2013-06-13

BTH-Blekinge Institute of Technology
Uppsats inlämnad som del av examination i
DV1446 Kandidatarbete i datavetenskap.

Bachelor thesis

Procedural Rendering of Geometry-Based Grass in Real-Time

Markus R. Tillman

Abstract

Since grass is abundant on our planet it plays an important role in the rendering of many different outdoor scenes. This study focuses on the real-time rendering of many individual grass blades with geometry. As a grass blade in real life is very thin and have a simple shape it can be represented with only a handful of vertices. The challenge is introduced when a meadow of grass is to be rendered as it can contain billions of grass blades. Two different algorithms were developed; one which use traditional vertex buffers to store and render the grass blades while the other makes use of textures. Quantitative data was generated from these algorithms. Among this data were images of the scene. These images were subjected to a questionnaire to collect qualitative information about the grass. All the generated data was then analyzed and interpreted to find advantages and disadvantages of the algorithms. The buffer-based algorithm was found to be slightly more computationally efficient compared to the texture-based algorithm. The quality of the visual result was perceived to be towards good while the realism was perceived as mediocre at best. The advantage of the texture-based algorithm is that it allows more options to handle the grass blades data when rendering. Using the terrain data to generate the grass blades was concluded to be advantageous. The realism of the grass could have been improved by using a grass blade texture as well as introducing variety in density and grass species.

Keywords: procedural grass rendering, grass rendering, real-time grass rendering.

Abstrakt

Eftersom gräs är rikligt på vår planet spelar den en viktig roll vid renderingen av många olika utomhusscener. Denna studie fokuserar på realtidsrendering av många individuella gräsblad med geometri. Eftersom ett gräsblad i verkligheten är mycket tunnt och har en enkel form kan den representeras med endast en handfull vertiser. Utmaningen introduceras när en äng av gräs ska renderas eftersom som den kan innehålla miljarder gräsblad. Två olika algoritmer utvecklades, en som använder traditionella vertex buffrar för att lagra och rendera gräsbladen medan den andra använder sig av texturer. Kvantitativ data genererades från dessa algoritmer. Bland denna data fanns bilder av scenen. Dessa bilder utsattes för ett frågeformulär för att samla in kvalitativ information om gräset. All den data som genereras analyserades och tolkades för att hitta fördelar och nackdelar med algoritmerna. Den bufferbaserade algoritmen upptäcktes vara beräkningsmässigt effektivare jämfört med den texturbaserade algoritmen. Den upplevda kvalitén på det visuella resultatet ansågs vara närmare bra medan realismen uppfattades som medioker i bästa fall. Fördelen med den texturen-baserad algoritm är att den tillåter fler möjligheter att hantera gräsblads-data vid rendering. Slutsatsen av att använda terrängens data för att generera gräsbladen sågs vara fördelaktigt. Realismen av gräset kunde förbättrats genom att använda en gräsblads-textur, samt variation i densitet och gräsarter.

Tillman Markus.

Address: Sweden, Blekinge län, Karlskrona, Amiralitetsgatan 6a.

Email: markustillman91@gmail.com.

Supervisor: Sennersten Charlotte.

ISSN-number:

The reason I chose this field of work started when I was searching different techniques of how to render grass where I stumbled upon Brano Kemen's blog [1]. I found his work interesting where his take on representing individual grass blades with geometry was something I had not seen anywhere else before. This made me want to pursue and research different ways to render individual grass blades with geometry which made me choose this particular field of work for this bachelor thesis.

Acknowledgements

I would like to acknowledge my supervisor Charlotte Sennersten for invaluable insight in writing this thesis, technical artist student Per Axmark for making many of the illustrations, Björn Jeffsell for providing the terrain texture, all the people who helped me with the qualitative aspect of this work and my home review panel of esteemed judges; mother, father and brother.

Table of contents

2013-06-13

BTH-Blekinge Institute of Technology
Uppsats inlämnad som del av examination i
DV1446 Kandidatarbete i datavetenskap.

Bachelor thesis

1 Introduction

Grass is very abundant in many places on our planet. It therefore plays an important role in the rendering of many natural outdoor scenes in computer graphics. Representing grass blades with 3D-geometry is redundant in many cases when many species of grass have very thin blades. The contribution would just be too small to be worth the extra computation. The geometric form of a grass blade is relatively simple compared to other objects such as a sphere and can be represented realistically with only a handful of vertices[1]. However, due to the abundance of grass blades, this poses a great challenge when introduced to computer graphics. The sheer amount of grass blades needed to be rendered in a field of grass just simply makes it impossible to do in real-time with today's hardware. Therefore, approximations have to be made when representing the shape of the grass, its physical simulation and lighting calculations. For this study, real-time is defined as: no noticeable delay experienced by the user where feedback is perceived as instantaneous. This is defined to be at 60 frames per second (FPS[2]) or higher. Games and simulators provide a particular big challenge where the demand for realism is high and the grass-rendering technique has to share resources while still maintaining a real-time performance. According to David Mc Feely: "Gamers and consumers are demanding greater realism…" [2]. This is in good agreement with what Boulanger stated in 2005, even if it was 8 years ago: "Grass rendering seems to be a research subject of little interest. In fact, numerous video games companies are interested in it. Nowadays, an important competition exist between these companies to create the most beautiful games. Studying grass rendering will allow for more natural looking images since more and more games take place in outdoor environments." [3].

1.1 Background

The rendering of individual blades of grass using geometry gives a detailed and realistic impression and this study explores the limits of this technique on current hardware. It was stated in GPU GEMS that a detailed modeling of the individual blades of grass is not meaningful, because the number of polygons that would be required for larger meadows would be much too high [4]. However, representing individual grass blades with geometry has become more and more appealing in later years. This is mainly due to today's graphics cards being able to render millions of polygons in real-time. This makes grass blades represented with geometry a possibility for real-time rendering. According to Edward Lee, games such as the PSN[3] title *Flower*[4] have proven that grass rendering is possible on current console technology [5]. However, a geometry-based approach of rendering grass blades is still naïve because it is computationally heavy compared to other methods. Also, most of this computation is a waste in the case where a grass blade is far from the viewer and do not contribute much or anything to the final image of a scene. As Kévin Boulanger puts it: "Rendering lots of triangles for grass blades which have a size of a pixel is useless" [3]. That said, rendering detailed grass near the viewer is still of interest and applicable with a geometry-based approach on current hardware. For rendering grass in the distance, a computationally less expensive rendering technique is better suited. To combine the best use of resources, a level-of-detail (LOD)-system should be applied to reduce computation without noticeable loss of quality. A LOD-system consists of a set of levels, usually 3, where the first (closest) in relation to the camera position is said to be the foreground and the following levels are called middleground and background. The purpose of each consecutive level is to reduce the computation without the loss of quality. This is

[1] A vertex is a set of attributes defining a point in space.
[2] Not to be confused with first-person shooter.
[3] Playstation Network.
[4] Developed by thatgamecompany™. Website: http://thatgamecompany.com/. [Accessed: 16-May-2013].

reinforced by the fact that less and less detail of an object contribute to the final image of a scene the further away the object is from the camera.

1.1.1 Related Work

Much research about rendering grass in real-time has been done over the years. The methods developed and commonly used are geometry, volume and image-based rendering where the use of billboards seems to be, or at least, have been the most popular one. Another method that will be briefly presented in the end of this chapter is the use of NURBS[5]-curves. See Figure 1 for an illustration of using these methods to render an object. The volume-based rendering method will cover 2 different methods, one where voxels[6] are used and one where 2D-slices are used. One important factor is what type of grass one is trying to model and another is how accurately it shall be done. In other words; what the budget and limitations are. The realistic appearance of grass, or any other form of natural vegetation, depends on two aspects: precise modeling of the shape, and accurate lighting of the model [6].

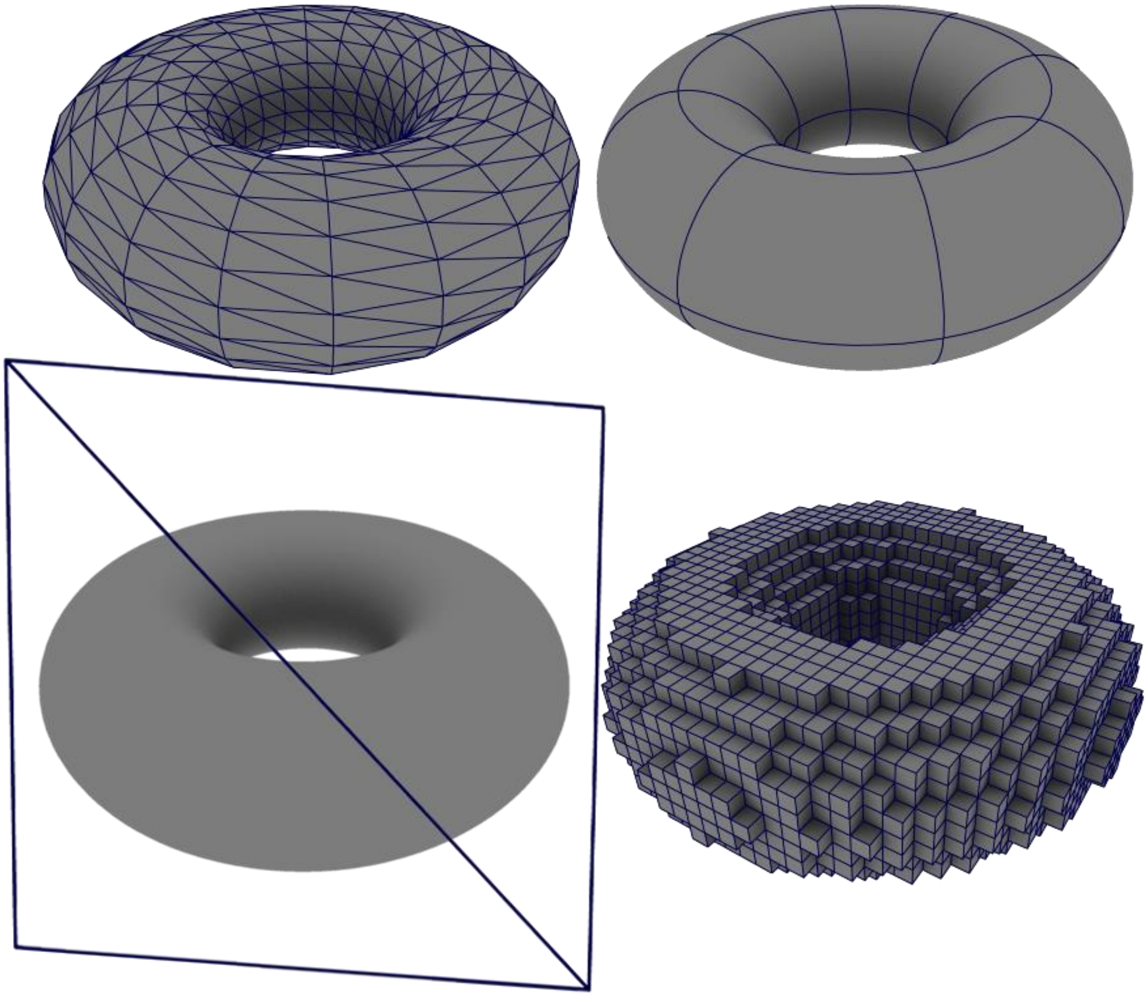


Figure 1. *An object is represented using different methods. Top left: Geometry. Top right: NURBS-curves. Bottom left: Image-based method using a textured billboard. Bottom right: Volume-based method using voxels. (Illustration made by Per Axmark.)*

[5] Non-Uniform Rational Basis-Spline.
[6] Volumetric pixels.

Technology is progressing with a rapid pace which means hardware becomes more and more computationally efficient. This makes the geometry-based rendering of grass more and more appealing for games and simulations with grass as this method offer a high level of detail.

1.1.1.1 Geometry-Based Rendering

Geometry-based rendering is a precise method of modeling grass. Modeling individual grass blades with this method is easy and also provides more detailed lighting than other methods. Another advantage of this method is that objects modeled with geometry offer full parallax[7] effect. An example of using geometry to represent an object is shown in Figure 2.

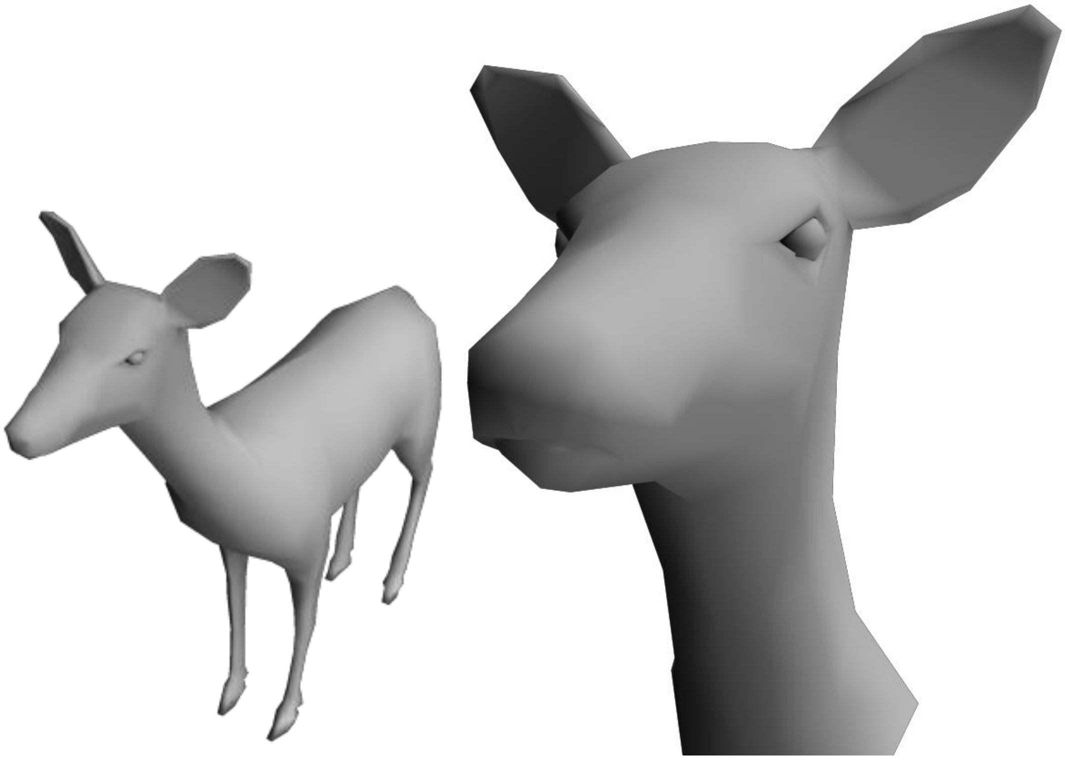


Figure 2. *An object represented with geometry. To the left: An overall view of the object. To the right: A close view of the same object. (Illustration made by Per Axmark.)*

There are many different ways on how the modeling of a grass blade can be done. Boulanger et al.[7] and Lemme[8] use a LOD-system where the first level consists of grass blades modeled with geometry. They approximate them with two-sided quadrilateral strips of zero thickness together with a texture with transparency to remove the edges and give the blades a shape more similar to grass blades in real life. To define the shape of the grass blades Boulanger et al.[7] use trajectories generated by a particle system presented by Reeves[9]. Another method to define the shape of a grass blade is the use of cubic Hermite curves[10, pp. 587–589].

[7] The apparent displacement of an object due to camera movement. E.g. objects in the distance appear to move slower than objects closer to the camera.

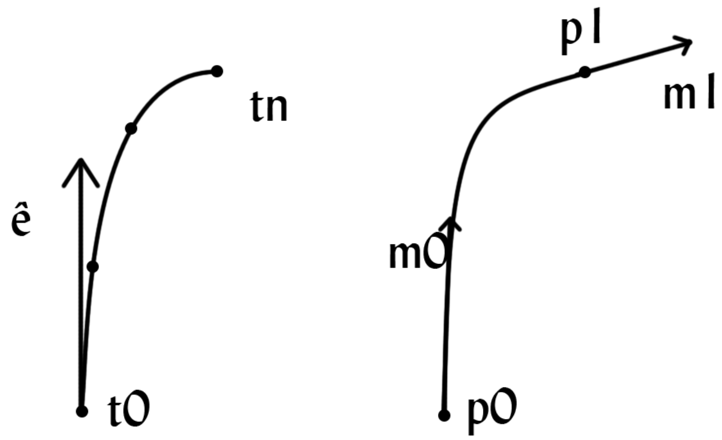


Figure 3. *To the left: A trajectory of a particle. To the right: A cubic Hermite curve. (Illustration made by Per Axmark.)*

Another solution proposed by Perbet and Cani[11] is to represent the grass blade as a chain of line-segment primitives. Guerraz et al.[12] presents a similar method where they use a hermite curve and a chain of billboards instead. Like Boulanger, Lee[5] uses a LOD-system. But instead of using a different rendering method for the different LOD-levels, he reduces the amount of points[8] that make up the shape of the grass blades and the density of the grass for each consecutive level. Another system which uses geometry to represent individual grass blades in each LOD-level is presented by Kemen[1]. The difference is that he keeps the same amount of points per grass blade throughout all the LOD-levels. Instead, in each consecutive level, he reduces the number of blades generated from a single point on the terrain. Also, when the height of the grass blade is below a specified threshold, he makes two blades instead of just one while still maintaining the same number of points that make up the shape of a single bladed grass blade. This doubles the apparent density for the short grass.

1.1.1.2 Volume-Based Rendering.

This method renders objects represented by voxels. These voxels each represent a volume of space [10, p. 502]. The benefit of this technique is that it allows the possibility to look inside an object, see Figure 4. For a visualization of storing an object represented by voxels within a 3D-texture and a more detailed object represented with voxels, see Figure 5 and Figure 6.

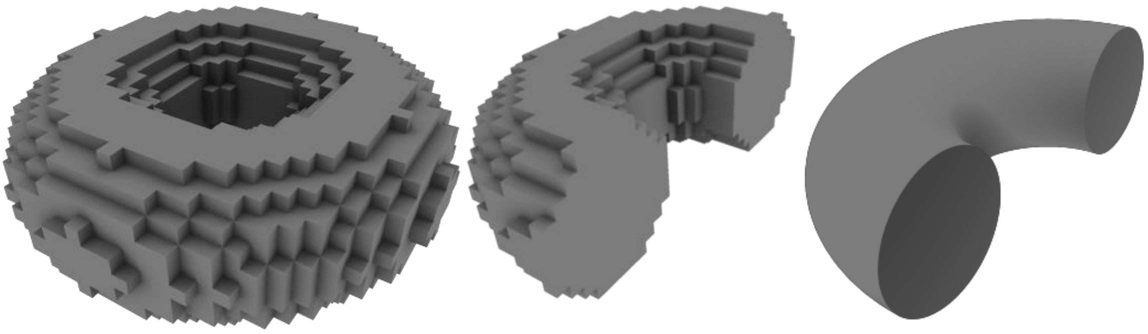


Figure 4. *An illustration of the insides of an object represented with voxels. Left: Original object from Figure 1. Middle: Same object cut in half. Right: Same object cut in half at high resolution with different lighting to highlight the cut. (Illustration made by Per Axmark.)*

[8] A point in this study is defined as a position in 3D space.

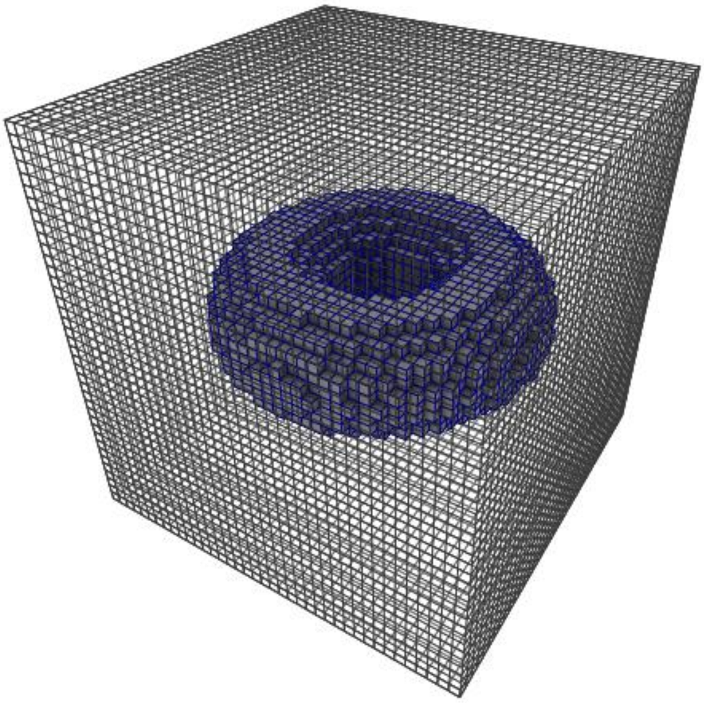


Figure 5. *Visualization of a 3D-texture containing volumetric data of an object. (Illustration made by Per Axmark.)*

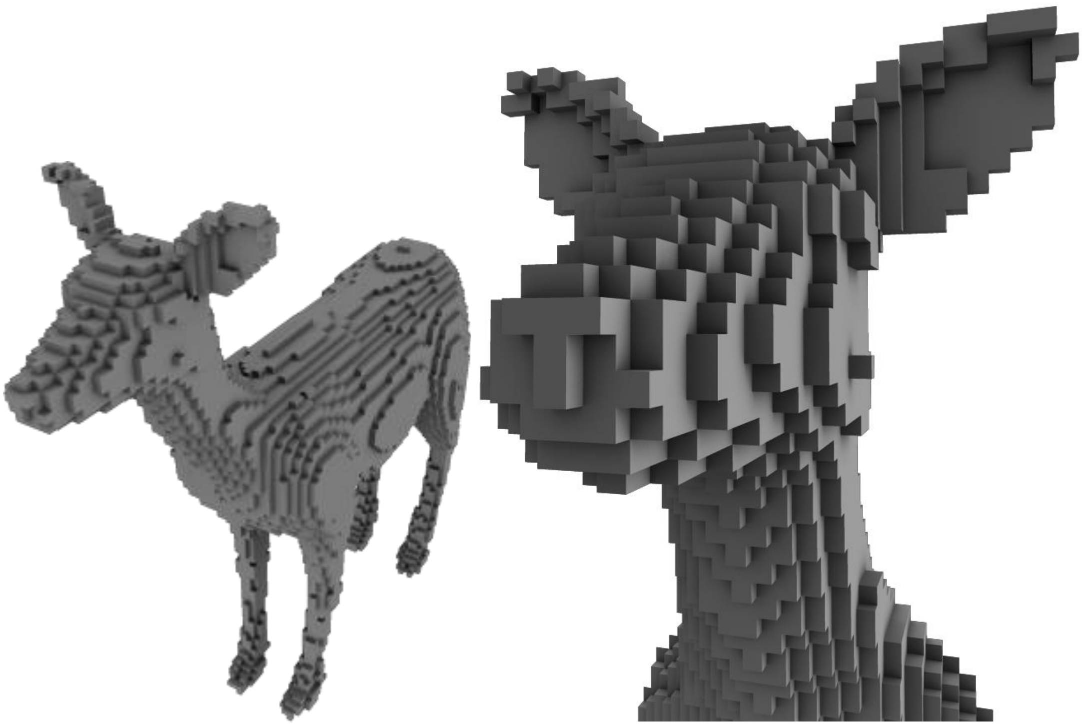


Figure 6. *An object represented with voxels. To the left: An overall view of the object. To the right: a close view of the object. (Illustration made by Per Axmark.)*

A variation of this method is to treat the voxel data as a set of two-dimensional image slices and render them using a textured quadrilateral. “The volume dataset is sampled by a set of equally spaced slices in layers perpendicular to the view direction. These slice images are then rendered in sorted order so that alpha compositing works properly to form the image." [10, p. 502]. An illustration of how this can look is depicted in Figure 7 and Figure 8.

