Procedural cloud shader

Requirement specification

Project 2

The goal of this project is to research and implement a procedural, volumetric cloud shader. The following document reveals the process of creating such a shader 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 10, 2020

Berner Fachhochschule — Haute école spécialisée bernoise — Bern University of Applied Sciences

Contents

1 General

1.1 Purpose

This document serves the purpose of defining and clarifying the goals, which the project 'Procedural cloud shader' 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 29, 2020	Matthias Thomann	Initial draft
0.2	March 9, 2020	Matthias Thomann	Rework based on discussed points

2 Scope of Work

2.1 Initial Situation

With shaders making up a large part of visual effects in games and in game development generally, they have become more and more important throughout the years. Due to their high flexibility, it is possible to create a wide variation of implementations as well as cheap alternatives to otherwise highly complex and computationally demanding simulations, such as simulating water, fire or clouds.
This project specifically focuses on clouds. But to achieve a realistic look and feel of the clouds, certain methods and knowledge are required. The motivation for this project is to implement such a shader based on information gathered during the given period.

2.2 Goals

The primary goal of the project is to research and document rendering techniques for real-time procedural cloud shaders. Additionally, a prototype is to be implemented based on the newly discovered knowledge.

2.2.1 Mandatory Goals

The following tasks must be accomplished during the project:

- Understanding of the basic nature of clouds
- Understanding of what makes good clouds in games
- Research common methods and algorithms involved in rendering procedural clouds, including...
  - volumetric rendering
  - procedural noise generation algorithms
  - the concept of ray marching

2.2.2 Optional Goals

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

- Light scattering illumination and sub-surface scattering
- Performance optimization
- Simulation of gas

2.3 Vision

This section defines a high-level vision for future work involving the results and prototypes of this project.
As described in section 4, several prototypes about different rendering techniques will be created. By combining these outcomes and with the backing of all research findings, a complete cloud system could be achieved. The desired outcome ideally looks similar to the image depicted in Figure 1. A rendered version of such a cloud system can look elusively realistic compared to an actual photograph, like in Figure 2.

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

Figure 2: 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 project it is intended that the knowledge and prototypes 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 should be considered.

2.4 Educational Objectives

Educational objectives include shader programming, rendering techniques, common algorithms used in computer graphics, a general understanding about the role of clouds in games and other topics.

2.5 Used software

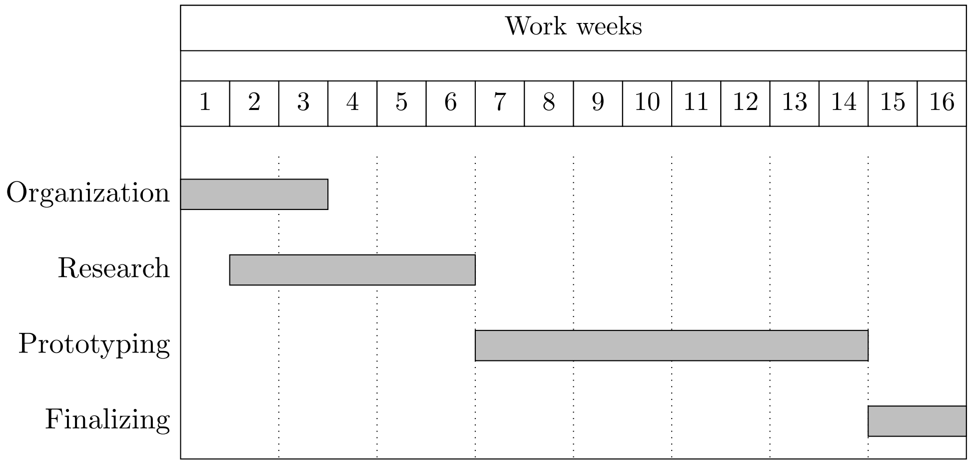
All documentation will be made in Visual Studio Code. The shader will be implemented in Unity. The chosen language is HLSL. For the presentation, Microsoft PowerPoint will be used.

3 Planning

3.1 Schedule

The time frame of the semester spans over exactly 16 weeks. Being worth 4 ECTS points, this project assumes a maximum work load of 8 hours per week, resulting in a total of 128 hours.

Over the course of the term, the project will be split into four primary task groups, namely organization, research, prototyping and finalization. Put into relation with the duration of the project, an 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	30%
Prototyping	50%
Finalizing	10%

4 Requirements

All requirements are split into two major groups, namely research and development requirements. Each of those are derived from the goals listed in subsection 2.2.

4.1 Research requirements

Number	R.1
Name	Understanding of 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 what makes good clouds in games
Description	Just as for real world clouds, clouds in games must also be examined. This will lead to a clear perception of what is important when rendering clouds in a simulated environment, opposed to the real world example.

Number	R.3
Name	Research common methods and algorithms involved in rendering procedural clouds
Description	As a matter of course, it is imperative to research and keep records of widely-used and popular approaches for rendering clouds must be researched and documented. Those methods and algorithms are to be compared and evaluated. They include at least the following topics: • volumetric rendering • procedural noise generation algorithms • the concept of ray marching

4.2 Development requirements

Number	D.1
Name	Implementing a prototype about volumetric rendering
Description	To fully comprehend volumetric rendering, it is best to implement a prototype, showcasing how such this rendering technique works and how it is built.

Number	D.2
Name	Implementing a prototype about noise generation
Description	It is necessary to understand procedural noise generation in order to achieve ”natural randomness” as it appears in nature. That is why a prototype must be built for commonly used noise algorithms.

Number	D.3
Name	Implementing a prototype about ray marching
Description	In shader programming, ray marching is a key technique to determine the surface distance of very complex objects, like a cloud. Thus, to see what ray marching is all about, a prototype must be built.

Number	D.4
Name	Evaluation of suitable parameters for a cloud shader
Description	When creating a shader, some parameters should be made public, so the user can influence the outcome of the shader by only tweaking said variables. It is best to find out which parameters are the most influential on the output of the shader.

5 Testing

5.1 Testing of Prototypes

All prototypes will be created and tested in Unity. For each one, the following test cases can be used to verify and evaluate the prototype.

Case	Test case	Expected result
1	Working code	The shader code contains a vertex function as well as a fragment function and compiles.
2	Shaded outcome	The rendered image corresponds with the outcome of the related prototype as described in subsection 4.2.

6 Results

6.1 Documentation

The following documents must be submitted for grading:

- Requirement specification
- Project documentation indcluding:
  - Conceptual formulation
  - Comparison and details fo methods and algorithms
  - Details of implementation
  - Result report

6.2 Implementation of the Shader

The Unity project, including the implemented shader code, will be managed and stored in the given Gitlab repository[3]. This will also serve as a form of submission for grading.

6.3 Presentation

A presentation will be held on the second last friday of the term, June 5th, 2020.

6.4 Submission terms

The project must be submitted in digital form before June 12th 2020, 12pm.

Glossary

HLSL High Level Shading Language. 3

References

[1] *Rendered image of volumetric clouds*. [Online]. Available: `https://100assets.ru/product/weather-maker-unity-weather-system-sky-water-volumetric-clouds-and-light/`.

[2] *Photographic reference of clouds*. [Online]. Available: `https://www.mnn.com/earth-matters/climate-weather/stories/types-of-clouds`.

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