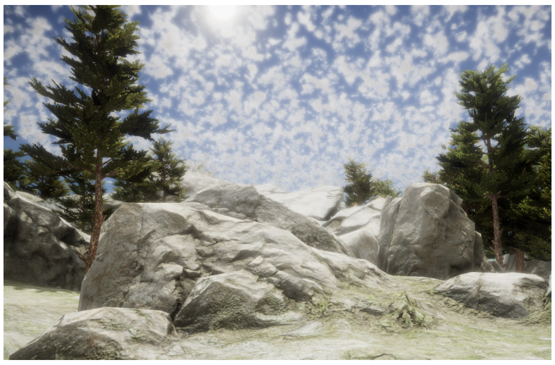


Figure 2: Result of the previous work's shader (Day).

During that project, some other important topics have been researched. Among those were volumetric rendering, Perlin and Voronoi noise generation algorithms, and a technique called *ray marching*.
The implementations of those algorithms and methods will most likely be reused in this thesis and will be adapted and improved accordingly.

3.2 Goals

As the title of the thesis suggests, this work will primarily focus on clouds and cloudscapes. The primary goal of the project is to research and implement rendering techniques for a real-time procedural weather rendering system.
The goals will be split into two distinct groups: mandatory and optional. However, this section only defines high-level goals. A detailed specification of all requirements can be found in section 4.

3.2.1 Mandatory Goals

The following tasks must be accomplished during the project:

- Understanding of different layers of clouds

- Understanding of compute shaders
- Implement a weather rendering system
- Incorporate real-time weather data from *meteoblue*

3.2.2 Optional Goals

For further optional work, these tasks can be looked into:

- Incorporate topological landscape models from *swisstopo*
- Automatic validation of realism of rendered cloudscapes
- Automatic comparison of rendered cloudscapes and photographs
- Automatic categorization of rendered cloudscapes
- Performance optimization

3.3 Vision

This section defines a high-level vision for future work involving the results and implementations of this thesis. As listed in the primary goals, the weather rendering system will be based on compute shaders. Compared to the prototype from the previous project, this is expected to result in a much better performance. That in turn, allows for a more complex and realistic model.
With the incorporation of real-time weather data and the use of topological landscape data, any given weather scenario could be simulated and rendered. The desired outcome ideally looks similar to the image depicted in Figure 3. A rendered version of such a cloud system can look elusively realistic compared to an actual photograph, like in Figure 4.

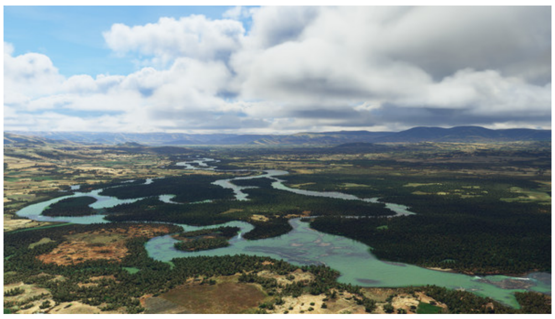


Figure 3: A rendered image of volumetric clouds [1].

Figure 4: A photographic reference of clouds [2].

The first thought about the practical use of a fully-fledged volumetric cloud system might be a video game, since clouds are often a significant part of outdoor scenery in games. However, for this thesis it is intended that the knowledge and results acquired during the given period will be used to recreate a lifelike weather system instead.
To accurately reflect a weather system, conditions like precipitation, wind and cloudiness will be considered. This data is obtained from the firm *meteoblue*.

3.4 Educational Objectives

Educational objectives include shader programming, knowledge about compute shaders, rendering techniques, common algorithms used in computer graphics like noise generation, a general understanding of aspects needed to create a complete weather system and finally the incorporation of real-time data from a third party.

3.5 Used Software and Tools

All documentation will be written in LaTeX with Visual Studio Code. The shader will be implemented in Unity. The chosen shader language is High Level Shading Language (HLSL). For the presentation, Microsoft PowerPoint will be used.

4 Requirements

All requirements are grouped by type. This results in two major groups, which are research and development requirements. Each one of the requirements is derived from a goal listed in subsection 3.2.

4.1 Research requirements

Number	R.1
Name	Understanding the basic nature of clouds
Description	In order to be able to recreate a realistically looking cloud shape, one has to examine and understand the way a cloud is formed first. This includes the color, density, shape as well other characteristics of clouds.

Number	R.2
Name	Understanding of different layers of clouds
Description	Among other characteristics, altitude, humidity and atmospheric pressure dictate the look and genus of a cloud. The goal is to decide which cloud types are required for a believable weather system.

Number	R.3
Name	Understanding of compute shaders
Description	Compute shaders proved to be a highly efficient tool when it comes to heavy calculations, like simulations. To improve performance and therefore allow for more a sophisticated weather system, compute shaders have to be researched.

4.2 Development requirements

Number	D.1
Name	Noise generation based on compute shaders
Description	To make full use of the power of compute shaders, it is best to let them execute computationally demanding tasks. In this case, specifically noise generation.