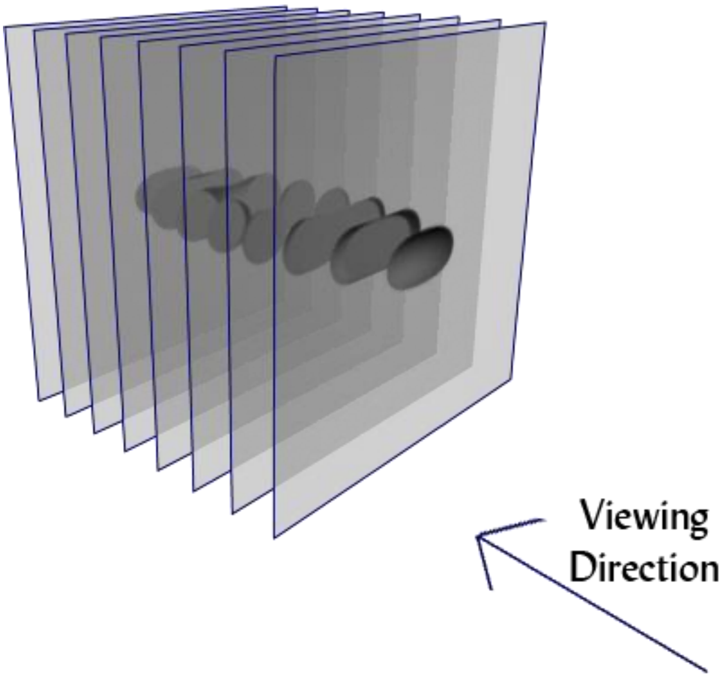


Figure 7. *An illustration of how a volume and its contents can be represented using 2D-slices. (Illustration made by Per Axmark.)*

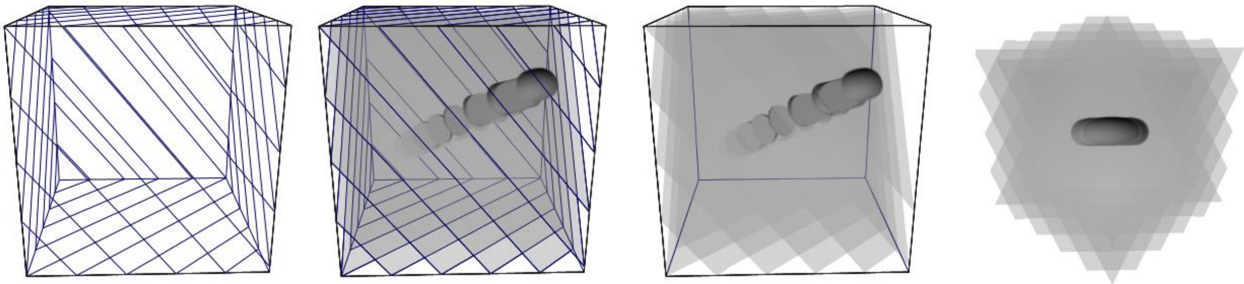


Figure 8. *An illustration showing the parts of volume rendering using slices. From left to right: The slices of the volume. The slices and its contents. The result when these slices are rendered together. Finally, the result when these slices are rendered together and directed towards the viewer without highlighting the bounding volume. (Illustration made by Per Axmark.)*

Instead of slices, Krüger and Westermann [13] developed the idea of casting rays[9] through the volume using the graphics processing unit (GPU)… the basic idea is that at each pixel, a ray is generated that passes through the volume, gathering color and transparency information from the volume at regular intervals along its length [10, p. 503]. See Figure 9 below for an illustration.

[9] Also called ray-tracing.

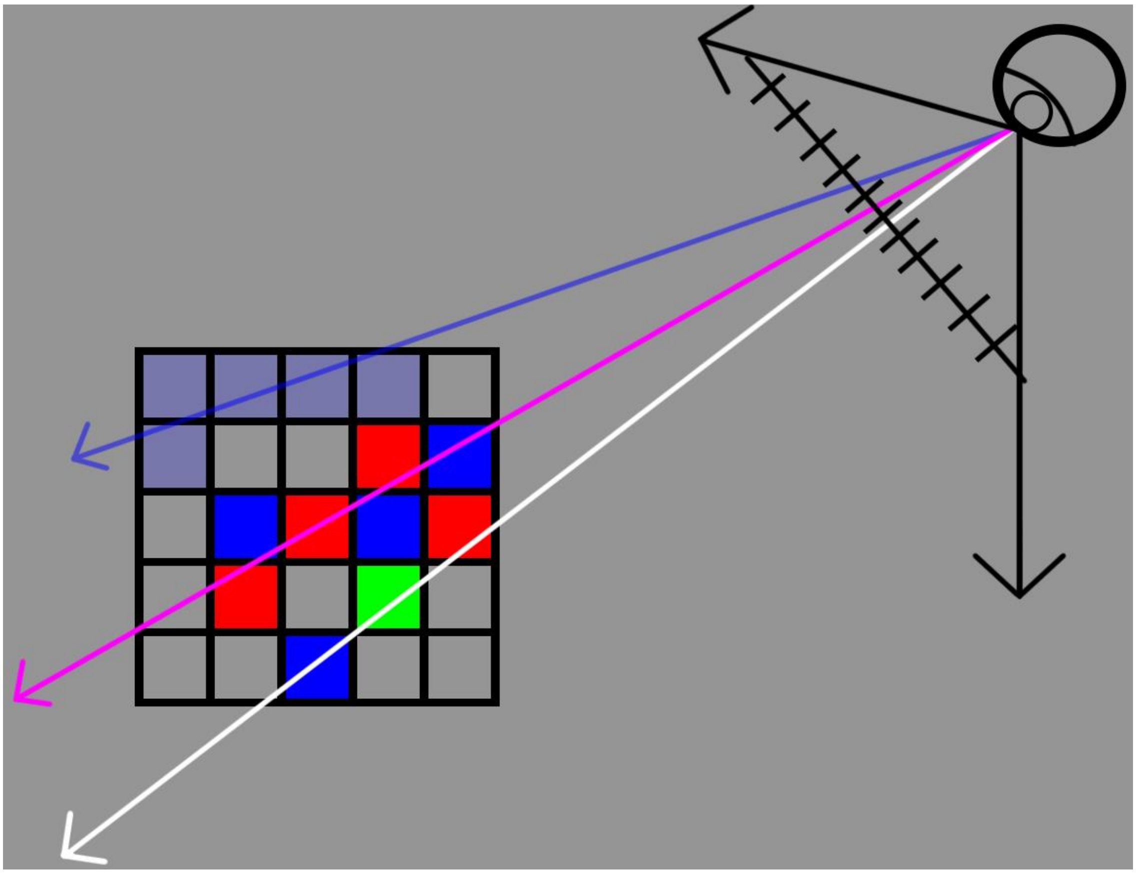


Figure 9. *An illustration of casting rays through a volume gathering color and alpha data using an infinite small value for the interval. How the color and alpha values are blended can differ. This image shows additive blending. (Illustration made by Per Axmark.)*

The advantage of volume rendering is that it very efficiently handles complex objects with many repetitive details. [3]. "…volumetric rendering can be used for grass rendered at middle distance from the viewer since it usually offers pleasant results with few aliasing." [3]. As grass blades are very thin and therefore has a very small volume, this method is not suited to render individual grass blades.

1.1.1.3 Image-Based Rendering

Rendering an image consists of 3 or 4 vertices making up a triangle or a quad which is covered with a 2D-texture. "A great advantage of representing an object with an image is that the rendering cost is proportional to the number of pixels rendered…" [10, p. 439]. Bump mapping methods such as normal mapping[10, pp. 187–190] for instance, can be applied to the image to give a sense of depth. An example of using normal mapping is shown in Figure 10 below.

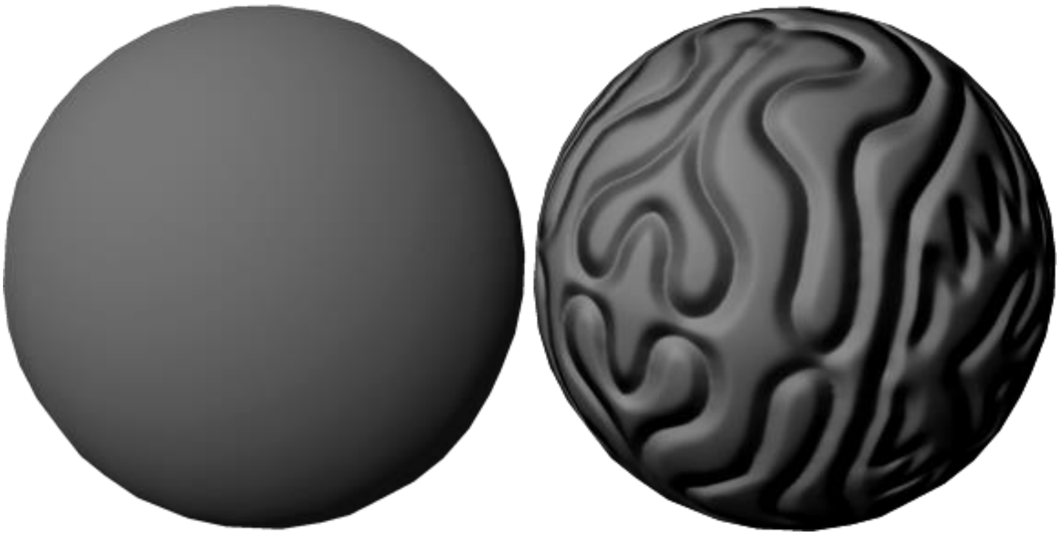


Figure 10. *To the left: An object rendered without normal mapping. To the right: Same object rendered with normal mapping. (Illustration made by Per Axmark.)*

The most common image-based method is billboards. A billboard is a textured polygon oriented based on the view direction [10, pp. 446–455]. It is also the cheaper among them in terms of computation due to its simplicity. See Figure 11 for a visualization and example of using a billboard.

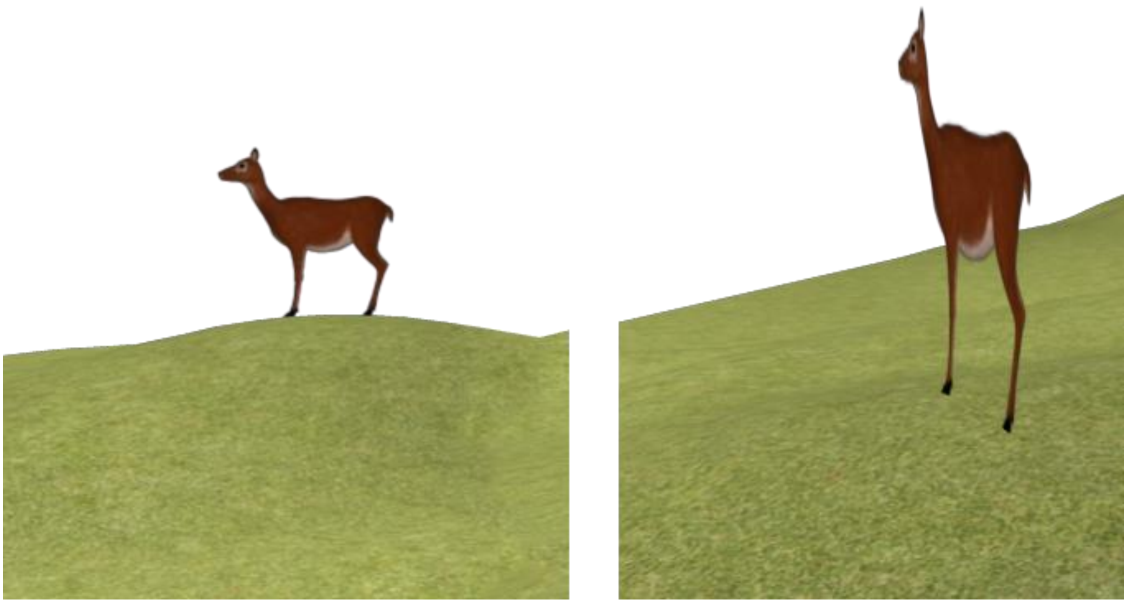


Figure 11. *A billboard with a texture of a deer seen from the side is oriented towards the viewer on the left. On the right, same billboard, still oriented as in the left image but seen from a different view point. (Illustration made by Per Axmark.)*

Due to their low complexity, they make an ideal method for rendering objects in the distance and/or where the need for detail is low. The drawback of this method is that it is prone to unwanted behavior. This behavior can happen when the view position and/or direction changes and when the billboard is used to represent an object that has an irregular surface. The effect becomes apparent when it is viewed up close while the viewer moves. This is a problem that is usually solved by using a LOD-system to switch the billboard with a higher detailed representation when the viewer moves closer to it.

1.1.1.4 NURBS-Curves Rendering.

Another method of modeling and rendering objects is by using NURBS-curves. An object with smooth surfaces can be represented with NURBS-curves instead of geometry. "…the smooth surfaces of artificially generated 3D models, that are most often represented as parametric surfaces, of which Non-Uniform Rational B-Spline (NURBS) is a popular form" [14]. This makes the object maintain their detail no matter how close or how far the viewer is in relation to it. An example of using NURBS-curves to represent an object is shown in Figure 12 below.

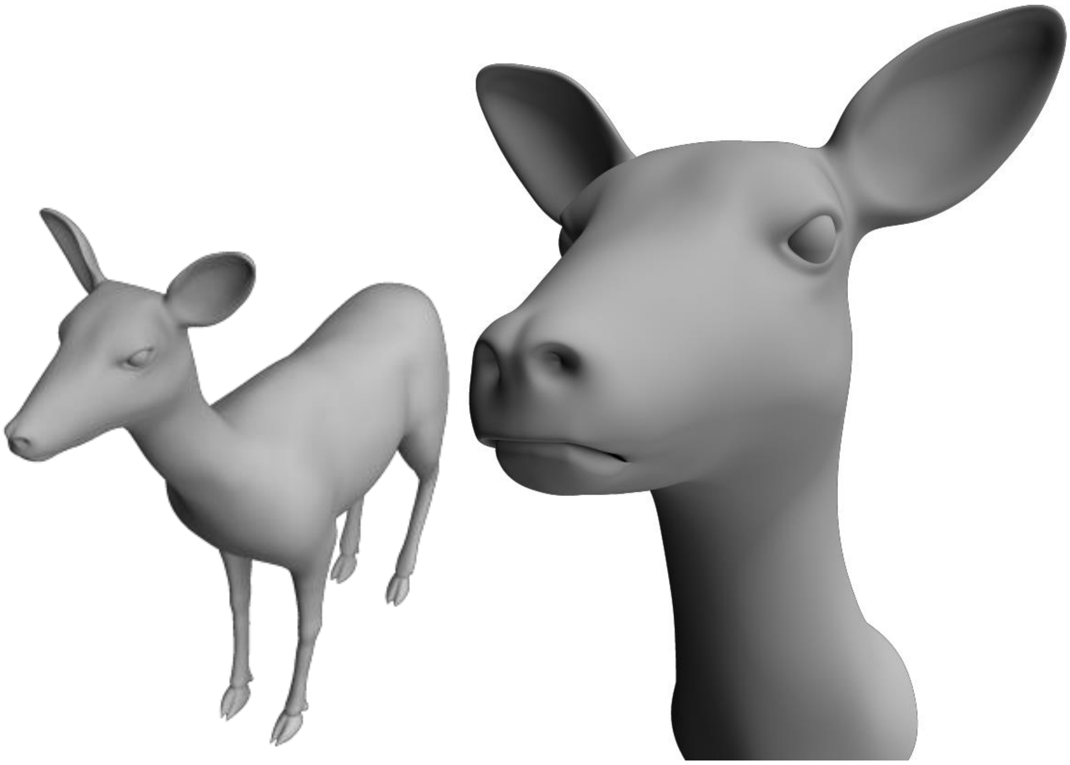


Figure 12. *An object represented using NURBS-Curves. To the left: An overall view of the object. To the Right: A close view of the object. (Illustration made by Per Axmark.)*

NURBS-Curves offer a more precise representation for smooth objects than other methods and offer great scaling possibilities which can suitably be used in tessellation. Curves are however more complex and not hardware supported which makes it less ideal for real-time uses, particularly in gaming. NURBS-curves offer advantages which make them attractive for design applications such as computer-aided design tools [15].

1.2 Purpose and goal

The purpose of this study is to explore two different ways of rendering grass represented with geometry. The implementation of volume-based, image-based and NURBS-curves rendering methods are therefore not addressed in this study.

The goal is to efficiently generate and render geometry-based grass in high density in relation to the camera's view point and direction in real-time.

1.3 Research questions

RQ1: What are the performance aspects of rendering geometry-based grass blades in a virtual 3D-environment in real-time?

RQ2: What are the benefits and drawbacks of using the data of the terrain to generate the grass blades?

RQ3: What is the perceived quality of the visual result?

1.3.1 Risks

The timeframe might not be sufficient enough to develop and implement two different algorithms in order to get a broader testing range.

Unforeseen problems might occur with one or both of the algorithms, which prevents its completion, which in turn nullifies testing.

Data gathered and assessed might not be enough to a conclusive answer the research questions.

1.4 Methodology

A quantitative method will mainly be used to analyze and evaluate the performance of each algorithm in order to answer *RQ1* and *RQ2*. Among the generated quantitative data is the rendered image of the scene. The evaluation of this image will be subjected to a qualitative method as the perceived quality of it can only be defined by a group of people. This qualitative data will be used to answer *RQ3*. The qualitative data generated will then be used together with the quantitative data to come to conclusions about the algorithms.

1.5 Bachelor thesis structure

The following parts of this thesis are structured as follows: The next chapter presents the implementation, data generation, data analysis and the interpretation of the data. Chapter 3 covers the results of the analysis of the generated data. The chapter that follows discusses the results and put them in context with the research questions and results from related work. Chapter 5 states the conclusions made from the discussion and answers the research questions. Finally, chapter 6 ends with proposed future work.

2 Experiment design

The experiment design has the purpose to find out how to measure the stated methodology. Figure 13 shows an illustration of the process used.

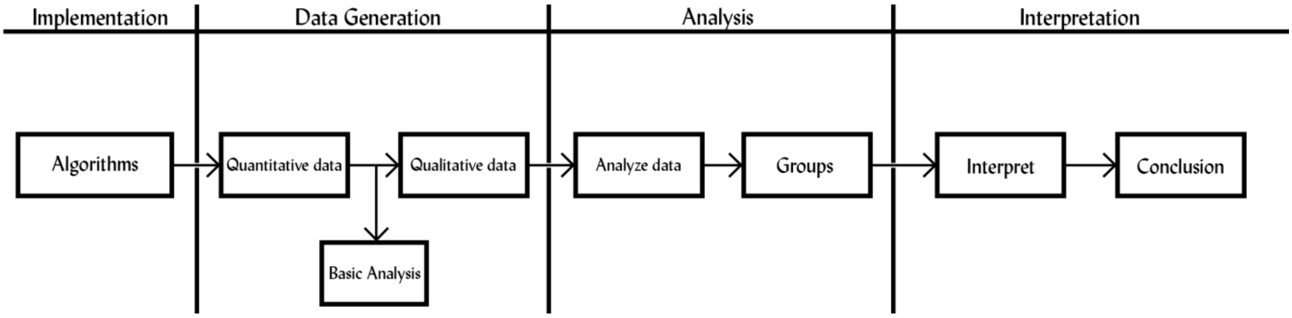


Figure 13. *Overview of the process. (Illustration made by Per Axmark.)*

First step is to design and implement two different algorithms. But before they can be implemented, a framework is needed. This framework will use the Direct3D 11 and Win32 application programming interfaces (API) developed by Microsoft.

As soon as an algorithm is implemented, quantitative data will be generated. This data will consist of the *quantitative* variables and other quantitative data. More on this in the chapter 2.2.1 Quantitative data generation. When all algorithms have been implemented, quantitative data will be generated using the same conditions. These conditions consist of the following quantitative *independent variables*[10]: the terrain and the number of grass blades generated based on the data of the terrain, the LOD-level values, the screen resolution, the lighting algorithm used and the same camera position and view frustum. Using the same camera position and view frustum will make it easier to spot differences in the scene when generating qualitative data. It will also make sure this data is not invalid when comparing images. These are the independent variables because they do not rely on other variables and are therefore only changed manually. Note that the LOD-level values represent distances. The quantitative data will be generated at different resolutions and LOD-level values to spot different bottlenecks.

Since the quantitative data produces visual results, namely the final image of the scene, qualitative data needs to be considered. How this data was generated is presented in 2.2.2 Qualitative Data Generation.

As recently mentioned, the independent variables are the number of grass blades to be generated, the terrain data, the LOD-level values, the lighting algorithm, screen resolution and finally the camera position and view frustum. These are used by the quantitative *dependent variables*[11]. The terrain data together with the number of grass blades to generate are used to create the grass blade vertices. These vertices contain the *root*-position, height, color and direction of the grass blades. The camera position and view frustum are used to calculate the amount of grass blades that may contribute to the scene of each frame. The grass vertices are then sent to the graphics card where the LOD-level values are used by the geometry-shader to determine the final number of grass blades to render as well as their geometric complexity. After the grass blades have been created in the geometry shader, the lighting algorithm is applied to them in the pixel shader and the final image of the scene is then rendered. This marks the end of the algorithms' execution. The number of frames the algorithms can render in a second is measured and used to calculate the number of milliseconds it took to render one frame. Rendered images are used to generate the qualitative data

[10] Variables that only can be changed manually by the user, i.e. a variable the user have direct control over.
[11] Variables that rely on other variables (independent or dependent) where the user only has indirect control.

and are therefore a *qualitative independent* variable. Note that if one ignores the fact the quantitative and qualitative aspects, the rendered images are a dependent variable. But for this study, the variables are split into these aspects, and the images will be considered as a qualitative independent variable. The qualitative independent variables are the questions asked about these images. For a complete view of the variables hierarchy, see Figure 14 below.

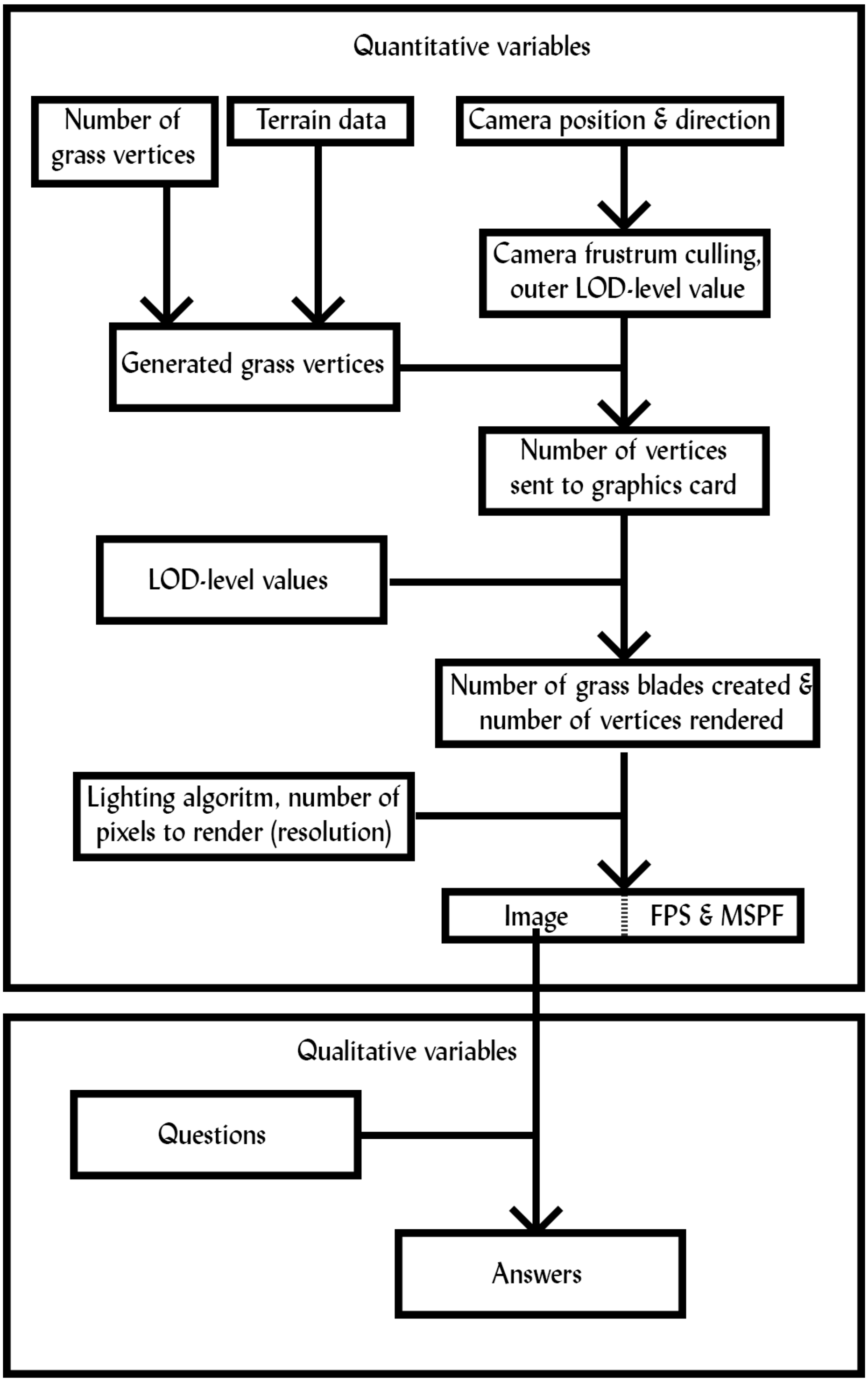


Figure 14. *An illustration of the hierarchy of the quantitative and qualitative independent and independent variables. (Illustration made by Per Axmark.)*

While this data is produced, some basic analysis of the quantitative data can be done. For instance, the performance of each algorithm in milliseconds and total frame rate can be compared with each other. As soon as both data types have been generated, further analysis can be done where the different data is put in context with each other.

When the analysis is complete, it's time to interpret and evaluate the generated data and come to conclusions about it. How this was done is presented in 2.4 Interpretation.

To improve the quality of the research results, the data generation, analysis and interpretation will be done twice for the qualitative data. First set of results will be of a smaller scale and is intended to give an early feedback on the algorithms and give clues on how they can be improved in terms of visual performance.

2.1 Implementation

2.1.1 Terrain

Rendering grass is not complete if not put in context in a natural scenario. Grass grows on non-uniform surfaces such as hills and cliffs. The terrain therefore plays an important role in portraying the grass in its natural environment. Even for flat surfaces, such as a lawn, a terrain texture is still needed to be rendered beneath the grass unless it completely covers the surface or else the grass will appear to be floating in space.

The attributes of the vertices of the terrain are the position in 3D, texture coordinates and normal. These three attributes sum up to 8 floats which are aligned over two float4 structures to save bandwidth when sending them to the GPU. The x and z-position and texture coordinates of the vertices are generated in a simple grid pattern. The y-position of the vertices are generated using 2D simplex noise developed by Ken Perlin [16]. Code taken from Eliot Eshelman[12], which is based on the work of Stefan Gustavson [17]. Figure 15 shows noise generated and used by a piece of terrain and Figure 16 shows noise applied to and textured on the terrain.

[12] http://www.6by9.net/simplex-noise-for-c-and-python/. Accessed 07-May-2013.

Figure 15. *Noise generated with Perlin's simplex noise. White represents a high value while black represents a low value.*

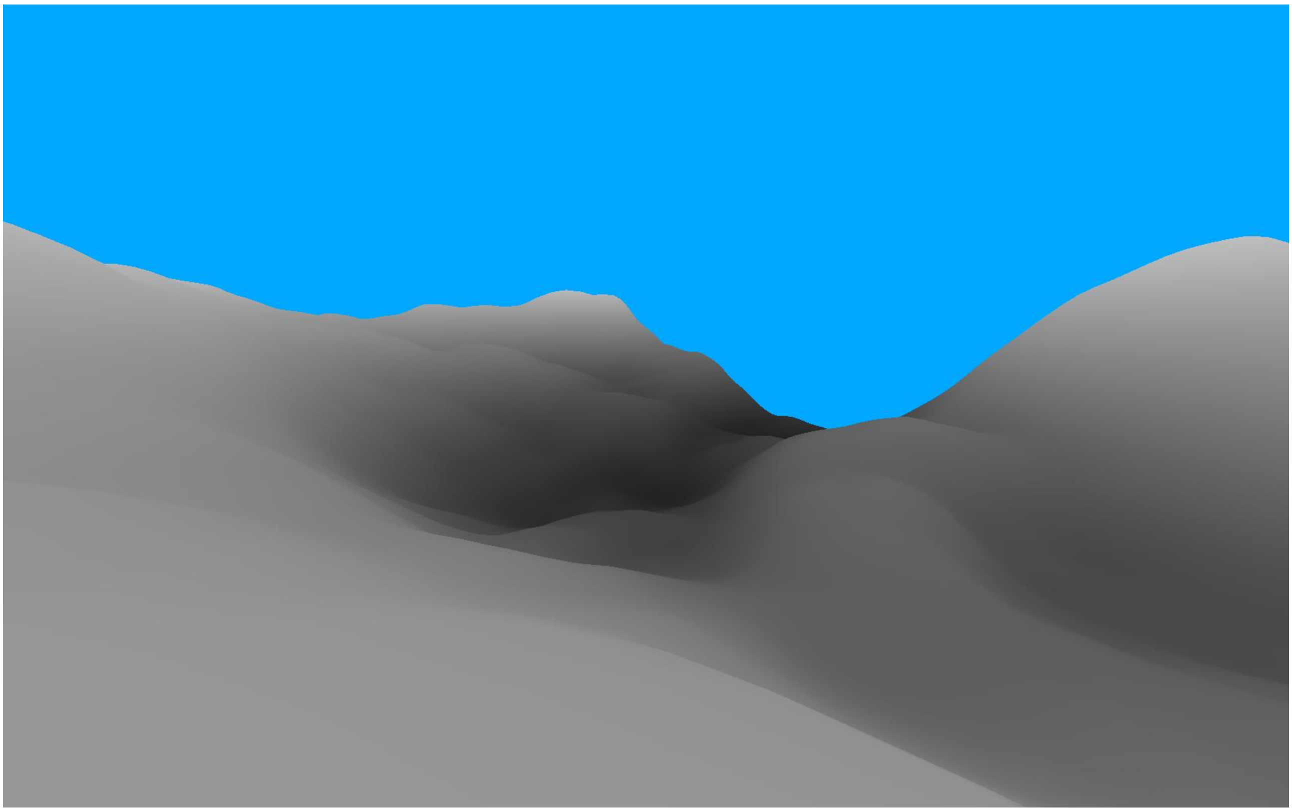


Figure 16. *Terrain applied and textured with noise; low height values result in black, high height values result in white.*

The terrain uses a grass-texture which is shown in Figure 17 below, and Figure 18 shows how the terrain looks when rendered with this texture.

Figure 17. *The terrain's grass*-*texture. (Image courtesy of Björn Jeffsell.)*

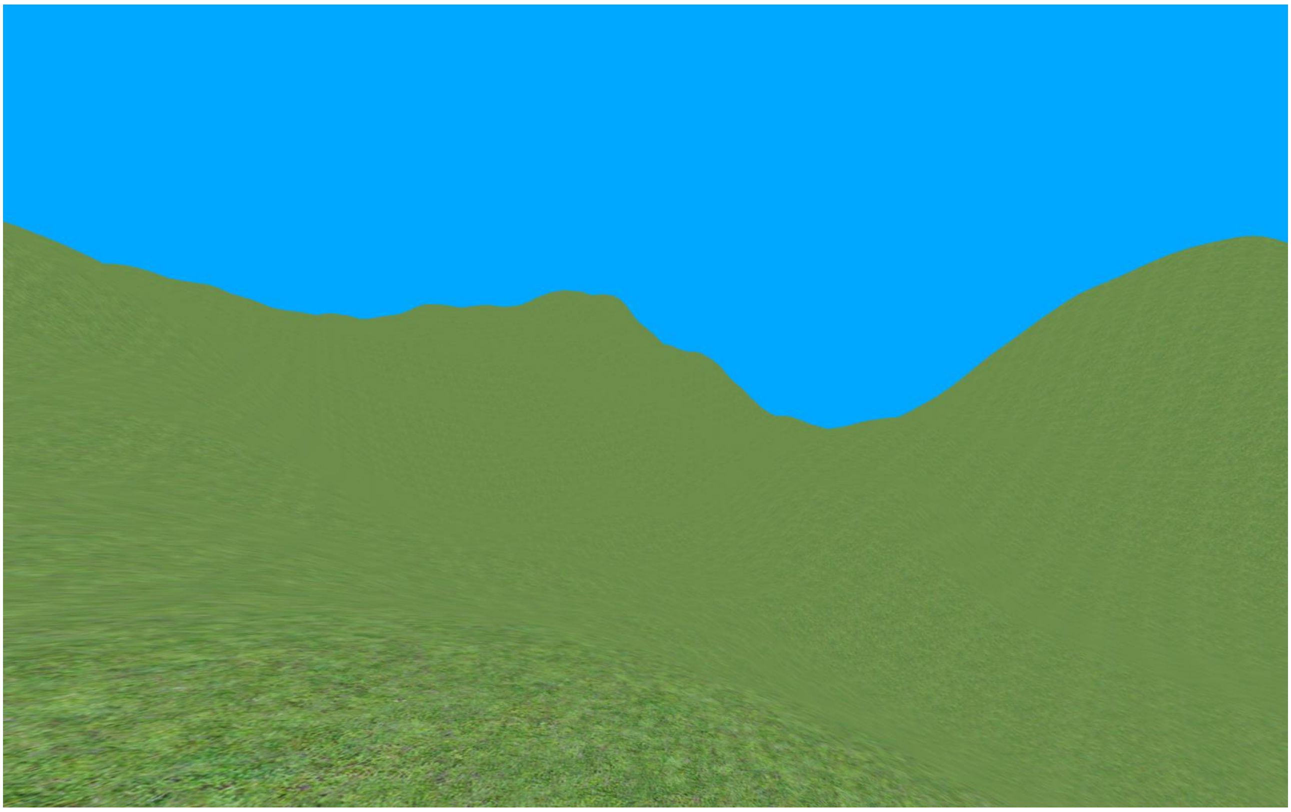


Figure 18. *Terrain textured with a grass-texture.*

2.1.2 The Algorithms

The purpose of using different algorithms is to make a comparison of them in order to find and better understand their advantages and disadvantages. The first algorithm, called *vBuffer*, as its name suggests, uses a traditional vertex buffers to store and render the grass blades. The second algorithm, called *texArray*, is an algorithm that uses texture arrays instead of vertex buffers to render the grass blades. The design of the algorithms identified the following shared *implementation* variables (not to be confused with the quantitative variables):

- The terrain.
- The culling algorithm.
- The LOD-system.
- Shape of the grass blade.
- Randomized positioning of the grass blades.
- Randomized height of the grass blades.
- Colorization of the grass blades.
- Randomized orientation/rotation of the grass blades.
- Lighting of the grass blades.

The terrain will provide the y-position for the root position of the grass blades. The culling algorithm will increase the number of grass blades that can be rendered without reducing the frame rate. The LOD-system will improve the performance further without significantly reducing the visual quality of the grass. The geometry of the grass blades are created with different detail for each LOD-level. The shape of the grass blade is created from points generated with a particle system. These points are then sent to the geometry shader through a constant buffer and are then used to make up the shape of the grass blades. Since a geometry shader is used for this, the algorithms require a shader model of 4.0[13] or higher. The grass blades are neither front nor back face culled so that they can be seen from every angle except when the direction of the grass is perpendicular to the viewpoint. A generated grass vertex contains the 3D-position, height, color and direction attributes. The first and highest LOD-level uses a grass blade shape with 7 points created from 4 points generated by a particle trajectory. These points are defined in a 3 dimensional space. Two variables are used to determine the width of the shape. First variable is the base (root) width of the blade and the other is the top width of the grass blade. The width between any vertices created between the root and top is a linear interpolation between the variables. Note that the top and tip is not the same thing. Figure 19 illustrates how a grass blade shape is done.

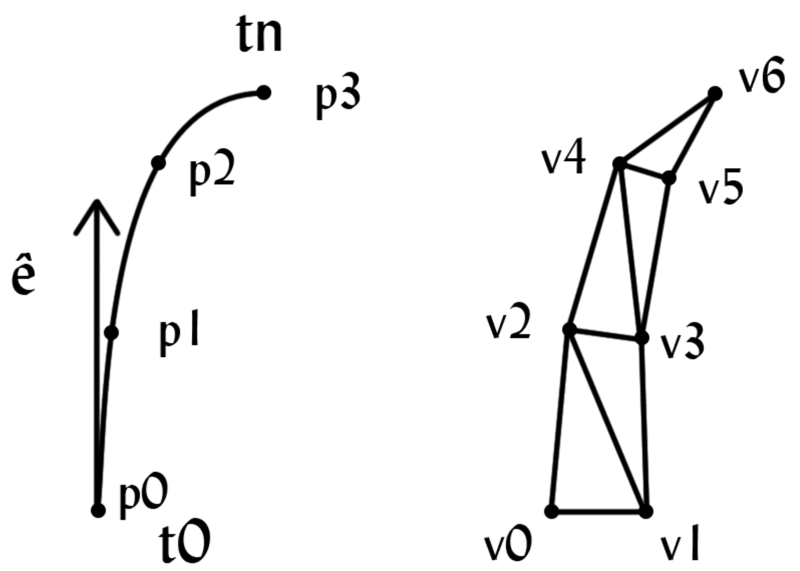


Figure 19. *To the left: A trajectory of a particle, as seen in Figure 3. pX is a point generated along this trajectory with a defined interval. To the right: A geometric representation of a grass blade using points from the trajectory of the particle in the left side of the image. vX is a grass blade point created using the points generated from the particle trajectory. V0, v2, v4 and v6 correspond to p0, p1, p2 and p3. The distance between v0 and v1 is the root width. The distance between v4 and v5 is the top width. (Illustration made by Per Axmark.)*

Each consecutive LOD-level use one less point, e.g. 5 grass blade points created from 3 particle points and 3 grass blade points created from 2 particle points, respectively. These grass blade points are put in a constant buffer which the geometry shader has access to and use these points when creating the individual grass blades. Figure 20 shows the shape and geometric complexity of the different grass blades used for each LOD-level.

[13] Available with NVidia GeForce 8 Series and Radeon HD 2000 Series released in 2006 or newer. http://msdn.microsoft.com/en-us/library/bb509657(v=vs.85).aspx. [Accessed: 16-May-2013].

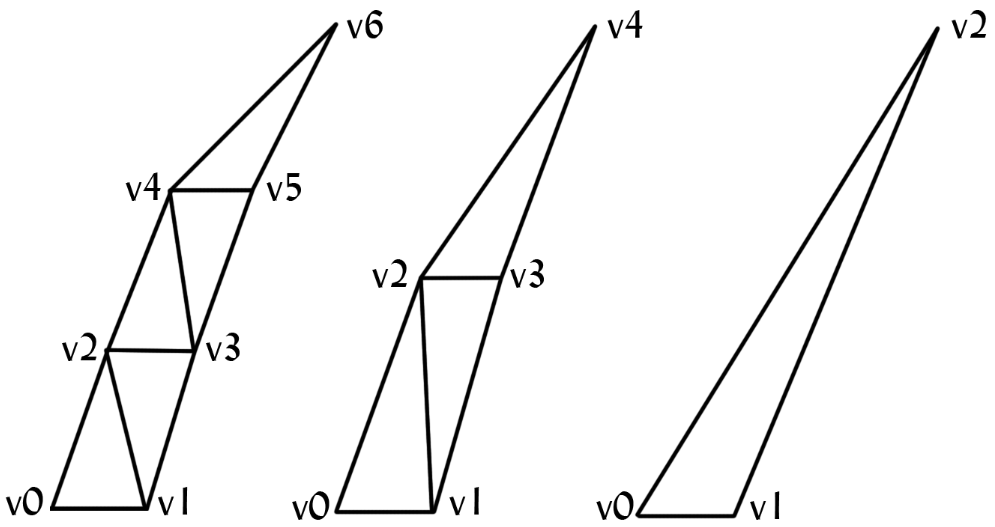


Figure 20. *Representation of a grass blade with varying geometric complexity. To the left: 7 points. Middle: 5 points. To the right: 3 points. (Illustration made by Per Axmark.)*

The randomization of the position, colorization, orientation and rotation of the grass blades is important to create a non-uniform pattern to give a realistic look as grass is not uniformly distributed, grown and colored in real life. The details of how this is done are presented in 2.1.5 Randomization. To simulate lighting, Phong's reflection model [18] is used by both of the algorithms.

The base color of the individual grass blades is sampled from the texture of the terrain by converting the x and z-position of the grass vertices to texture coordinates. The result is shown in Figure 21 below.

Figure 21. *The texture of the terrain below the grass is used to color the individual grass blades.*

The immediate drawback of doing this is that the grass blades are colored in other shades than green. This is because the terrain's grass texture does not only contain green texels. This problem can be avoided by sampling neighboring texels and making a comparison to determine if a grass blade should be created and if so, what color it should have. Figure 22 below shows an illustration of the problem and Figure 23 shows an explanation of why this happened.

Figure 22. *An illustration of the drawback of using the terrain texture to color the grass blades. Some of the grass blades in the middle of the screen within the red circle are colored with a shade of purple.*

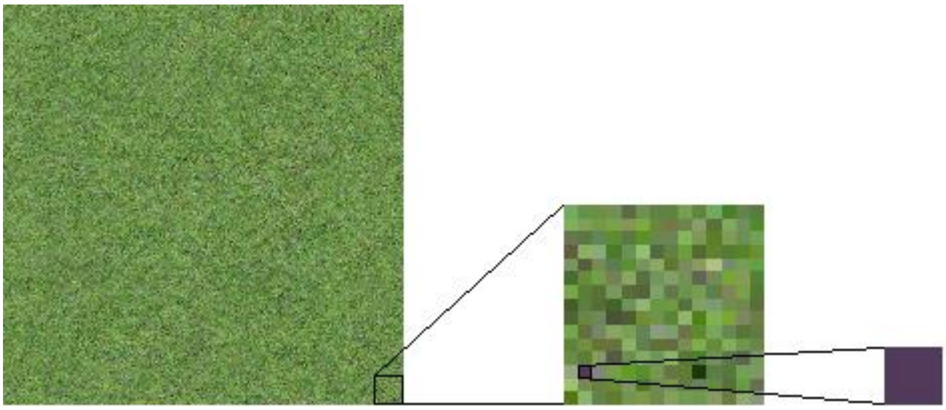


Figure 23. *The terrain's grass texture. First after zooming in can you see that not all the texels are a shade of green.*

The surface normals of the terrain can also be made use of unless the terrain surface is flat and the normal of the surface is known. For instance, if the surface is too steep, grass cannot grow on it. Figure 24 below shows a comparison of using and not using the terrain's surface normals to orient the grass blades.