Number	D.2
Name	Incorporation of real-time weather data from *meteoblue*
Description	In order to achieve a high degree of realism, real-time weather data will be used. *Meteoblue* offers different data package contracts, of which the "basic_1h" is to be acquired. The usage of the data package requires a physical location. Since author lives in Switzerland, the chosen location is: • Bern, Switzerland

Number	D.3
Name	Periodcally store photographs of 360-degree cameras
Description	A comparison of real-time weather data and an actual photographic reference from that date and time will prove to be useful. This is why the images from such cameras will be stored on a local file system. There are many camera systems that offer 360-degree footage free of charge. The chosen system is that of the company *Seitz* called *Roundshot* [3], with these locations: • Roundshot camera Bantiger, Switzerland [4] • Roundshot camera Gurtenpark, Switzerland [5]

Number	D.4
Name	Implement a weather rendering system
Description	In order to achieve a high degree of realism, real-time weather data will be used. *Meteoblue* offers different data package contracts, of which the "basic_1h" is to be acquired [6]. The usage of the data package requires a physical location. Since author lives in Switzerland, the chosen location is: • Bern, Switzerland

5 Testing

5.1 Testing of Prototypes

The project will be implemented and tested in Unity. For testing, the following test cases can be used to verify and evaluate the implementation.

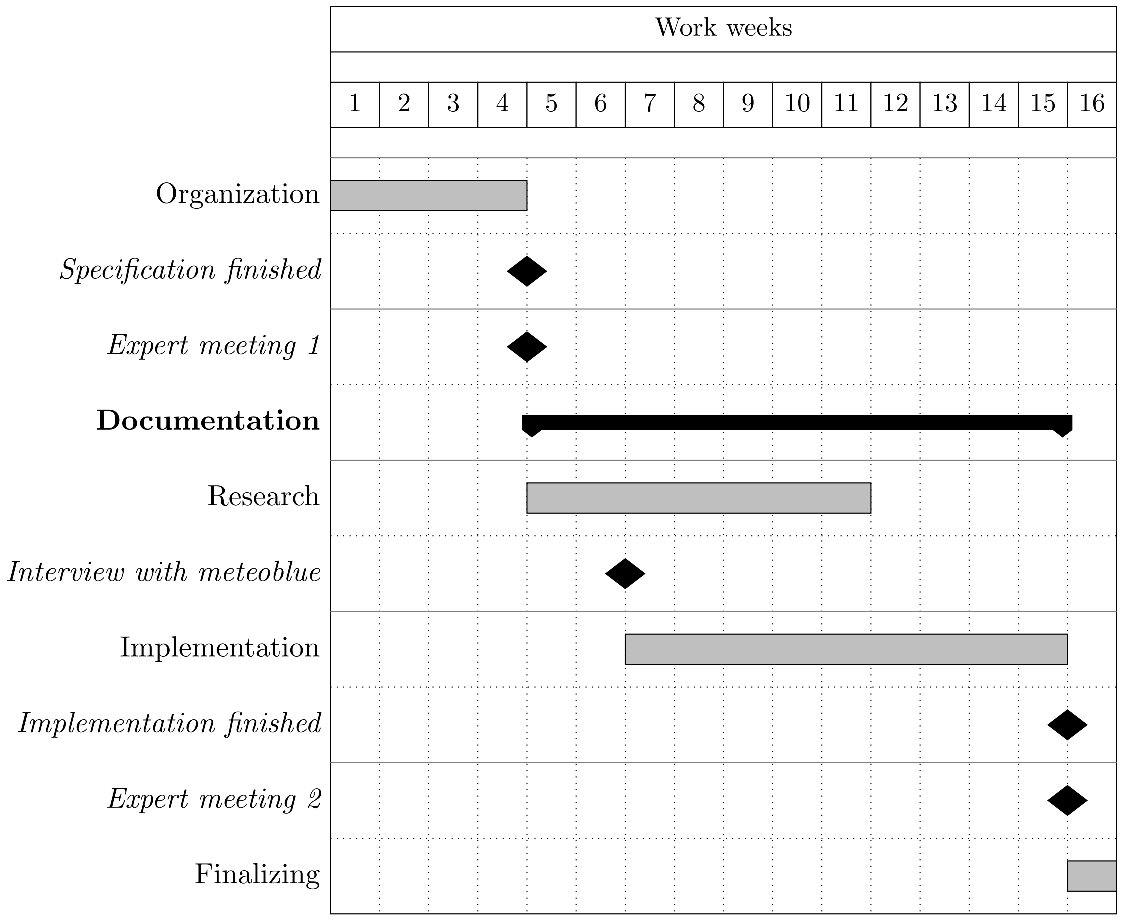
Case	**Test case**	**Expected result**
1	Working code	The shader code contains a vertex function as well as a fragment function and compiles successfully.
2	Shaded outcome	The rendered image corresponds with the outcome of the related prototype as described in subsection 4.2.
3	Visual comparison	The final rendered output should, to some extend, look similar to a photographic reference image, as seen in Figure 4. For comparison, photographs taken from installed Roundshot systems can be used.
3	Performance	The shader code should run with good performance and should not show visible stutters or frame drops.