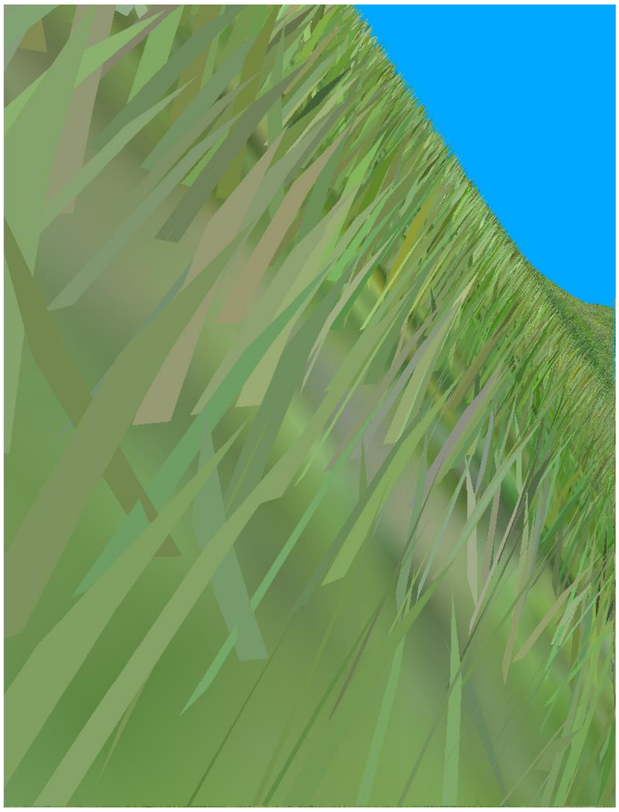
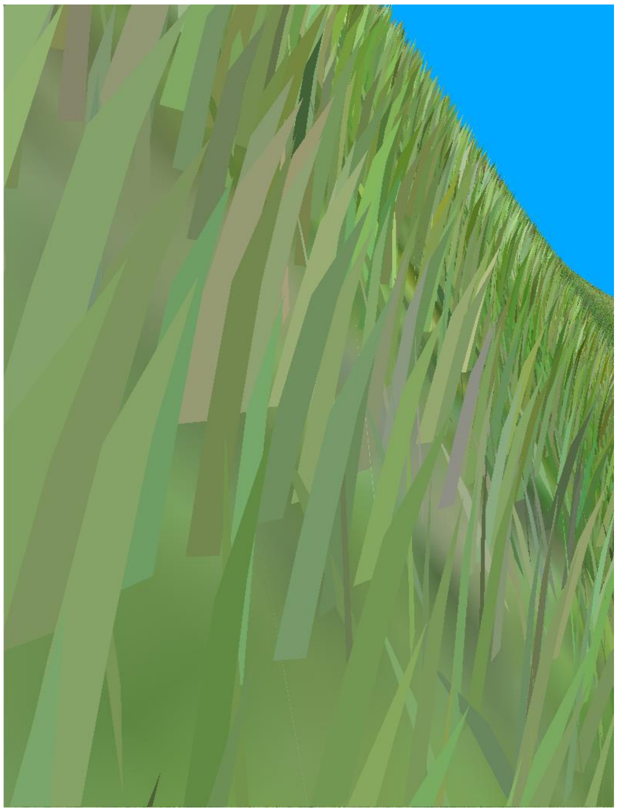



Figure 24. *A comparison of using (image to the left) and not using (image to the right) terrain surface normals to orient grass blades.*

Due to the amount of work involved to handle all the special cases of using all the terrain data to generate the grass vertices, this study only make use of the elevation of the terrain when generating the grass vertices. Instead, this study focuses on identifying the advantages and disadvantages of using the terrain data to generate the grass vertices in order to answer research question *RQ2*.

2.1.2.1 Algorithm 1 (*vBuffer*)

The generated grass vertices are stored in vertex buffers which are sent to the graphics card when needed for rendering. The vertex shader used by this algorithm is extremely simple and just passes the data it receives from the vertex buffer to the geometry shader. The geometry shader input data structure and vertex shader looks like this:

```
struct GSInput // Geometry shader input data structure.
{
	float4 posWHeight	: POSW_HEIGHT;		// XYZ: position, W: height.
	float3 color		: COLOR;		// XYZ: RGB-color.
	float3 dir		: DIR;			// XYZ: Direction of object.
};

GSInput VSScene(GSInput input) // Vertex shader.
{
	return input;
}
```

The geometry shader's output is the vertices that make up the shape of the grass blades after they have been oriented using the input and camera data. The number of vertices that are created depends on the LOD-level the grass vertices reside in. The vertex count also depends on the number of additional grass blades that are generated from the same grass vertex sent from the vertex shader. These additional blades are created using the data of the original grass blade created from the sent vertex together with a position offset. This offset is based on the density of the grass to ensure that the density remains uniform so that a pattern does not emerge. The only thing the pixel shader is in charge of is the lighting of the grass blades which is explained in 2.1.6 Lighting.

The use of preprocessor macros allows flexible creation of multiple shaders with different functionality without the need to write completely new shader code. This has led to faster testing of different functionality such as seeing the differences between spawning one and multiple grass blades per grass vertex in run time. This has been especially helpful for the quantitative data generation (see 2.2.1 Quantitative data generation) since much of the data could be gathered without the need to change variables and recompile the program.

2.1.2.2 Algorithm 2 (*texArray*)

The difference between algorithm one and two is that this algorithm uses one-dimensional texture arrays to store and render the grass blades instead of vertex buffers as algorithm 1 does. Since a vertex shader is always needed for rendering, and the algorithm does not use a vertex buffer, the vertex shader now looks like this:

```
uint VS(uint vertexID : SV_VertexID) : VERTEX_ID
{
	return vertexID;
}
```

Where `vertexID` is a number which increments by one for each draw call. This number is used as an index to sample the grass vertices stored in the texture arrays.

The benefit of using textures instead of buffers is that additional grass blades can be generated per vertex as the geometry shader has access to enough data to be able to accurately create them. This is done by sampling the nearby vertices in the textures of the vertex being evaluated. These samples are then interpolated based on where the additional grass blades are to be created between the samples. This is more efficient as sending data from the central processing unit (CPU) to the GPU is relatively expensive [19].

2.1.3 Culling

Sending millions of vertices to the graphics card that do not contribute to the final image is useless. A camera view frustum culling [10, pp. 771–778] technique is therefore implemented to cull away vertices to improve performance. Using such a technique provide better performance-scaling when adding more pieces of terrain and grass blades in the world. "One of the most time consuming tasks for a modern graphics processor is to shuffle data between the CPU and the GPU…"[14] [19]. This makes it extra important to cull away vertices that do not contribute to the scene as a grass field can contain millions of grass blades.

[14] Translated by Markus Tillman.

Each piece of terrain with associated grass is split into smaller pieces. These pieces are then checked with the camera frustum to determine if they are inside or intersecting it using axis-aligned bounding box (AABB) versus frustum culling. Note that these AABB's are reused for the grass and are expanded by the maximum height of the grass blades to include all of them. Culled pieces are excluded from rendering as the culling test concluded that they will not contribute to the scene in any way. Figure 25 illustrates AABB vs frustum culling.

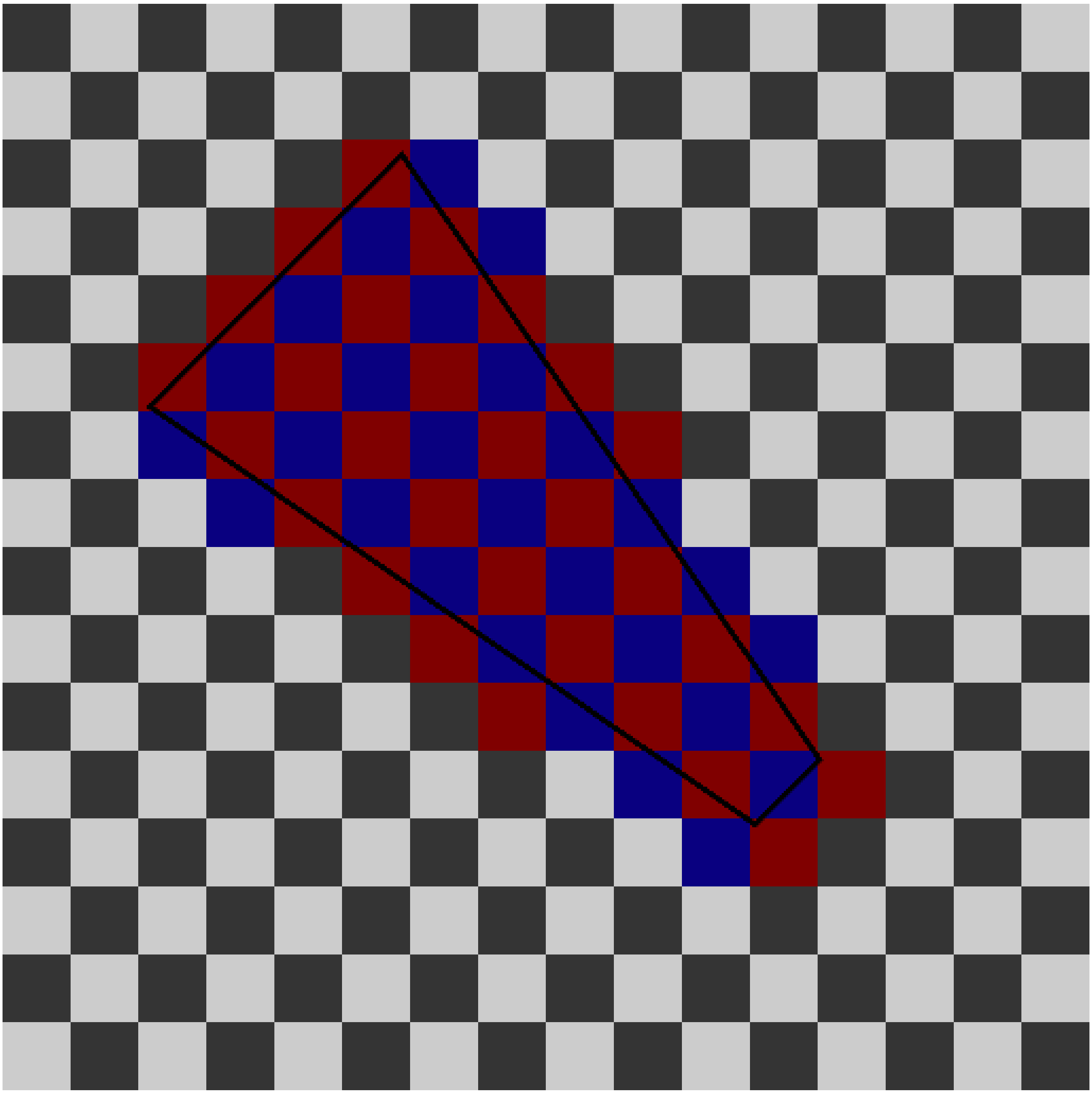


Figure 25. *An illustration of frustum culling. The red and blue squares are either inside or intersecting the frustum and therefore might contribute to the final image of the scene. (Illustration made by Per Axmark.)*

Mentioned earlier, each piece of terrain and its associated grass is split into smaller pieces. The terrain data of the pieces that were not frustum culled are then sent to the graphics card for rendering. The grass vertices in these pieces is then checked if it is within the outer LOD-level's value. If not, they are excluded from rendering as the outer LOD-level's value determines the range in which the grass blades will be rendered. As such, it's unnecessary to send the grass vertices that are outside this range to the GPU as they will not contribute to the scene. See Figure 26 for an illustration of this type of culling.

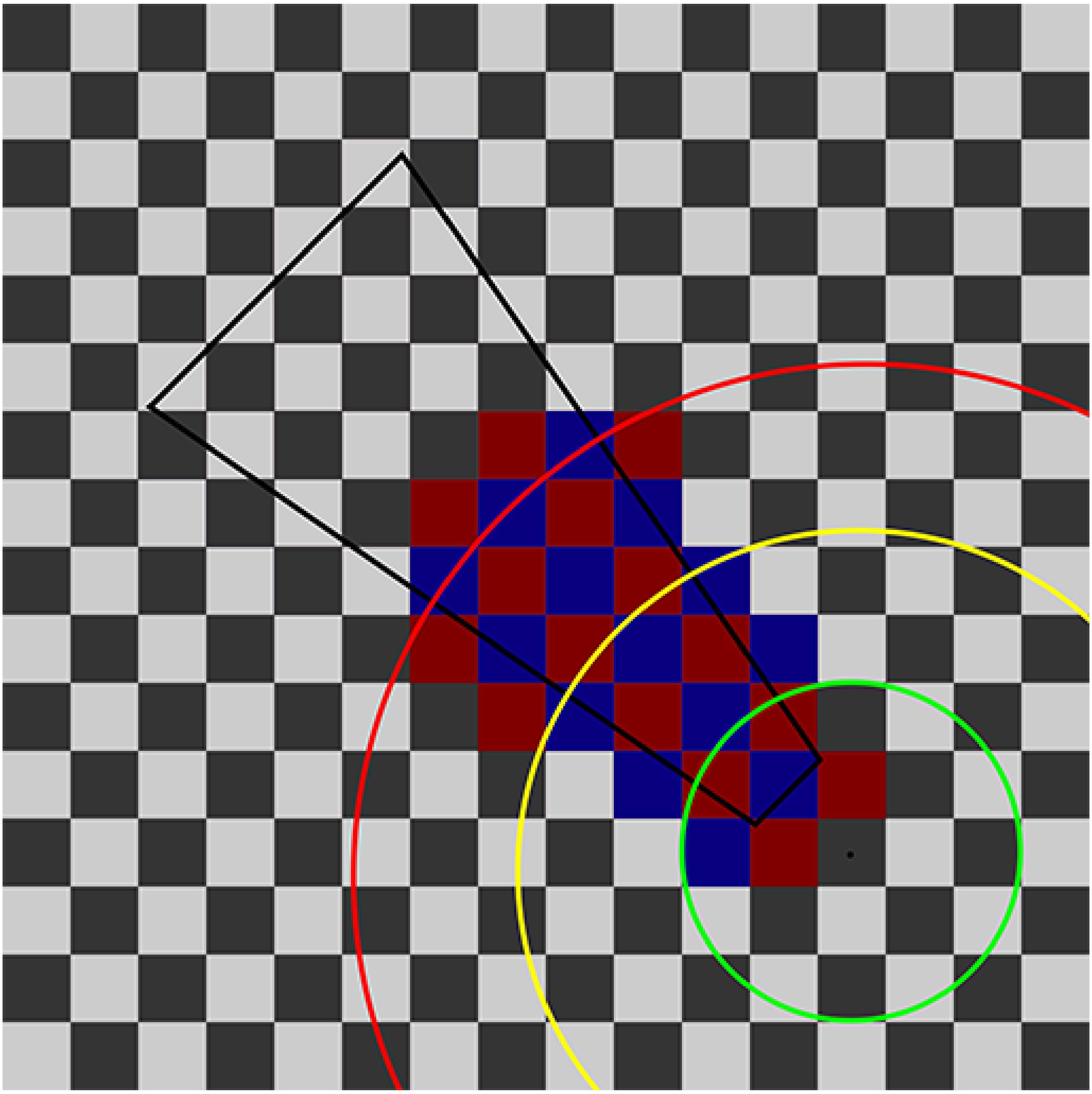


Figure 26. *An illustration of LOD-culling. First LOD-level is shown with a green circle, second level with a yellow circle and third with red circle. The red and blue squares are either inside or intersecting the outer (red) LOD-level's radius. (Illustration made by Per Axmark.)*

The grass vertices of the pieces that pass this culling test are then sent to the graphics card for rendering as it has been determined that they might contribute to the scene. As final culling, the grass blades furthest away are culled on the GPU to make a smooth border. More on this in the following chapter.

2.1.4 LOD

A simple LOD-system is used for the grass. It uses three levels and its calculations are done on the GPU in the geometry shader. The purpose of the LOD-system is to improve performance without reducing the perceived visual quality of the grass.

The variables and data associated with the LOD-system are the values for the three LOD-levels and a grass model with varying geometric complexity for each of them. The LOD-level values represent distances. The outer LOD-level's value is set to a range where the grass blades start to only contribute to one or no pixels as it is useless to render them at this point. This starts to happen when the outer LOD-level's value is around 32 or higher for this particular implementation.

As the geometry of the grass blades are created in the geometry shader, the LOD data is used to determine what LOD-level a grass blade reside in. This is done by calculating the distance between the grass vertices' position and the camera's position. This distance is then compared with the LOD-levels' radii to see which is greater. This is to determine the geometric detail the shape of the grass blade should have. As briefly mentioned in the previous chapter, if a grass vertex is outside of the outer LOD-level's radius, it is omitted from rendering. This is done to smooth out the edge of the grass field. The transition between the third and second LOD-levels is almost unnoticeable even when moving. However, the transition between the first and second LOD-levels is noticeable when moving as the shape of the grass blade of each LOD-level differ. The transition can be made less noticeable with better modeling of the grass blades. Figure 27 below shows the LOD-levels highlighted.

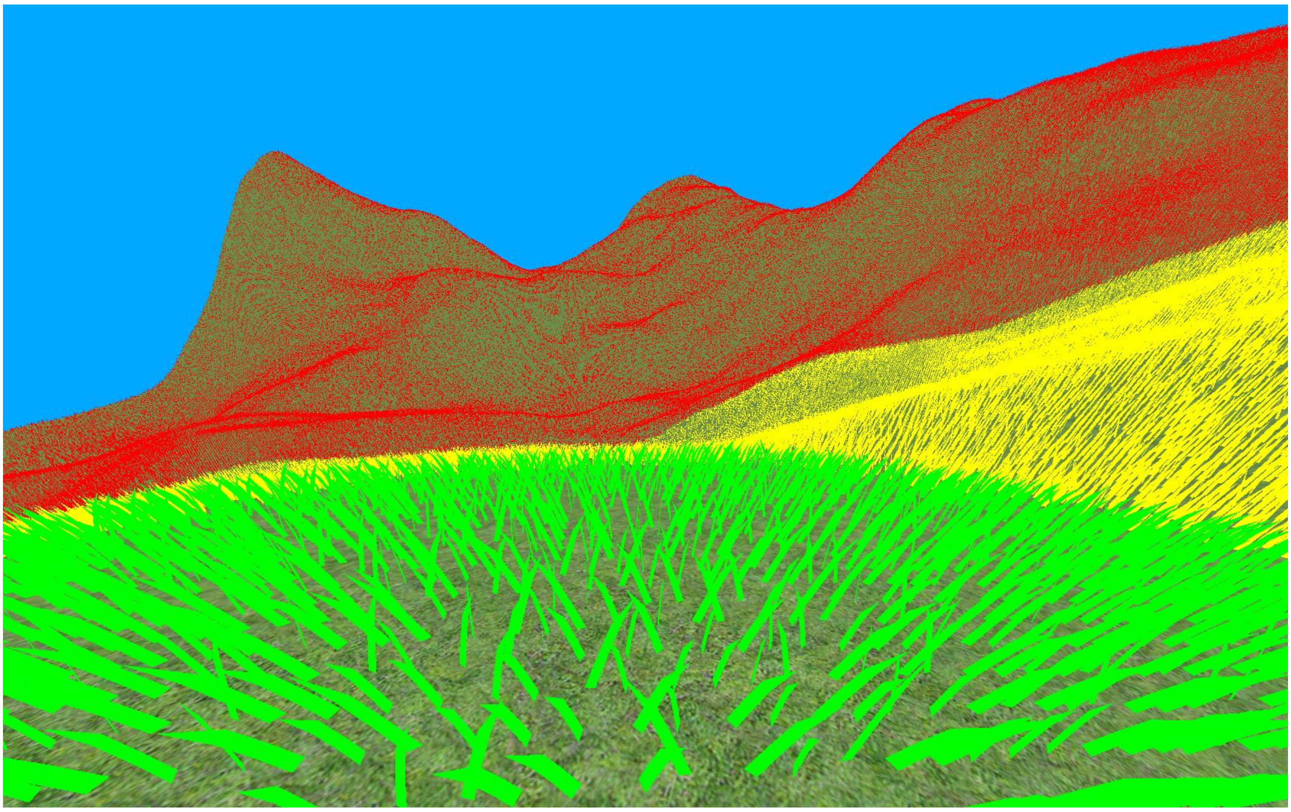


Figure 27. *LOD-levels of the grass blades highlighted. First (closest) LOD-level is green, second is yellow and third is red.*

2.1.5 Randomization

The most important part is to distribute the grass blades in a non-uniform manner and give them varying heights and color in order to give a more realistic impression. Also, grass never grows in one particular direction in real life, so the grass blades have to be randomly oriented as well.

Using raw noise generated from Perlin's Simplex noise [16] simply will not do as this generates a recurring pattern. Figure 28 and 29 show how the scene looks when the grass blades are generated with this raw noise.

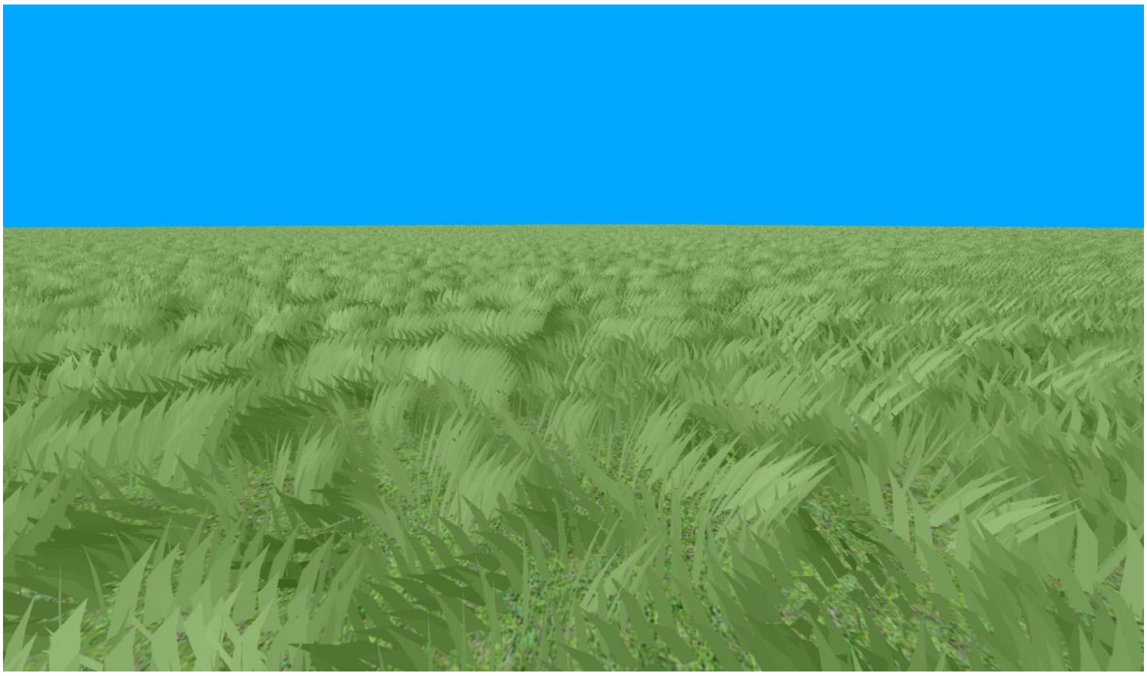


Figure 28. *A grass field generated with raw noise.*

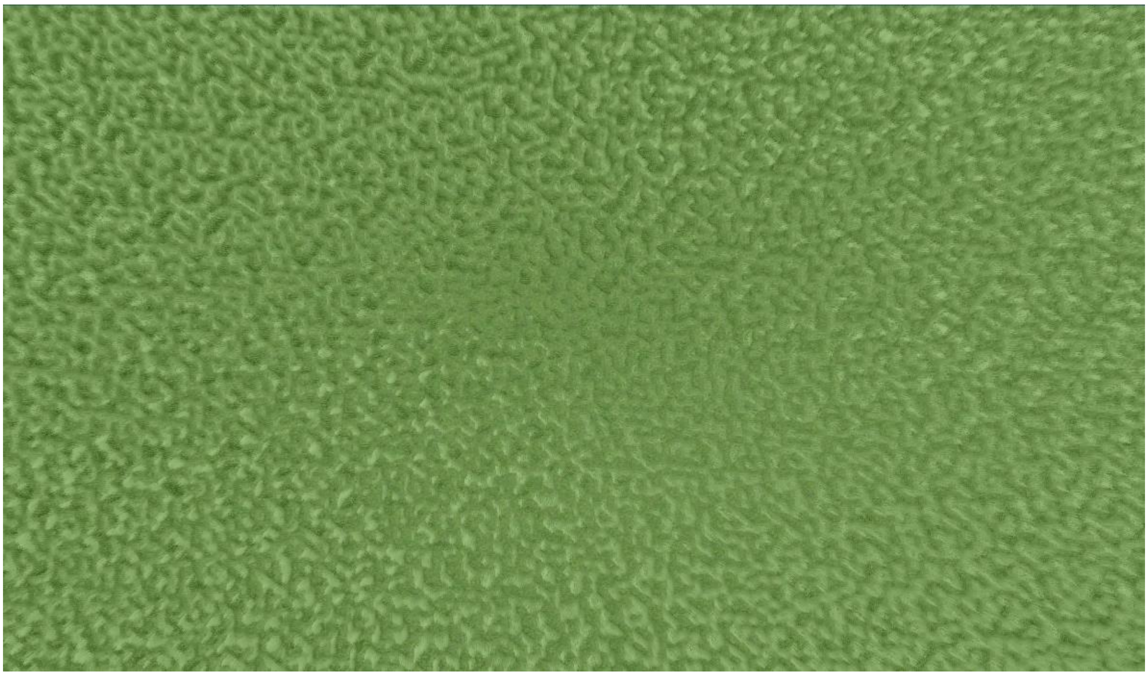


Figure 29. *Same field as in Figure 28 from a top-down view.*

To simulate differences between the individual grass blades, an “octaves noise function”[15] is used. Using an octaves noise function also reduces the recurring pattern significantly. However, this is still not good enough. To simulate non-uniformity caused by different kinds of soil, nutrients in the ground, accessibility to water and sunlight, additional octave noise functions are used. By blending the noise generated from these functions, we get an even better result as the recurring pattern is reduced further and we get a non-uniform pattern. The noise generated from *two* octaves noise functions are shown in Figure 30 and the blended result in Figure 31 and Figure 32. Note that these are not used in the final implementation.

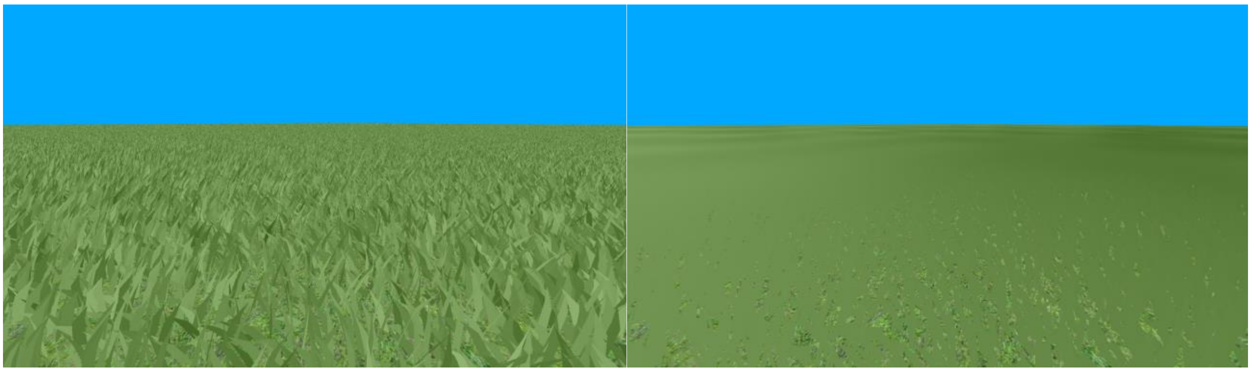


Figure 30. *Left side: Color of the grass generated using an octave noise function with high scale. Right side: Color of the grass generated using an octave noise function with a low scale.*

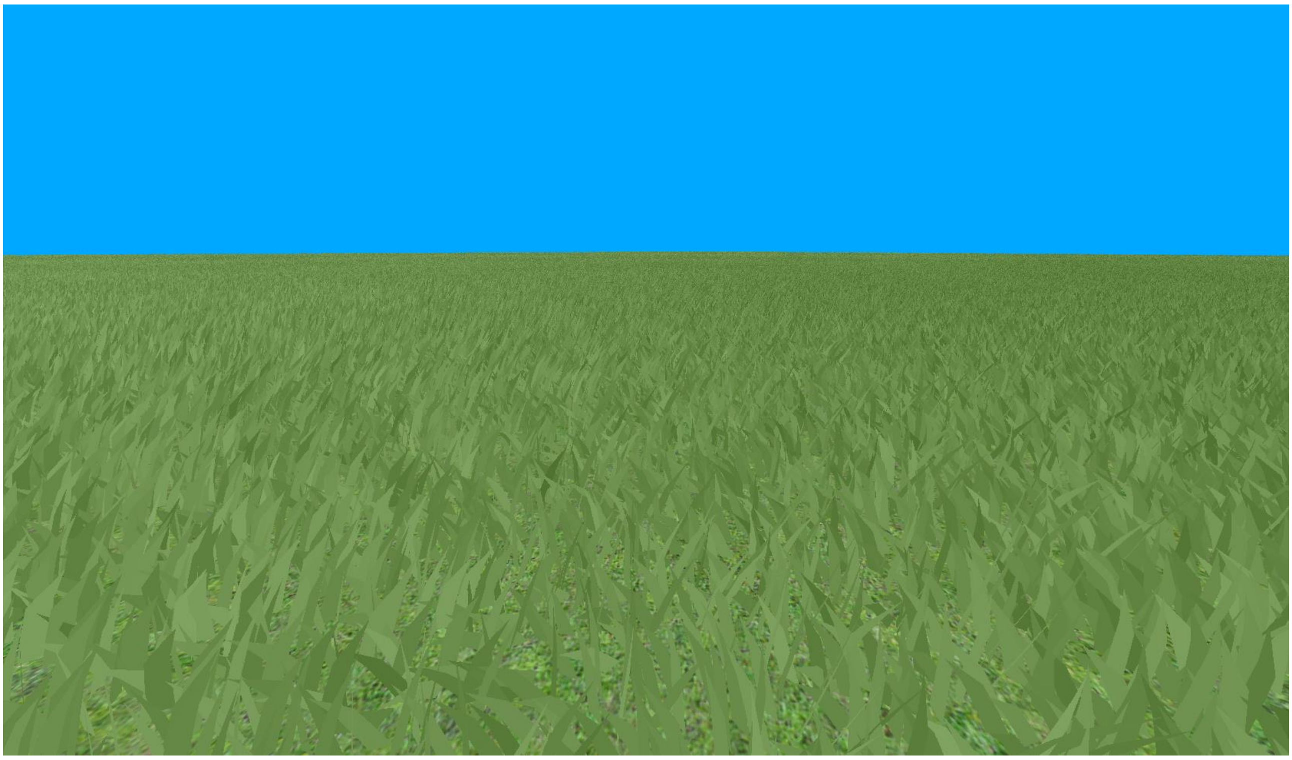


Figure 31. *The result of blending the result from the two noise functions shown in Figure 30.*

[15] A set of raw noise functions with different frequencies blended together for a smoothed result.

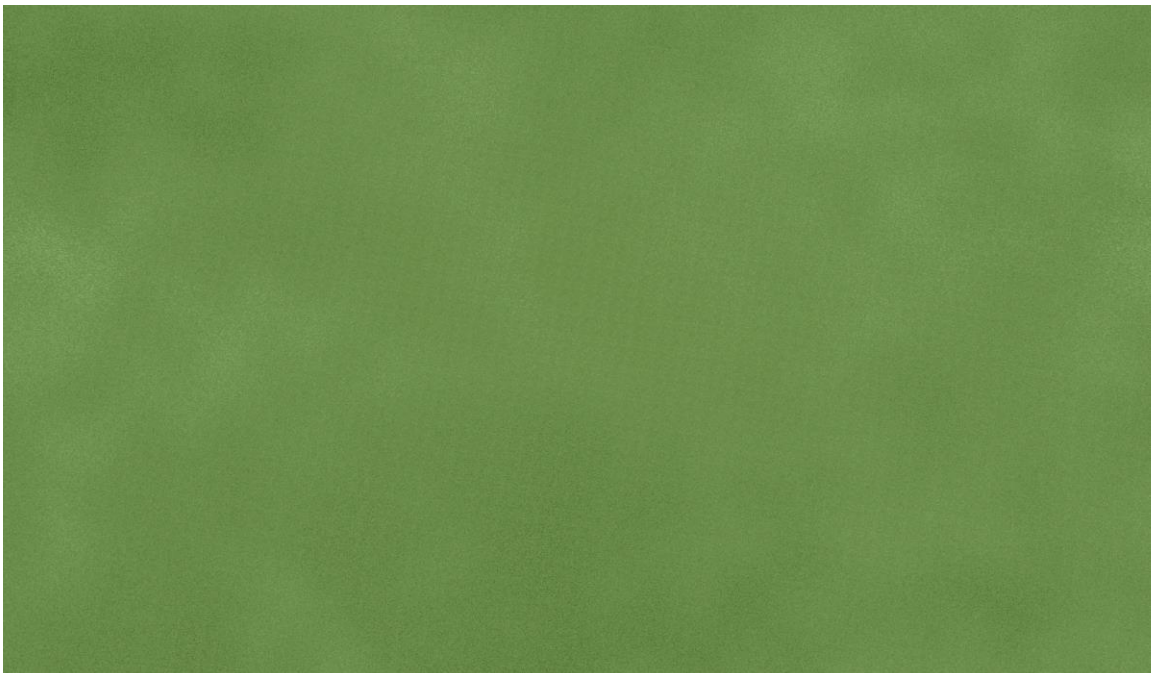


Figure 32. *Same scene as in Figure 31 seen from a top-down view.*

Simply using different shades of green is a bit dull, so each RGB[16]-channel of the color of the grass blades uses a blend of noise functions with different parameters. The result looks more appealing, shown in Figure 33 and 34.

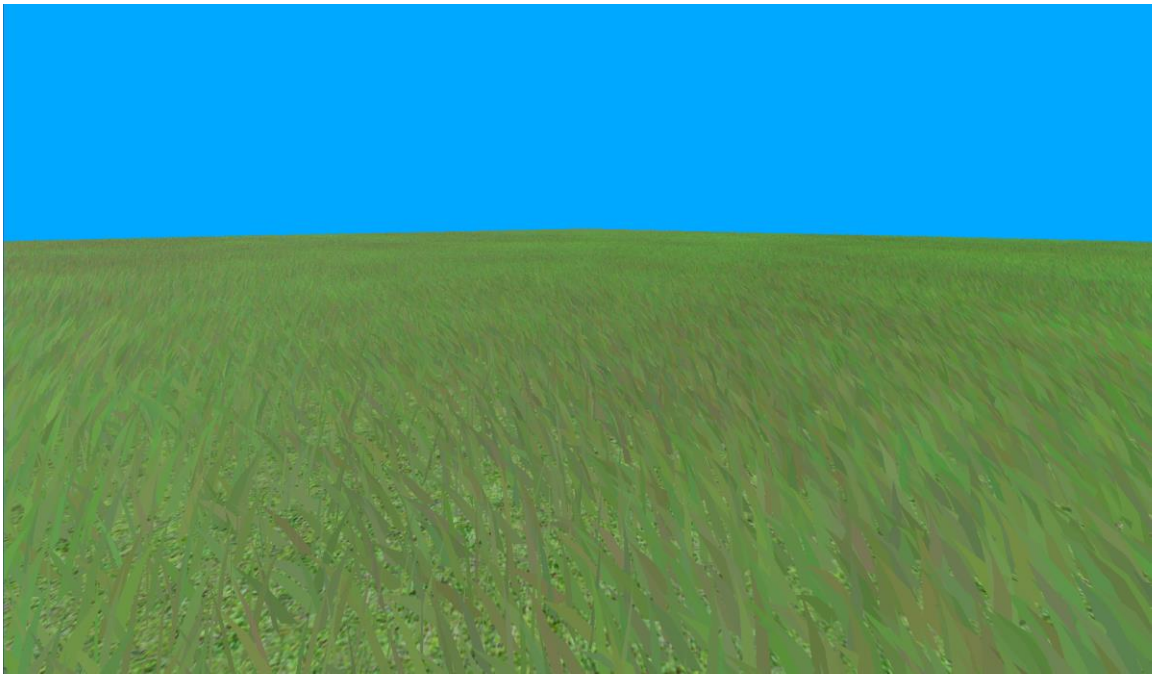


Figure 33. *The result of blending different noise functions for each color channels.*

[16] Red/green/blue.

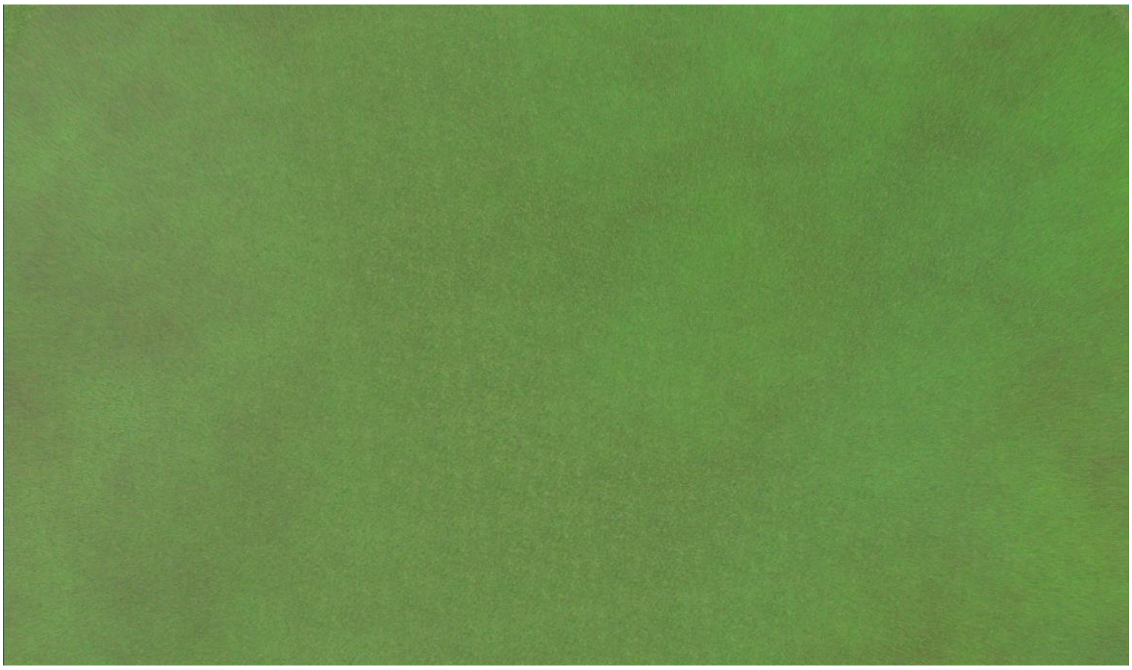


Figure 34. *Same scene as in Figure 33 seen from a top-down view.*

2.1.6 Lighting

The lighting of the grass blades is done in the pixel shader with Phong's reflection model [18]. The only modification made to the lighting algorithm is that the height of the grass blades and density of the grass field is used to approximate lack of ambient lighting caused by the density of grass blades. The ambient reduction is most intense at the root of the blades and decreases as it goes to the tip of the grass blade. Figure 35 below shows the lighting algorithm added to the scene in Figure 33.

Figure 35. *The scene as it looks when the lighting algorithm is added. The color of the grass blades becomes darker towards the base of the blades.*

2.2 Data generation

2.2.1 Quantitative data generation

Now that the implementation variables have been covered, the focus lies on the quantitative variables. The quantitative data consists of the following quantitative variables and other quantitative data:

- Settings
  - Camera view frustum.
  - Resolution (number of pixels rendered).
  - What was rendered (terrain, grass with algorithm 1 or 2).
  - The settings of the grass.
    - LOD-level values.
    - Number of grass blades sent to the graphics card.
- The number of samples taken.
- The average number of draw calls made per frame.
- The average number of vertices sent per frame.
- The time it took to gather the data.
- The average case of frames per second and milliseconds it took to render a frame.
- The worst case of frames per second and milliseconds it took to render a frame.
- The best case of frames per second and milliseconds it took to render a frame.

The data is generated along a pre-defined path taken with at least 10000 samples over a period of at least 40 seconds to give a guarantee of a minimum precision in the result and to ensure the comparison of the data that follows is valid. This is done using the same path and settings. This is also done separately for each algorithm so that the performance of each of them is measured. The purpose of generating this data is to use it to determine the performance of the algorithms and identify their advantages and disadvantages as well as their limits.

The following setup was used to generate the data:

- Operating system: Windows 7 professional.
- Motherboard: Asus P8Z77-V LX Z77.
- Main Memory (RAM): 8 gigabyte (2x4 gigabyte) 1866 Hz Corsair DDR3 CL9 Vengeance.
- CPU: Intel core i5 3570k 3.4 GHz.
- Video card: NVidia GeForce GTX 670.

2.2.2 Qualitative data generation

Mentioned earlier, the data generation, analysis and interpretation of the qualitative data will be done twice to get improved results. The qualitative data will be generated by using two questionnaires where feedback from people is collected. The first questionnaire is intended to give an early feedback on how the visual result of the algorithms can be improved and the second is intended to give the final qualitative results on the algorithms. The purpose of the generated qualitative data is to give an insight of the perceived visual quality of the grass.

The first question of the *first* questionnaire is a simple mathematical question and its purpose is to filter off unwanted automated response. It is followed by general questions about age, gender and gaming experience. The user then defines what they perceive as quality by choosing one image among three, see Figure 36 below. The question that comes next is a control to see if they have an eye for detail. The question asks if the user can see a recurring pattern in an image. Finally, questions are asked about the contents of an image of the scene of the, at the point the questionnaire was made, final version of the grass. For a complete view of the first questionnaire, see Appendix A1.

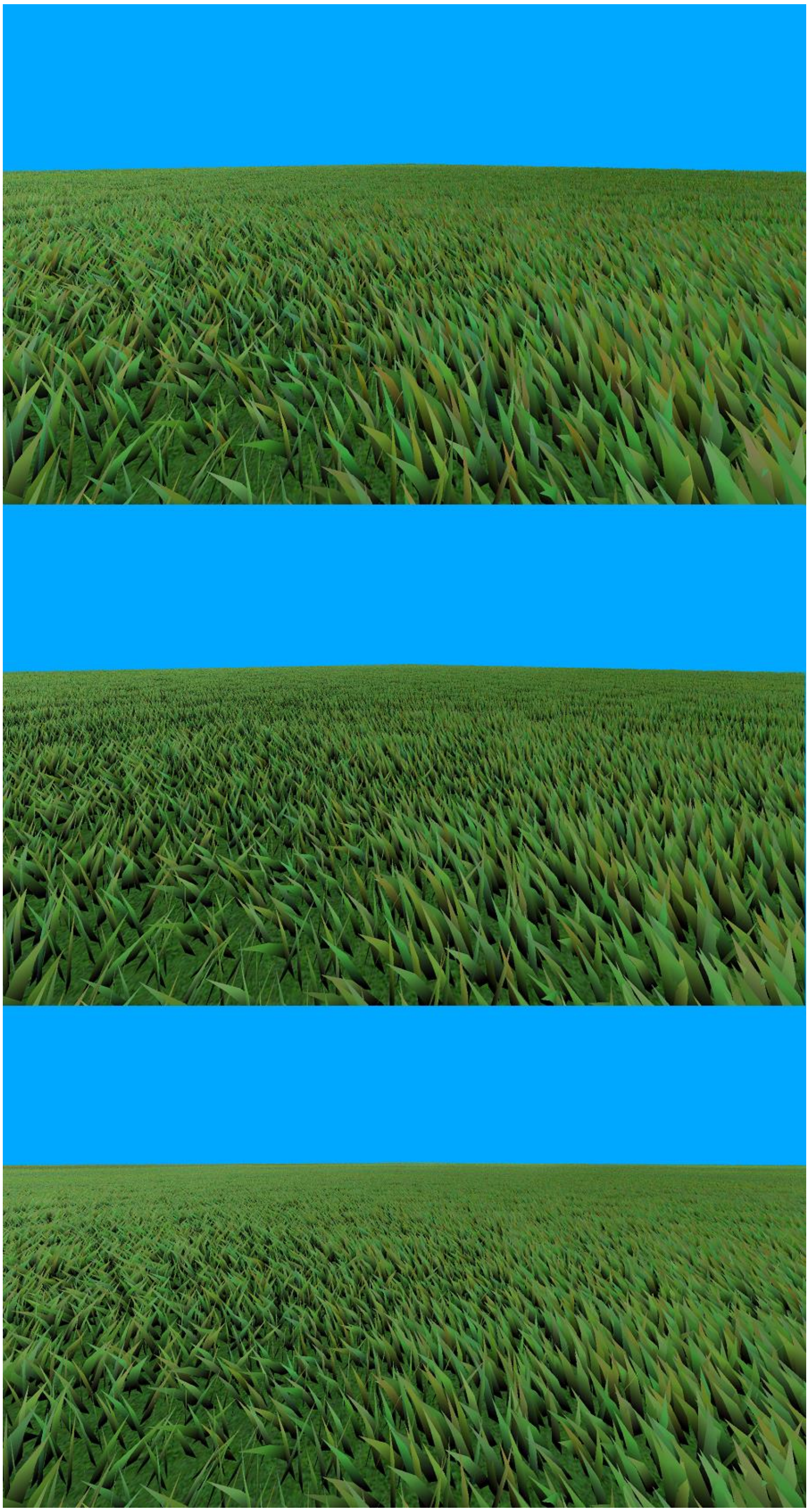


Figure 36. *First image (top) shows grass blades rendered with high LOD-level values and slightly more variation in color and grass height than the grass blades in the other two images. Second image (middle) shows grass rendered with low LOD-level values and therefore contains less geometric complexity. Third image (bottom) shows grass blades rendered with high LOD-level values with the same colors as the grass blades in the second image.*

Based on the answers from the first questionnaire, and some additional processing, a *second* questionnaire was made. The changes from the first questionnaire were that the question designed to filter off automated response as well as the questions about gaming experience was unnecessary, due to the fact that automated response can be filtered by using a control question and that neither game mechanics nor gameplay is implemented. Other modifications were made, where the most important ones were the reference for quality and pattern. These were added to provide a base to compare to when interpreting the answers of the questions concerning the quality and patterns of the grass. For a complete view of the questionnaire, see Appendix B1.

2.3 Analysis

2.3.1 Quantitative data analysis

The quantitative variables and other quantitative data were saved to a text file where one line in the file contains a value of a variable and an identifier (the variable name) to what the value represents. In other words, the quantitative data is already structured and calculated and is therefore already in a manageable form. This means that it can be interpreted right away and therefore does not need to be analyzed. The only exception is the number of vertices that were rendered. The camera view frustum data together with the values of the LOD-levels, the number of points of the shape of the grass blades of each LOD-level as well as the density and number of grass vertices sent to the graphics card can be used to calculate the exact number. Note that this is not the same number as the number of vertices that were sent to the graphics card. The number of vertices that were rendered is not calculated as it is not used in this study. Thus, the camera view frustum data is only there for future interest of knowing the exact number of vertices that were rendered. The quantitative data does not need to be processed any further than compiling it together into a graph to give a better overview of it.