6 Project management

6.1 Schedule

The time frame of the semester spans over exactly 16 weeks. Being worth 12 ECTS points, this project assumes a maximum work load of 22.5 hours per week, resulting in a total of 360 hours.

Over the course of the term, the project will be split into four primary task groups, namely organization, research, implementation and finalization. Put into relation with the duration of the project, the estimated schedule looks like this:



For each task group, the following distribution of time and effort is estimated:

Task group	**Predicted effort**
Organization	10%
Research	35%
Implementation	50%
Finalizing	5%

6.1.1 Organization

The first task group focuses on creating and finishing the project specification. This includes a technical interview with *meteoblue* as well as the first meeting with the examination expert, Dr. Eric Dubuis. The interview with the firm *meteoblue* will be scheduled and held during the first couple of weeks of the research task.

6.1.2 Research

The research spans over the course of almost two months. It also continues being active during the first half of the implementation. This is necessary, as the topics will be further investigated when implementing them, which results in more research.

6.1.3 Implementation

After researching each relevant topic thoroughly, the implementation can begin. In this task, the weather system will be created in Unity. Also, during research and implementation, the documentation will be continuously updated.

6.2 Project Organization

There are two kind of meetings during the project. They will be thoroughly documented in the project's journal.
Should a physical meeting be impossible for some reason, an online meeting via Microsoft Teams will be held instead.

6.2.1 Weekly meetings

A meeting will be held on a weekly basis to discuss the progress of the thesis, possibly arisen issues as well as planned work for the upcoming week.

Name	**Role**	**Participation**
Matthias Thomann	Author	Mandatory
Prof. Urs Künzler	Tutor and reviewer	Mandatory

6.2.2 Expert meetings

Additionally, both before the research task begins and after the implementation task has ended, a meeting with the external examination expert will be held.

Name	**Role**	**Participation**
Matthias Thomann	Author	Mandatory
Dr. Eric Dubuis	Examination expert	Mandatory
Prof. Urs Künzler	Tutor and reviewer	Optional

6.3 Project Results

The following items must be submitted for grading:

- **Documenation**
  The documentation includes this document as well as the thesis' scientific paper.
  - Requirement specification
  - Thesis paper
- **Implementation of the System**
  The Unity project, including all implemented shader code, will be managed and stored in the given Gitlab repository [7]. This will also serve as a form of submission for grading.
- **Presentation**
  A public presentation will be held on the second last friday of the term, June 9th, 2021.
- **Defense of the Thesis**
  The bachelor's thesis defense will be held after the term, on a day between June 21st, 2021 and July 14th, 2021. The exact date is yet to be announced.

- **Defense of the Thesis**
  The bachelor's thesis defense will be held after the term, on a day between June 21st, 2021 and July 14th, 2021. The exact date is yet to be announced.

6.3.1 Submission Terms

The following items must be submitted.

Item	**Description**	**Due Date**
Specification	This document	March 19th, 2021
Book entry	An advertising one-page description of the thesis	to be announced
Poster	An advertising poster of the thesis (A1 format)	June 7th, 2021
Video clip	An advertising one-minute video clip of the thesis	June 17th, 2021
Thesis	The thesis paper and all of the source code	June 17th, 2021
Thesis print	The printed thesis including a CD with all source code	June 21th, 2021

Glossary

Compute shader A shader which runs on the GPU but outside of the default render pipeline. 5

HLSL High Level Shading Language. 5

LaTeX A high-quality document preparation system designed for the production of technical and scientific documentation. 5

Noise generation Noise generation is used to generate textures of one or more dimension with seemingly random smooth transitions from black to white (zero to one). 3, 5, 7

Procedural Created solely with algorithms and independant of any prerequisites. i, 3

Ray marching Ray marching is a type of method to approximate the surface distance of a volumetric object, where a ray is cast into the volume and stepped forward until the surface is reached. 3

Shader A piece of software which runs on the GPU, rendering geometrically defined objects to the screen. 3, 5

Volumetric This describes a technique which takes a 3D volume of data and projects it to 2D. It is mostly used for transparent effects stored as a 3D image. 3

References

[1] *Rendered image of volumetric clouds.* [Online]. Available: `https://www.gosunoob.com/guides/change-live-weather-problems-in-flight-simulator-2020/`.

[2] *Photographic reference of clouds.* [Online]. Available: `https://bantiger.roundshot.com/`.

[3] *Seitz: 360° roundshot camera system.* [Online]. Available: `https://www.roundshot.com/`.

[4] *Roundshot camera: Bantiger.* [Online]. Available: `https://bantiger.roundshot.com/`.

[5] *Roundshot camera: Gurtenpark.* [Online]. Available: `https://gurtenpark.roundshot.com/`.

[6] *Meteoblue: Weather data packages.* [Online]. Available: `https://docs.meteoblue.com/en/apis/weather-data/introduction`.

[7] *Gitlab: Thesis repository.* [Online]. Available: `https://gitlab.ti.bfh.ch/cpvr-students/cloud-shader`.