2.3.2 Qualitative data analysis

In order to deal with the large variety of answers given in the questionnaire, this qualitative data needs to be sorted and grouped by similarities in the answers to be able to efficiently interpret it. The average of all the answers will be calculated and put into matrices to give a summarized overview. This will make the data much more manageable and make it much easier to interpret.

2.4 Interpretation

After the data has been analyzed and made manageable, it’s time to interpret it before it can reflect on the implementations and conclusions can be made. The quantitative data is pretty straightforward here; if all the variables are the same for both algorithms, but the performance is not, then the algorithm with the highest performance is the one with the highest FPS. Now, this does not necessarily reflect on the advantages and disadvantages of the algorithms. An understanding of the algorithms is needed to be able to see their benefits and drawbacks. For instance, as algorithm 2 uses textures instead of buffers to render the grass, it has the advantage of being able to accurately generate several grass blades per vertex.

The qualitative data is a bit more complex to interpret as there are more variables to consider. First off, the groups of similar answers, if any, need to be interpreted separately using the average of the answers in the groups. After this has been done, the groups can be interpreted as a whole, using the average of all the answers. For instance, if a group containing the answers from ages 15-25 have all answered yes on a specific question, and another group containing the answers from ages 60-80 have all answered no on this question, which group is “right”?

3 Results

3.1 Quantitative data results

Different quantitative variables were used in the following chapters. The only variable changed between the different results in 3.1.1 Resolution comparison was the resolution variable and these results were meant to gather data about the pixel shader. In the following chapter, 3.1.2 LOD-levels comparison, the results were generated using different values of the LOD-level variables. The other variables remained unchanged. This is followed by 3.1.3 Creating additional grass blades where the results were generated by creating a different number of additional grass blades per vertex sent to the graphics card for each result. After this comes a short benchmark test that was done to see what the limits of the algorithms and hardware were. Finally, this chapter ends with an image of the final scene of the grass rendered with the final version of the implementation.

3.1.1 Resolution comparison

The configurations shared between the resolutions are the number of vertices sent to the graphics card and the LOD-level values, resulting in the same number of vertices rendered. For a list of the values of the quantitative variables of interest, see table 1 below.

LOD-level 1	8
LOD-level 2	16
LOD-level 3	32
Number of vertices sent to graphics card	~140 000

Table 1. *The values of the quantitative variables used for the resolution comparison tests.*

For a visualization of the values of these variables, see top image in Figure 41. The performance of the algorithms using this configuration is shown in Figure 37 and Figure 38.

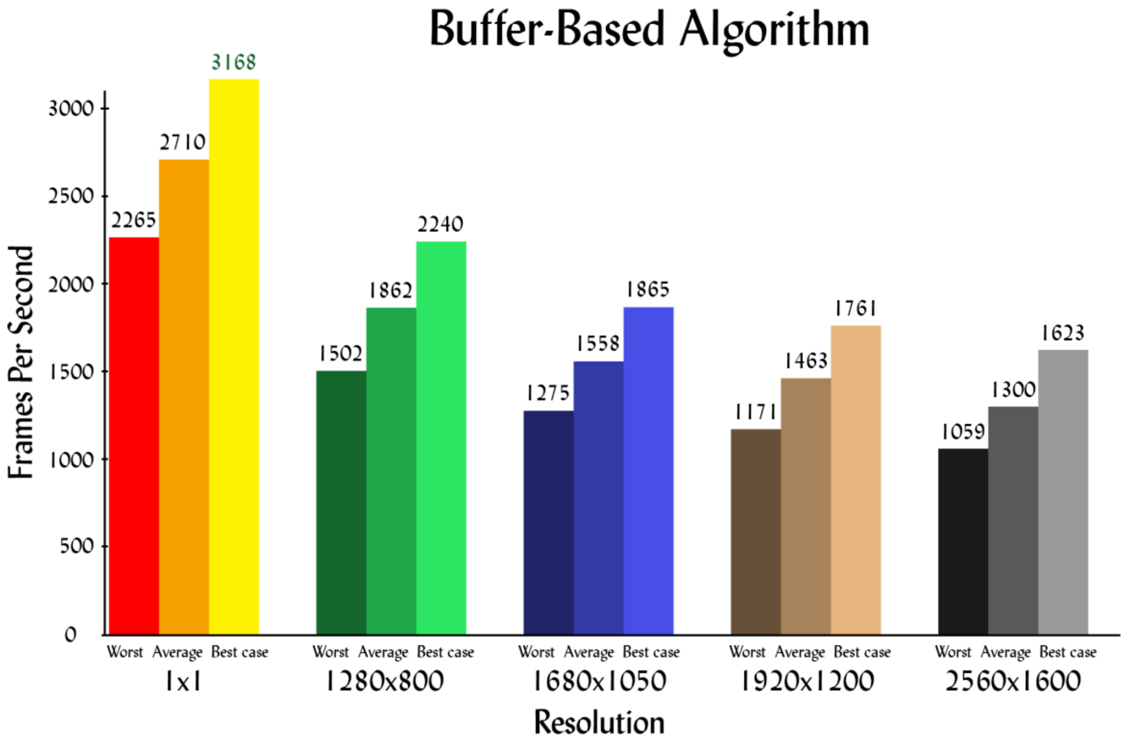


Figure 37. *Graph of the buffer-based algorithm showing the result in FPS of different resolutions with the best, average and best case in terms of FPS of each resolution. Values are rounded off to lowest integer. (Illustration made by Per Axmark.)*

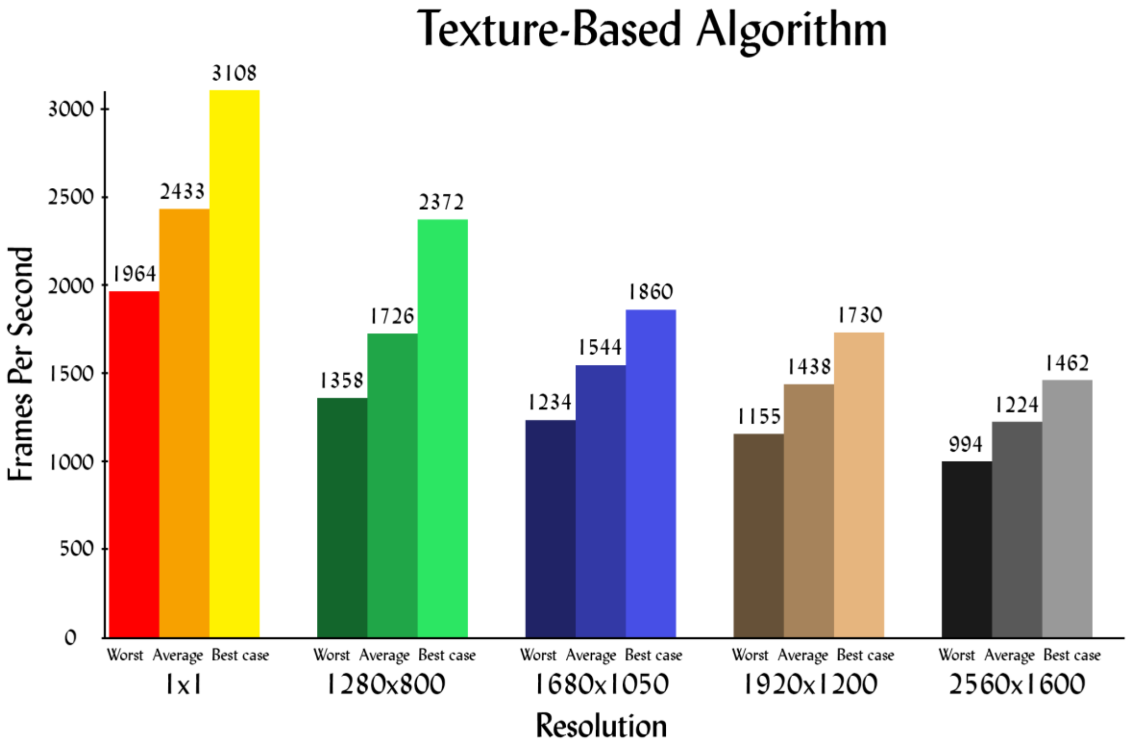


Figure 38. *Graph of the texture-based algorithm showing the result in FPS of different resolutions with the best, average and best case in terms of FPS of each resolution. Values are rounded off to lowest integer. (Illustration made by Per Axmark.)*

3.1.2 LOD-levels comparison

The following results used the same resolution of 1680x1050. Two sets of four different settings were used when generating the results for each algorithm. These settings were aptly named *low*, *normal*, *high* and *ultra*. The LOD-levels' values double per setting of the first set. The second set used the first set's highest LOD-level's value for all LOD-levels. The quantitative variables and their values of interest were as follows:

First result (low)	**Set 1**	**Set 2**
LOD-level 1	4	16
LOD-level 2	8	16
LOD-level 3	16	16
Number of vertices sent to graphics card	~50 700	~50 700
Resolution	1680x1050	1680x1050

Table 2. *The values of the quantitative variables used for the sets of the first result (low).*

Second result (normal)	**Set 1**	**Set 2**
LOD-level 1	8	32
LOD-level 2	16	32
LOD-level 3	32	32
Number of vertices sent to graphics card	~139 800	~139 800
Resolution	1680x1050	1680x1050

Table 3. *The values of the quantitative variables used for the sets of the second result (normal).*

Third result (high)	**Set 1**	**Set 2**
LOD-level 1	16	64
LOD-level 2	32	64
LOD-level 3	64	64
Number of vertices sent to graphics card	~480 000	~480 000
Resolution	1680x1050	1680x1050

Table 4. *The values of the quantitative variables used for the sets of the third result (high).*

Fourth result (ultra)	Set 1	Set 2
LOD-level 1	32	128
LOD-level 2	64	128
LOD-level 3	128	128
Number of vertices sent to graphics card	~1 786 300	~1 786 300
Resolution	1680x1050	1680x1050

Table 5. *The values of the quantitative variables used for the sets of the fourth result (ultra).*

The results are shown in Figure 39 for algorithm 1, and Figure 40 for algorithm 2. For a visualization of the LOD-levels' values, see Figure 41.

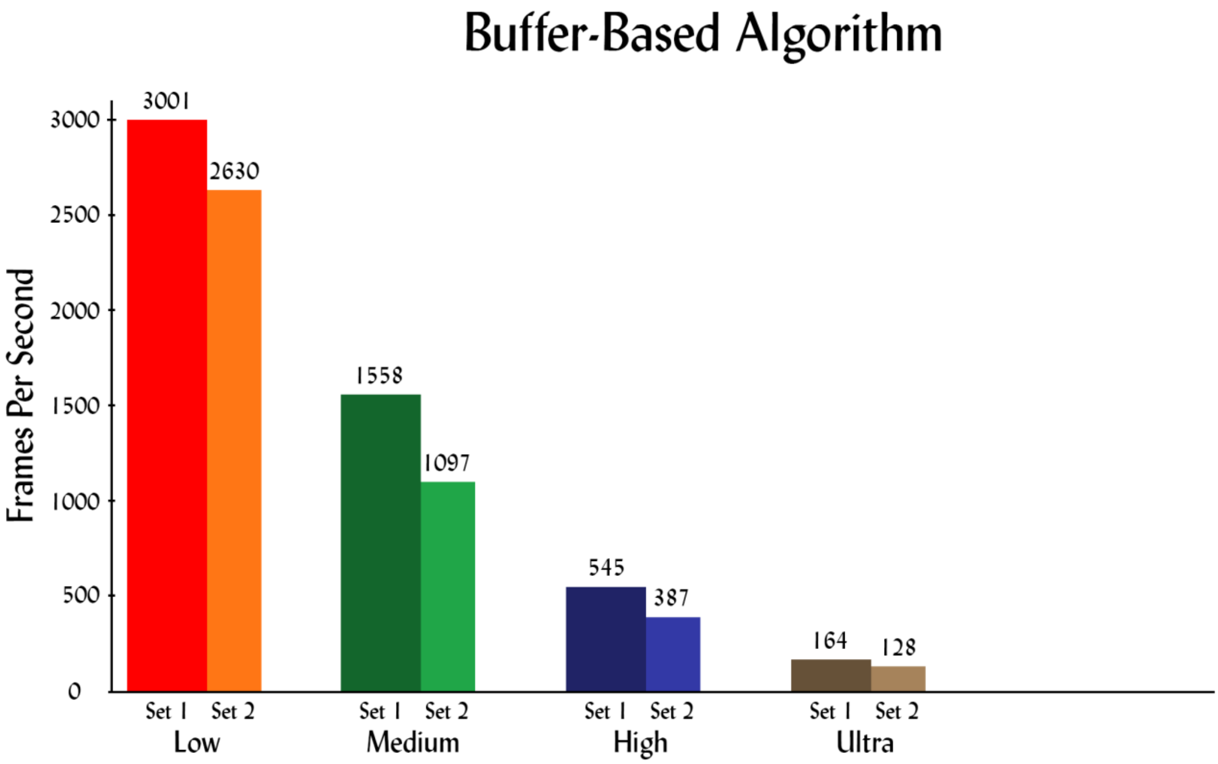


Figure 39. *Graph showing the result in FPS of the buffer-based algorithm using different LOD-level values. Values are rounded off to lowest integer. (Illustration made by Per Axmark.)*

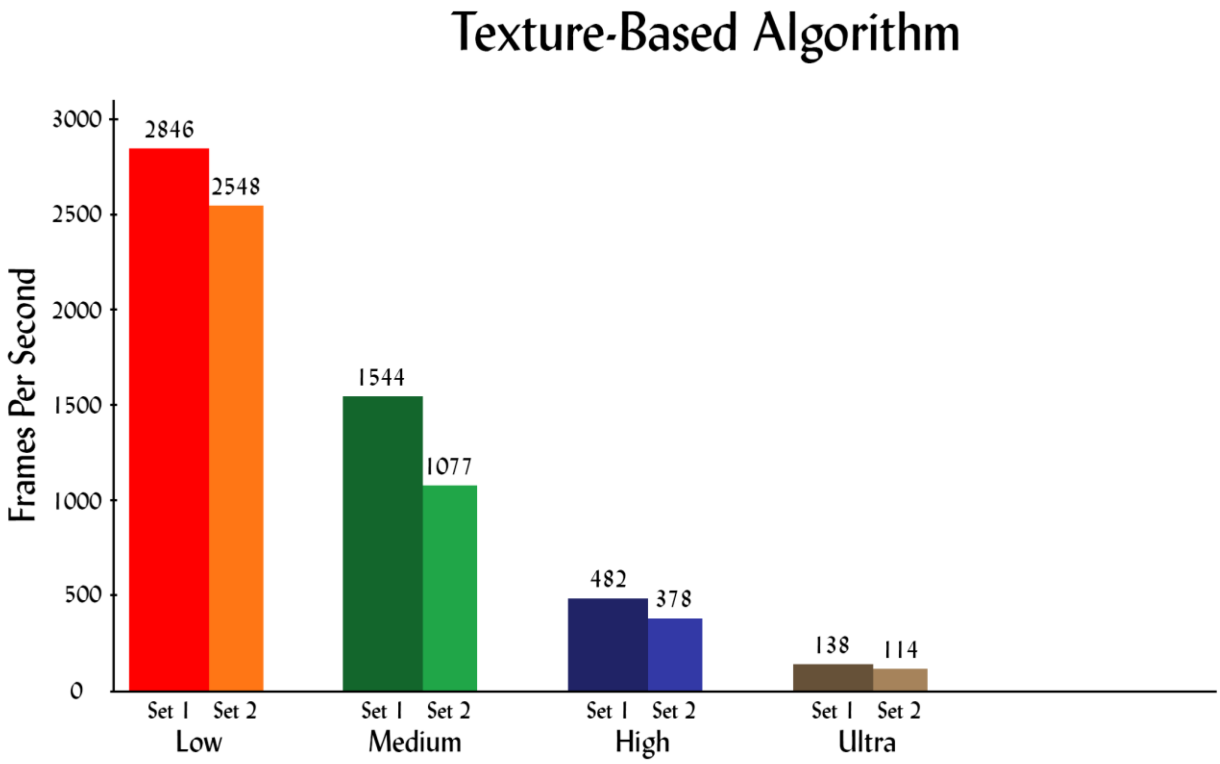


Figure 40. *Graph showing the result in FPS of the texture-based algorithm using different LOD-level values. Values are rounded off to lowest integer. (Illustration made by Per Axmark.)*

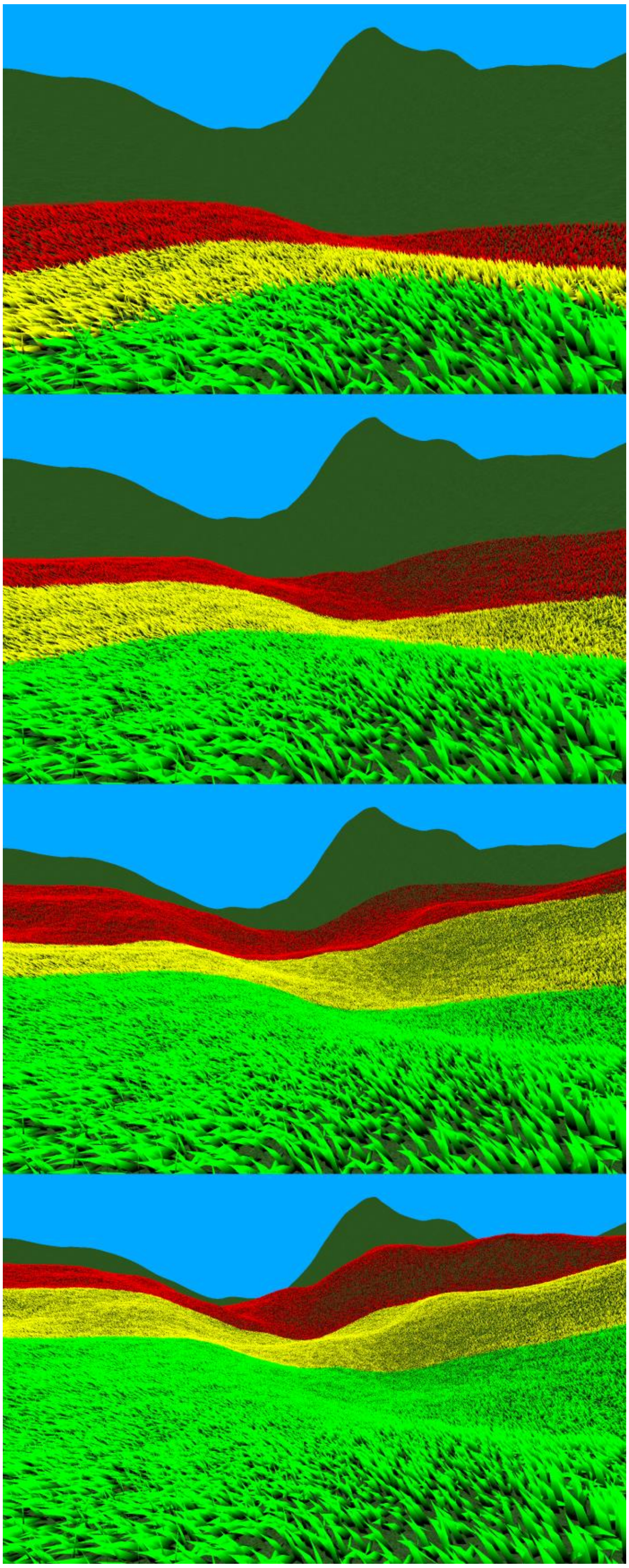


Figure 41. *A visualization of the LOD-levels' values used to gather the results of the LOD-level comparisons. From top to bottom: First result (low) to the fourth result (ultra).*

3.1.3 Creating additional grass blades

For the coming results, the same number of grass blades rendered was approximately the same and the only quantitative variables that changed between these results were the number of grass vertices sent to the graphics card and the number of grass blades created from these vertices. A total of 3 results were generated for each algorithm where 0, 3 and 8 extra grass blades per vertex were created. The first result was named A and the following B and C. See the tables and Figures below for a complete list of the variables and their values as well as the results of using these values.

Result A	
Number of grass blades created per vertex	0
LOD-level 1	8
LOD-level 2	16
LOD-level 3	32
Number of vertices sent to graphics card	~139 800
Number of grass blades created	~139 800
Resolution	1680x1050

Table 6. *The values of the quantitative variables used to generate the first result.*

Result B	
Number of grass blades created per vertex	3
LOD-level 1	8
LOD-level 2	16
LOD-level 3	32
Number of vertices sent to graphics card	~36 400
Number of grass blades created	~145 600
Resolution	1680x1050

Table 7. *The values of the quantitative variables used to generate the second result.*

Result C	
Number of grass blades created per vertex	8
LOD-level 1	8
LOD-level 2	16
LOD-level 3	32
Number of vertices sent to graphics card	~15 500
Number of grass blades created	~139 500
Resolution	1680x1050

Table 8. *The values of the quantitative variables used to generate the third result.*

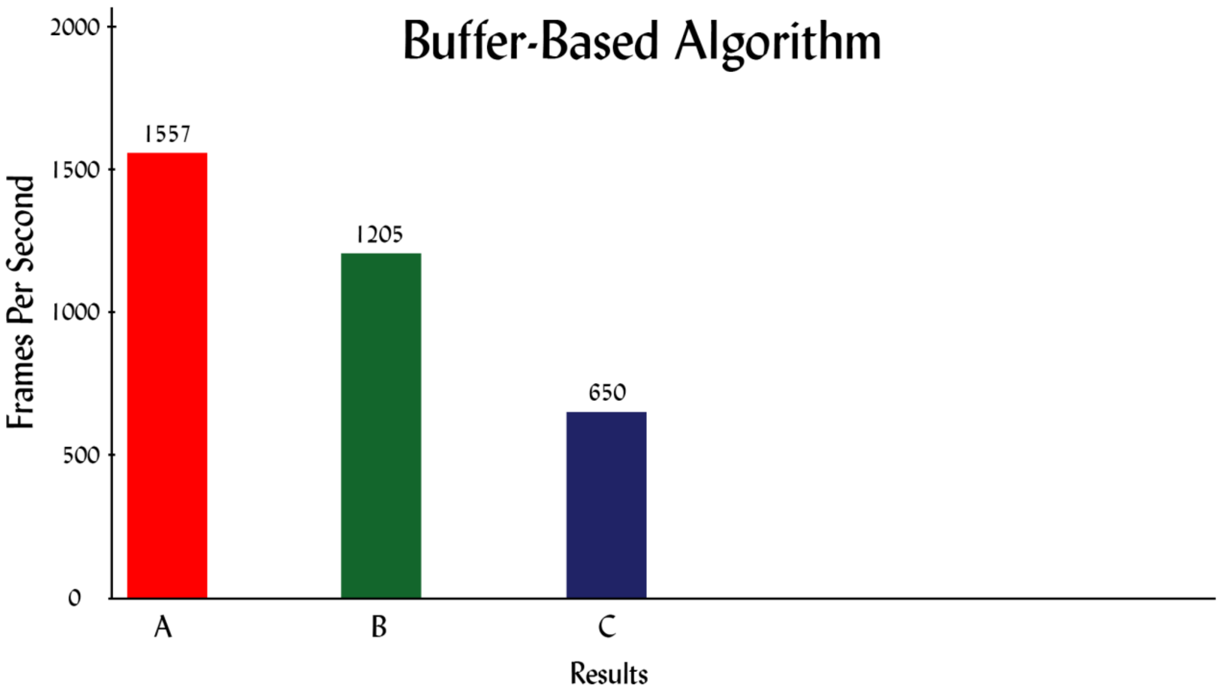


Figure 42. *Graph showing the results in FPS of the buffer-based algorithm using different amounts of data sent to and created in the geometry shader. Values are rounded off to lowest integer. (Illustration made by Per Axmark.)*

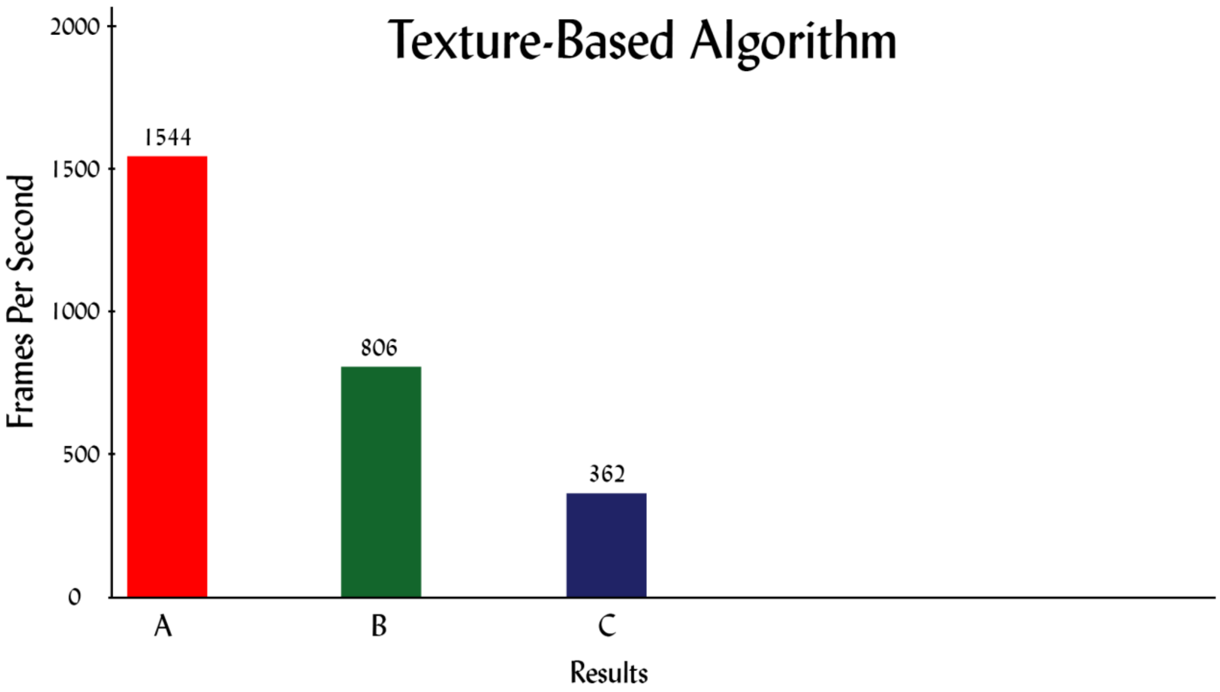


Figure 43. *Graph showing the results in FPS of the texture-based algorithm using different amounts of data sent to and created in the geometry shader. Values are rounded off to lowest integer. (Illustration made by Per Axmark.)*

3.1.4 Benchmark

The settings used for this benchmark is as follows: resolution of 1680x1050 pixels and LOD-level values at virtual infinity. 9 144 576 vertices were sent to the graphics card and 64 012 032 vertices were created from these and rendered. The buffer-based algorithm's performance in terms of FPS averaged to 27.56 FPS and the texture-based algorithm to 17.04 FPS. In other words, the buffer-based algorithm took 36.28 milliseconds to render one frame and the texture-based algorithm took 58.69 milliseconds to render one frame.

3.1.5 Final scene

After two iterations of feedback on the perceived visual quality of the grass and changes made to the algorithms based on this feedback, the last and final version of the grass now looks as in Figure 44 below.

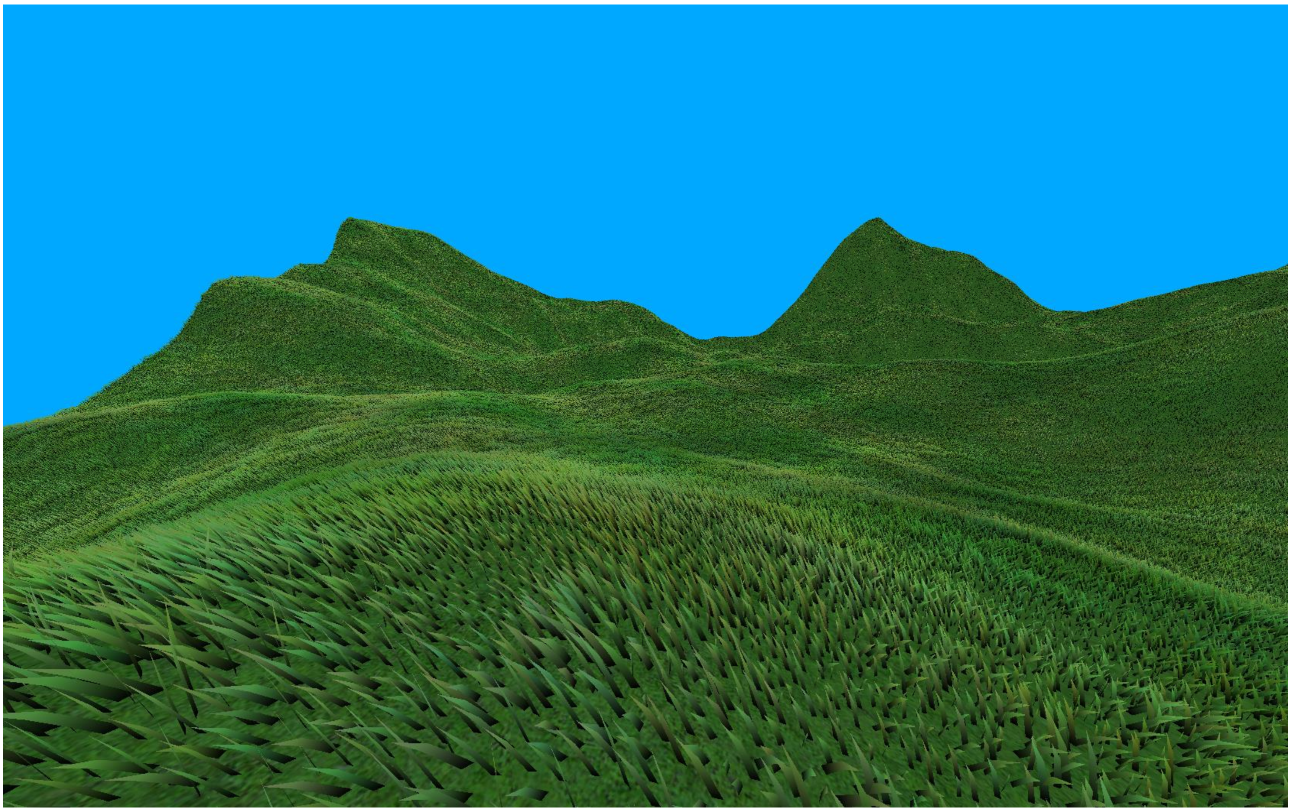


Figure 44. *Final version of the grass.*

3.2 Qualitative data results

The total number of people who took the liberty of filling in the *first* version of the form was 21. The results from this questionnaire are listed in table 9. See Appendix A1 for the complete list of questions with their associated images and Appendix A2 for a raw list of the results.

Q1. Age.	**Average age of participants was 23.10 years.**
Q2. Gender.	**95.24% of participants answered male.**
Q3. How many hours per week do you spend on playing games?	**Average of 13.33+ hours.**
Q4.How many of these hours do you spend on playing games in 3D (3 dimensions) where grass is a part of the game?	**Average of 9.66+ hours.**
Q5.Choose the image which you perceive to contain the highest quality.	**23.81% chose image 1.** **42.86% chose image 2.** **33.33% chose image 3.**
Q6. In the image above, do you see a recurring pattern?	**81% said no.**
Q7. If your answer was yes, what do you see? In other words, what made you say yes?	**See Appendix A2 for the results**

Q8. Do you see a recurring pattern in the image above?	**81% said no.**
Q9. If your answer was yes, what do you see? In other words, what made you say yes?	**See Appendix A2 for the results**
Q10. In the image above, how does the grass look in close view?	**Average score of 2.71 out of 5.**
Q11. In the same image, how does the grass look in the distance?	**Average score of 3.62 out of 5.**
Q12. Should the overall color of the grass change?	**42.9% said yes, darker.** **4.7% said yes, lighter.** **52.4% said no.**
Q13. Should there be more variation in the color of the grass?	**66.66% said yes.**
Q14. Should the width of the grass blades change?	**85.7% said that the grass blades should be thinner.** **14.3% said it was fine as is.**
Q15. Should the height of the grass blades change?	**23.81% said that the grass should be shorter.** **14.38% said that the grass should be taller.** **61.9% said that the height should not change.**
Q16. Should there be more variation in the height of the grass?	**71.2% said yes**
Q17. Does the grass look realistic?	**Average score of 2.43 out of 5.**
Q18. Does the grass in general look good?	**Average score of 3.48 out of 5.**
Q19. Would you have liked to see variation in density and grass species?	**76.2% said yes, both.** **9.5% said yes, density.** **4.8% said yes, grass species.** **9.5% said no.**

Table 9. *The results of the first questionnaire. The results are in percentage or average depending on the type of answer.*

Based on the feedback from the first questionnaire, the grass was changed as well as the questions and associated images; see Appendix A1 and Appendix B1. The total number of answers given in the second questionnaire amounted to 33. Some of the questions asked and their results are shown in the table below. See Appendix B1 for the complete list of questions with their associated images and Appendix B2 for a raw list of results.

Q1. Age.	**Average age of participants was 25.1 years.**
Q2. Gender.	**70.0% of participants were male.** **30.0% were female.**
Q3. Please rate the quality of the grass in the image (image 1) above.	**Average score of 3.2 out 5.**
Q4. The image (image 2) above shows several patterns*, do you see a pattern in the image (image 3) below?	**36.4% said they saw a pattern up close.** **6.1% said they saw a pattern in the distance.** **15.1% said they saw a pattern in between.** **42.4% said they did not see a pattern.**
Q5. If you saw a pattern, what type of pattern did you see?	**44.5% said they saw a pattern in the direction (rotation) of the grass blades.** **21.2% said they saw a pattern in the positioning of the grass blades.** **22.2% said they saw a pattern in the coloring of the grass blades.** **27.3% said they saw a pattern in the height of the grass blades.**
Q6. Should there be more variation in the color of the grass?	**42.4% said yes.** **57.6% said no.**
Q7. Should there be more variation in the height of the grass?	**54.5% said yes.** **45.5% said no.**
Q8. Should the width of the grass blades change?	**81.8% said yes, thinner.** **0% said yes, wider.** **18.2% said no.**
Q9. Should the overall color of the grass blades change?	**6.1% said yes, darker.** **15.1% said yes, lighter** **78.8% no.**
Q10. Should the overall height of the grass blades change?	**21.2% said yes, shorter.** **9.1% said yes, taller.** **69.7% said no.**
Q11. What is the quality of the grass in close view?	**Average score of 3 out of 5.**
Q12. What is the quality of the grass in the distance?	**Average score of 3.70 out of 5.**
Q13. What is the quality of the grass in between?	**Average score of 3.88 out of 5.**
Q14. What is the overall quality of the grass?	**Average score of 3.67 out of 5.**

Q15. Does the grass overall look realistic?	**Average score of 2.88 out of 5.**
Q16. Would you have liked to see variation in density and grass species	**60.6% said yes, both.** **12.1% said yes, density.** **9.1% said yes, species.** **18.2% said no.**

Table 10. *The results of the second questionnaire. The results are in percentage or average depending on the type of answer.*

4 Discussion

First the quantitative results will be discussed and then the benefits and drawbacks of using the terrain data to generate the grass blade vertices. The chapter will then end with a discussion of the qualitative results, beginning with the first wave of results.

Looking at Figure 37 and Figure 38 in 3.1.1 Resolution based comparison, the result for the resolution of 1x1 almost shows the performance of the geometry shader as the vertex shader executes nothing and the pixel shader only executes a few lines of instructions for this pixel. Mentioned earlier in this study, all the pixel shader does is calculating the lighting of the scene. As the resolution increases, so does the load on the pixel shader resulting in a linear decline in FPS.

By looking at the LOD-level comparison in chapter 3.1.2, as the outer LOD-levels' range is increased, so is the number of vertices sent to the graphics card. Judging by the results of set 1 in Figure 39 and Figure 40, where different values were used for each LOD-level, the drop in performance measured in frames per second seems to drop linearly as the number of vertices sent to the graphics card is increased linearly. Comparing these numbers with the results of set 2 in the same Figures, the load on the geometry shader seems to increase exponentially as LOD-level's values are also exponentially increased. The relation of the LOD-level values with the performance is therefore linear. Comparing the algorithms show that the buffer-based algorithm is superior to the texture-based algorithm under the same conditions in terms of FPS. Mentioned earlier in this study, rendering grass blades that only contribute to one or no pixels is useless. This starts to happen when the LOD-level's value are roughly 32.

The average FPS is 1224 when rendering the scene at a resolution of 2560x1600 using the *texture-based algorithm* when the variable values from chapter 3.1.1 Resolution are used. In other words the grass takes, on average, 0.817 milliseconds to render. To be able to render a scene at an acceptable frame rate of 60 (real-time), the scene has to take less than 16.67 milliseconds to render. This means that the rendering of the grass constitute 4.902% of the rendering time. This may not seem much at first glance. However, if put in context with games, lots of other objects needs to be rendered, physics simulations need to be run as well as other factors need to be considered, this number starts to seem big. Rendering individual grass blades using geometry in a game is definitely possible as it has already been proven by the game *Flower*. However, as just mentioned, if put in context with a game that need to worry about many other things than rendering grass, such as *World of Warcraft*[17] or *Assassin's creed 3*[18], this number becomes a lot. The conclusion from this is that games of these types are not yet ready for rendering grass with geometry, but that we will in the near future begin to see more games using individual grass blades represented with vertices.

The buffer-based algorithm does not have sufficient data to accurately create additional grass blades per vertex whereas the texture-based algorithm does. For example, the positioning of the extra grass blades generated from a vertex could wound up in the air or under the terrain unless all vertices are on a planar surface and the orientation of this surface is known. Just looking at the results for creating additional grass blades per vertex in chapter 3.1.3 Creating additional grass blades tells you that it's a drawback to create additional grass blades per vertex as this reduces the FPS significantly. However, these results conflict with results from early testing of the buffer-based algorithm where there was a clear increase in FPS to send fewer vertices and create more grass blades per grass vertex. There should have been an increase in FPS but it was unclear as to why it was the opposite. Sadly, there was not enough time to find the cause and was left for future work to find out.

[17] Developed by Blizzard Entertainment™. Website: http://eu.blizzard.com/en-gb/. [Accessed: 21-May-2013].
[18] Developed by Ubisoft™. Website: http://www.ubi.com. [Accessed: 21-May-2013].

A disadvantage of not using the terrain data when generating the grass blades is that the grass has to be removed if they are positioned on top of sand, rock, water or any other surface grass cannot grow on. Note that different kinds of grass species can grow on different kinds of surfaces. The advantage of using the terrain data is that it can be used to determine where the grass should grow, how its growth should be, its color and orientation. For example, the elevation and type of the terrain can be used to determine what grass species to use. Another example is that the surface normals of the terrain can be used to determine if the grass blades should be generated and what its orientation should be as grass cannot grow on surfaces that are too steep. A disadvantage of using the terrain texture to color the grass blades is that the sampled texels could contain highly different values than the surrounding ones and therefore result in unwanted behavior, still requiring special attention. This disadvantage should disappear or at least be reduced by doing a good terrain texture and/or by sampling neighboring texels and making a comparison test to determine if it should be created and what properties it should have. See Figure 22 in 2.1.2 Algorithms for a reminder of this disadvantage. Before moving on to the qualitative results, one final advantage of using the terrain data to generate the grass vertices is worth mentioning. The canopy of the grass blades can be used to render the terrain where grass is not rendered. The advantage is that objects that are hidden in the grass are not visible even if the grass is not rendered [1].

In the *first* wave of the qualitative results, 81% said they did not see a recurring pattern in any of the images, this lead to the conclusion that a recurring pattern needed to be defined and used as a base when comparing the results. The perceived quality of the result sums up the score given on the questions of how the grass looks. This was also concluded to being needed to be defined for the same reason. The realism of the grass was perceived to be between bad and mediocre on a scale of 1-5, from very bad to very good. The overall quality was perceived to be right between mediocre and good using the same scale. Similar score was given on how the grass looked in close view and in the distance. Many said that the overall color of the grass should be darker. The majority said there should be more variation in the color and height of the grass and that the grass blades should be thinner. One piece of extra feedback given is worth mentioning as one of the participants said: “The grass looks good but I thought that they should be thinner to look more realistic.” This piece of feedback and other answers led to the conclusion that the width of the grass blades should be reduced to increase the realism. As a result of the interpreted results of the first questionnaire, the overall color of the grass was changed to a darker shade of green, as well as more variation in the color and height of the grass was introduced. Also, the width of the grass blades was reduced to increase the realism. It was concluded from the answers of the first questionnaire that a question specified to filter off automated response was unnecessary. It was also decided that a reference for quality and pattern was needed in order for the answers to the questions concerning these to be valid.

In the *second* wave of results, 36% said that they saw a pattern in close view to the grass and of these, most said that they saw a pattern in the rotation (direction) of the grass blades. Compared to the first wave of results, fewer participants said that there should be more variation in the color and height of the grass, but this number still remained a significant amount that cannot be ignored. The vast majority said that the width of the grass blades should change. Compared to the first wave of results where 43% said that the grass should overall be darker, 4 out of 5 now said that the overall color of the grass should not be changed. The quality of the grass used in the previous wave of results was given a score of 3.2 out of 5 where the quality of the grass modified based on the first wave of results was given a score of 3.7 out of 5. This, together with the differences in the answers from the first and second wave of results gives the conclusion that the grass was improved. As a result of the feedback from the second questionnaire, the variation in color and height was increased and the width of the grass blades was reduced. Note that a threat to the validity of the answers concerning the realism and size of the grass blades is that a reference, or a base, was not established for them. Another threat to the validity in the answers could be differences in the answers given by the people who answered both questionnaires, with the ones that only filled out the second questionnaire.

In both waves of results, threats to the validity of the answers given on the questions about the color of the grass is affected by the color scheme, contrast and brightness of the screen used by the people who answered these questions. Also, statistical significances may be a threat to the conclusions made from the answers. In both of the questionnaires, the majority said that they wanted to see a variation in density and grass species, which will be a topic for future study.

5 Conclusions

Following the flow of the previous chapter, the conclusions made from the qualitative results will be presented first and then the benefits and drawbacks of using the terrain data to generate the grass blade vertices. This chapter will then end with the conclusions made about the qualitative results. Each paragraph will answer the research questions *RQ1*, *RQ2* and *RQ3*, respectively.

Algorithm 1 is superior in terms of performance to algorithm 2 if they use the same conditions. Algorithm 2 becomes superior when it comes to creating additional grass blades as it has enough data to accurately create them. An advantage of generating extra grass blades per vertex is that this number can easily be managed by a LOD-system. Pointed out in the previous chapter, the biggest limitation and bottleneck is the sheer amount of data that needs to be sent to the graphics card in order to be able to render the grass. The drawback of algorithm 1 is that it does not have sufficient data to accurately generate extra grass blades per vertex unless it's on a planar surface where the normal of the surface is known. An advantage of using textures instead of buffers to store and use the vertices is that it opens up more options as the geometry shader has access to all the grass vertices when the grass blades are created.

Using the terrain data has its advantages and disadvantages. If done right, the grass blades can be automatically generated without the need for manual after-treatment. The disadvantage is that the generation of the grass vertices becomes dependent on the data of the terrain.

The perceived quality of the grass in terms of realism is close to mediocre as the average score of realism was 2.88 out of 5, ranging from 1, very unrealistic, to 5, very realistic. The cause is most likely because the majority wanted the grass blades thinner and wanted to see more variation in the color and height of the grass. As the color of the grass is now, it is just uniformly interpolated shades of green. Introducing a grass texture would most likely increase the realism by introducing non-uniformity in the coloring of the individual grass blades. Based on other feedback given, another conclusion is that the realism of the grass could be improved by using a better lighting model. The overall perceived quality of the grass was between mediocre and good, leaning towards good with an average score of 3.68 out of 5 on a scale of 1 to 5, very bad to very good. Pointed out by a participant, using an anti-aliasing algorithm would have removed the aliasing which most likely would improve the quality of the grass.

6 Future work

Previously discussed, there should have been an increase in FPS to send fewer vertices to the graphics card and compensate the number of grass blades rendered by creating additional grass blades per vertex. However, the results showed the opposite and it was unclear as to why this happened and an answer is worth endeavoring. It was also mentioned that the texture-based algorithm opened up more options which was not explored much in this study. For instance, the LOD-system could have managed the number of grass blades created per vertex as well as their size.

Rendering grass is a broad subject and this study left many things left to be done. Methods that would increase the realism of the grass are worth exploring. Examples of such methods would be to use an improved lighting model, a grass blade texture and anti-aliasing. For instance, screen space ambient occlusion (SSAO) could be added to or replace the lighting algorithm. Also, the majority of answers in both questionnaires clearly indicate that variation in the density of the grass as well as variation in grass species is of interest.

Further utilization of the terrain data to generate the grass blades is also worth exploring. For instance, edges of the generated grass fields caused by differences in the terrain data could be smoothed out by a post-processing step by using the neighboring grass blades and the terrain data.

Animation is also an important aspect of making the grass realistic where wind and gravity simulations and interact ability with the grass, such as treading is worth researching. To increase the density, and at the same time introduce less uniformity, another shape could be used for shorter grass blades.

To increase the performance in terms of frames per second, various acceleration algorithms could be applied by, for instance, using hierarchical view frustum culling.

Finally, it’s worth mentioning that grass rendering is not exclusive to entertainment such as games. For example, the simulation of different methods and scenarios of growing grass to be turned into fuel could be done to find the cheaper, more efficient and profitable method and environment.

7 References

[1] B. Kemen, “Outerra: Procedural Grass Rendering,” *Outerra*, 27-May-2012. [Online]. Available: http://outerra.blogspot.se/2012/05/procedural-grass-rendering.html. [Accessed: 04-May-2013].

[2] D. Mc Feely, *Rendering Interactive Flora for Computer Games*. 2013.

[3] K. Boulanger, “Real-Time Realistic Rendering of Grass with Lighting,” Master thesis, University of Rennes 1, Rennes, 2005.

[4] K. Pelzer, “GPU GEMS. Chapter 7. Rendering Countless Blades of Waving Grass,” 04-Apr-2013. [Online]. Available: https://developer.nvidia.com/content/gpu-gems-chapter-7-rendering-countless-blades-waving-grass. [Accessed: 09-Apr-2013].

[5] E. Lee, “Realistic Real-Time Grass Rendering,” Master Thesis, The Guildhall Southern Methodist University, Los Angeles, 2010.

[6] M. A. Shah, J. Kontinnen, and S. Pattanaik, “Real-time rendering of realistic-looking grass,” in *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, New York, NY, USA, 2005, pp. 77–82.

[7] K. Boulanger, S. Pattanaik, and K. Bouatouch, “Rendering Grass in Real-Time with Dynamic Light Sources and Shadows,” Research report, Jul. 2006.

[8] S. Lemme, “High-Quality Rendering of Grass in Real-Time,” Master Thesis, University of Potsdam, Potsdam, 2012.

[9] W. T. Reeves, “Particle systems - a technique for modeling a class of fuzzy objects,” *SIGGRAPH Comput. Graph.*, vol. 17, no. 3, pp. 359–375, Jul. 1983.

[10] T. Akenine-Möller, E. Haines, and N. Hoffman, *Real-time rendering*, 3rd ed. Wellesley, Mass.: A.K. Peters, 2008.

[11] F. Perbet and M.-P. Cani, “Animating prairies in real-time,” in *Proceedings of the 2001 symposium on Interactive 3D graphics*, New York, NY, USA, 2001, pp. 103–110.

[12] S. Guerraz, F. Perbet, D. Raulo, F. François, and M.-P. Cani, “A Procedural Approach to Animate Interactive Natural Sceneries,” in *Computer Animation and Social Agents (CASA)*, 2003.

[13] J. Kruger and R. Westermann, “Acceleration techniques for GPU-based volume rendering,” in *Visualization, 2003. VIS 2003. IEEE*, 2003, pp. 287–292.

[14] D. Santa-Cruz and T. Ebrahimi, “Coding of 3D virtual objects with {NURBS},” *Signal Processing*, vol. 82, no. 11, pp. 1581 – 1593, 2002.

[15] E. Dimas and D. Briassoulis, “3D geometric modelling based on NURBS: a review,” *Advances in Engineering Software*, vol. 30, no. 9–11, pp. 741 – 751, 1999.

[16] K. Perlin, “Improving noise,” in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2002, pp. 681–682.

[17] S. Gustavson, “Simplex noise demystified,” Linköping University, Linköping, Sweden, Research Report, Mar. 2005.

[18] B. T. Phong, “Illumination for computer generated pictures,” *Commun. ACM*, vol. 18, no. 6, pp. 311–317, Jun. 1975.

[19] “Nvidia tillkännager arkitekturen Volta - Grafikkort - SweClockers.com,” *Sweclockers, translated by Markus Tillman*. [Online]. Available: http://www.sweclockers.com/nyhet/16719-nvidia-tillkannager-arkitekturen-volta. [Accessed: 09-May-2013].

Appendix A1

The following show the contents of the first questionnaire.

Rendering grass in computer graphics

This survey is a part of a bachelor thesis at Blekinge Institute of Technology in Karlskrona. 22 questions will be asked and roughly takes 5 minutes to fill out. Its purpose is to give an insight in what we perceive as quality when it comes to the representation of grass in computer graphics. You will be anonymous and your answers will only be used for academic purposes. It's possible to fill out the survey more than once, but please refrain from doing so. Please feel free to distribute the link to this survey to whoever you like. The survey will close at 20-05-2013.

* Required.

0. Please prove that you are a human and not a machine. How much is two plus two? *

   Mark only one oval.

   - 3
   - 4
   - 5
1. Age. *
   - Their answer
2. Gender. *

   Mark only one oval.

   - Male
   - Female
3. How many hours per week do you spend on playing games? *

   Mark only one oval.

   - 0
   - 1-5
   - 6-10
   - 11-15
   - 16+
4. How many of these hours do you spend on playing games in 3D (3 dimensions) where grass is a part of the game? *

   Mark only one oval.

   - 0
   - 1-5
   - 6-10

- 11-15
- 16+

Figure 45. *Image 1.*

Figure 46. *Image 2.*

Figure 47. *Image 3.*

5. Choose the image which you perceive to contain the highest quality. *

Mark only one oval.

- Image 1
- Image 2
- Image 3

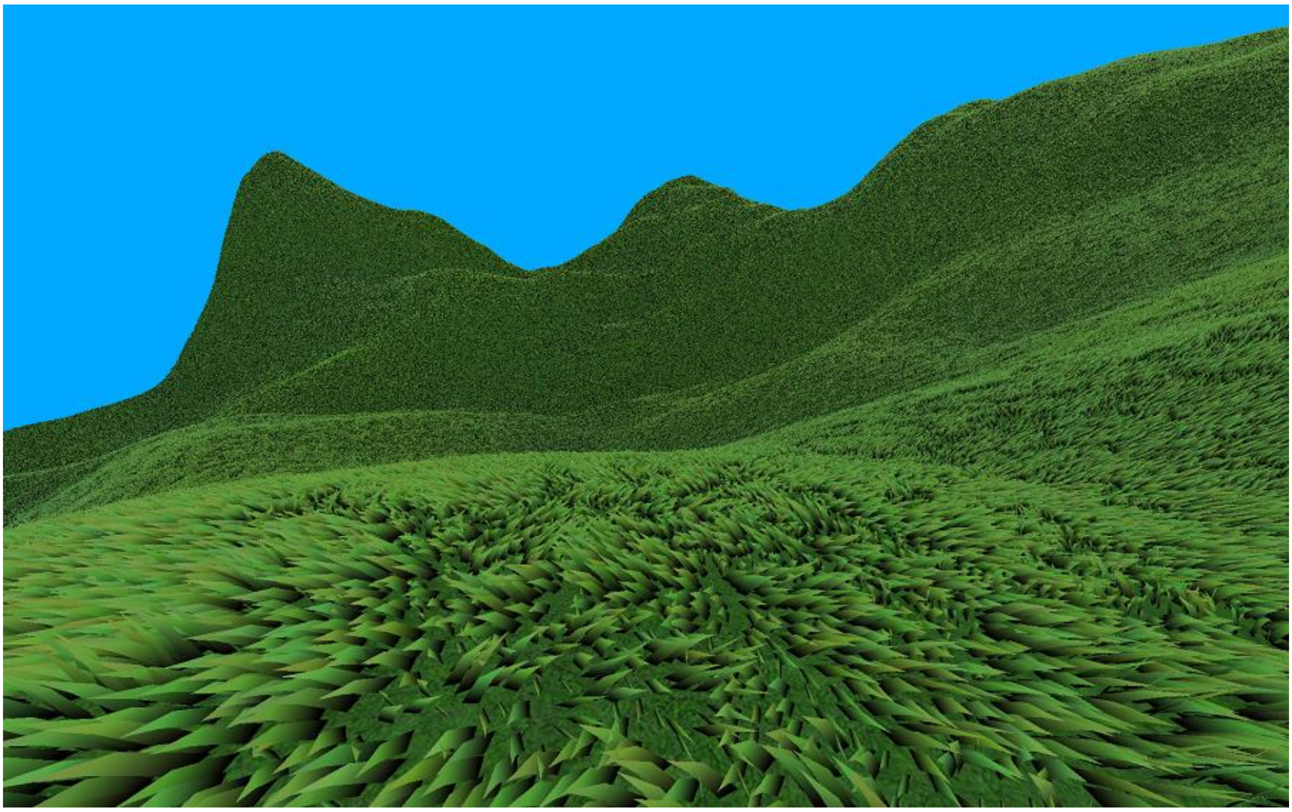
6. In the image above, do you see a ***recurring*** pattern? *

   Mark only one oval.

   - Yes
   - No

7. If your answer was ***yes***, what do you see? In other words, what made you say yes?

   - Their answer.



8. Do you see a ***recurring*** pattern in the image above? *

   Mark only one oval.

   - Yes
   - No

9. If your answer was ***yes***, what do you see? In other words, what made you say yes?

   - Their answer.

10. In the image above, how does the grass look in close view? *

    Mark only one oval.

    Select a value from a range of 1, Very bad, to 5, Very good.

11. In the same image, how does the grass look in the distance? *

    Mark only one oval.

    Select a value from a range of 1, Very bad, to 5, Very good.

12. Should the overall color of the grass change? *
    - Yes it should be darker
    - Yes it should be lighter
    - No.
13. Should there be more variation in the color of the grass? *
    - Yes
    - No.
14. Should the width of the grass blades change? *
    - Yes they should be smaller
    - Yes they should be wider
    - No.
15. Should the height of the grass blades change? *
    - Yes they should be shorter

- Yes they should be taller
- No.

16. Should there be more variation in the height of the grass? *
    - Yes
    - No.
17. Does the grass look realistic? *

    Mark only one oval.

    Select a value from a range of 1, Very bad, to 5, Very good.
18. Does the grass in general look good? *

    Mark only one oval.

    Select a value from a range of 1, Very bad, to 5, Very good.
19. Would you have liked to see variation in density and grass species?
    - Yes both
    - Yes variation in density
    - Yes variation in grass species
    - No.
20. Finally, if you were to search for information on how the grass was made, what key/search-words would you use? *
    - Their answer.
21. Other feedback.
    - Their answer.

Appendix A2

The following tables show the answers of the first questionnaire.

Timestamp	**0. Please prove that you are a human and not a machine. How much is two plus two?**	**1. Age**	**2. Gender**	**3. How many hours per week do you spend on playing games?**	**4. How many of these hours do you spend on playing games in 3D(3 dimensions) where grass is a part of the game?**	**5. Choose the image which you perceive to contain the highest quality.**	**6. In the image above, do you see a recurring pattern?**
5/16/2013 14:45:49	4	21	Male	16+	6-10	Image 3	No
5/16/2013 14:50:11	4	22	Male	16+	16+	Image 3	No
5/16/2013 15:06:40	4	22	Male	11-15	1-5	Image 2	Yes
5/16/2013 15:28:21	4	22	Male	16+	16+	Image 3	No
5/16/2013 15:45:59	4	22	Male	16+	16+	Image 3	No
5/16/2013 18:35:10	4	21	Male	16+	16+	Image 1	No
5/17/2013 13:02:29	4	22	Male	1-5	1-5	Image 2	No
5/17/2013 13:12:28	4	22	Male	6-10	6-10	Image 3	No
5/17/2013 13:23:22	4	22	Male	16+	0	Image 1	No
5/17/2013 13:56:03	4	22	Male	16+	16+	Image 3	No
5/17/2013 14:00:31	4	22	Male	16+	16+	Image 1	No
5/17/2013 14:05:46	4	22	Male	11-15	6-10	Image 2	Yes
5/17/2013 14:48:49	4	24	Male	6-10	6-10	Image 2	Yes
5/17/2013	4	23	Male	1-5	1-5	Image 2	No

14:51:50							
5/17/2013 16:04:25	4	22	Male	16+	0	Image 2	No
5/17/2013 16:26:37	4	22	Male	16+	6-10	Image 2	Yes
5/18/2013 14:52:11	4	29	Male	6-10	6-10	Image 1	No
5/18/2013 15:35:35	4	21	Female	16+	11-15	Image 1	No
5/18/2013 15:40:29	4	29	Male	16+	11-15	Image 3	No
5/18/2013 15:45:29	4	26	Male	16+	16+	Image 2	No
5/18/2013 18:30:26	4	27	Male	16+	6-10	Image 2	No
5/19/2013 18:53:38	4	21	Male	1-5	1-5	Image 3	No

7. If your answer was yes, what do you see? In other words, what made you say yes?	8. Do you see a recurring pattern in the image above?	9. If your answer was yes, what do you see? In other words, what made you say yes?	10. In the image above, how does the grass look in close view?	11. In the same image, how does the grass look in the distance?	12. Should the overall color of the grass change?	13. Should there be more variation in the color of the grass?	14. Should the width of the grass blades change?
	No		3	4	No	No	Yes, they should be smaller
	No		3	5	No	No	No
Small recurring patterns. Nothing much. Middle bottom of screen looks recurring to me. And to the right	No		4	2	No	No	Yes, they should be smaller
	No		3	4	No	Yes	Yes, they should be smaller
	No		3	2	Yes, it should be darker	Yes	No
	No		3	4	No	No	Yes, they should be smaller
	No		3	4	No	Yes	Yes, they should be smaller
	Yes	Everything follows a vector direction.	1	4	Yes, it should be lighter	Yes	Yes, they should be smaller
	Yes	Most the grass is pointing in	5	1	No	Yes	No

		the same direction.					
	No		3	5	No	No	Yes, they should be smaller
	No		2	2	Yes, it should be darker	Yes	Yes, they should be smaller
Upp ner, upp ner, upp ner.	Yes	Lutning	3	4	Yes, it should be darker	Yes	Yes, they should be smaller
	Yes		2	3	No	Yes	Yes, they should be smaller
	No		2	5	Yes, it should be darker	Yes	Yes, they should be smaller
	No		3	4	Yes, it should be darker	No	Yes, they should be smaller
Similar shapes all over.	No		2	2	Yes, it should be darker	Yes	Yes, they should be smaller
	No		4	4	No	No	Yes, they should be smaller
	No		3	5	Yes, it should be darker	Yes	Yes, they should be smaller
	No		2	5	No	Yes	Yes, they should be smaller
	No		2	3	Yes, it should be darker	Yes	Yes, they should be smaller
	No		1	4	Yes, it should be darker	Yes	Yes, they should be smaller
	No		3	2	No	Yes	No

15. Should the height of the grass blades change?	16. Should there be more variation in the height of the grass?	17. Does the grass look realistic?	18. Does the grass in general look good?	19. Would you have liked to see variation in density and grass species?	20. Finally, if you were to search for information on how the grass was made, what key/search-words would you use?	21. Other feedback
No	No	2	3	No	i want to answer questions about rendering grass bachelor thesis blekinge institute of technology	
No	Yes	3	4	Yes, both	render, grass, quality, graphics	
No	No	2	4	Yes, both	Grass, Render, geometry	
No	Yes	3	3	Yes, both	"Haha aww hell no, you're gonna have to find ur own daauummn keywords! Kidn, Realistic Grass Rendering is what Id search for I guess."	Pewpew. Special types of clutter on the ground is missing, such as moss and flowers etc
Yes, they should be taller	Yes	2	4	Yes, both	häst	
No	No	2	3	Yes, both	how to computer	

					graphic grass rendering	
No	No	3	4	Yes, variation in grass types	grass, digital, creation, realistic, methods, nature, model.	
Yes, they should be taller	Yes	2	4	Yes, both	Grass for games, procedural generated grass.	I think your biggest problem is the grass straw representation, it could change to a lot more realistic and visually pleasing results.
No	No	2	4	Yes, both	grass shader	
No	Yes	3	4	Yes, variation in density	animated 3D grass	
Yes, they should be shorter	Yes	2	2	Yes, both	How to roll a spliff	No
No	Yes	2	3	Yes, variation in density	Instance, outerra	The grass looks like the style that would fit in games of a more comic style, for example LoL, other than what would be expected in a real world setting.
Yes, they should be shorter	Yes	3	3	Yes, both	3d programming grass graphics	
No	Yes	3	4	Yes, both	making of 3d grass for games	

No	No	2	4	No	Grass generation	The grass looks good but i think that they should be thinner to look more realistic.
Yes, they should be shorter	Yes	1	2	Yes, both	3D grass generation	None. Sorry.
No	Yes	3	4	Yes, both	.	
Yes, they should be taller	Yes	4	4	Yes, both	procedural generated 3d grass in game	More vibrant colors.
Yes, they should be shorter	Yes	2	4	Yes, both	Grass rendering	
No	Yes	2	3	Yes, both	polygons	

Appendix B1

The following show the contents of the second questionnaire.

Rendering grass in computer graphics

This survey is a part of a bachelor thesis at Blekinge Institute of Technology in Karlskrona. 22 questions will be asked and roughly takes 5 minutes to fill out. Its purpose is to give an insight in what we perceive as quality when it comes to the representation of grass in computer graphics. You will be anonymous and your answers will only be used for academic purposes. It's possible to fill out the survey more than once, but please refrain from doing so. Please feel free to distribute the link to this survey to whoever you like. The survey will close at 29-05-2013.

* Required.

1. Age *
    - Their answer.
2. Gender *

    Mark only one oval.
    - Male
    - Female

Figure 48. *Image 1.*

3. Please rate the quality of the grass in the image (image 1) above. *

   Mark only one oval.

   Select a value from a range of 1, Very bad, to 5, Very good.

Figure 49. *Image 2.*

4. The image (image 2) above shows several patterns*, do you see a pattern in the image (image 3) below? *
   * patterns in the color, orientation, position and height of the grass blades.

   Mark only one oval.

   - Yes, up close (The part of the image that is below the blue line.)
   - Yes, in the distance (The part of the image that is above the red line.)
   - Yes, in between (The part of the image that is in between the red and blue lines.)
   - No.

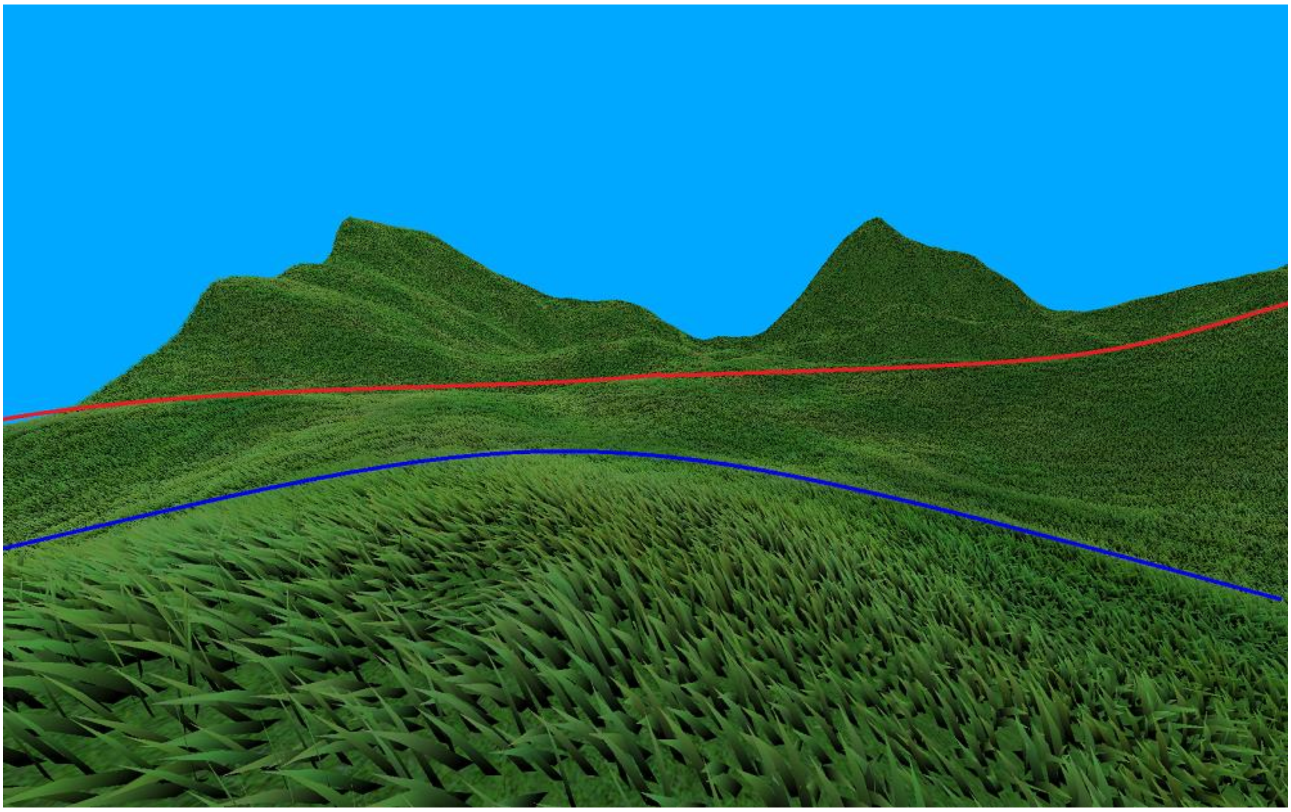


Figure 50. *Image 3.*

5. If you saw a pattern, what type of pattern did you see?
   How the grass blades are rotated (their direction), positioned, colored, the height or other.

   Check all that apply.

   - The rotation (direction of the grass blades.
   - Positioning of the grass blades.
   - Coloring of the grass blades.
   - Height of the grass blades.
   - Other. [Their answer].

The following questions refer to image 3.

6. Should there be more variation in the color of the grass? *

   Mark only one oval.

   - Yes
   - No

7. Should there be more variation in the height of the grass? *

   Mark only one oval.

   - Yes
   - No

8. Should the width of the grass blades change? *

   Mark only one oval.

   - Yes, they should be thinner
   - Yes, they should be wider
   - No

9. Should the overall color of the grass blades change? *

   Mark only one oval.

   - Yes, it should be darker
   - Yes, it should be lighter
   - No

10. Should the overall height of the grass blades change? *

    Mark only one oval.

    - Yes, they should be shorter
    - Yes, they should be taller
    - No

11. What is the quality of the grass in close view? *

    The part of the image (image 3) that is below the blue line.

    Mark only one oval.

    Select a value from a range of 1, Very bad, to 5, Very good.

12. What is the quality of the grass in the distance? *

    The part of the image (image 3) that is above the red line.

    Mark only one oval.

    Select a value from a range of 1, Very bad, to 5, Very good.

13. What is the quality of the grass in between?

    The part of the image (image 3) that is in between the red and blue line.

    Mark only one oval.

    Select a value from a range of 1, Very bad, to 5, Very good.

14. What is the overall quality of the grass? *

    Mark only one oval.

    Select a value from a range of 1, Very bad, to 5, Very good.

15. Does the grass overall look realistic?

    Mark only one oval.

    Select a value from a range of 1, Very unrealistic, to 5, Very realistic.1

16. Would you have liked to see variation in the density and grass species?

    Mark only one oval.

- Yes, both
- Yes, variation in density
- Yes, variation in grass species.
- No

17. Other feedback.

- Their feedback.

Appendix B2

The following tables show the answers of the second questionnaire.

Timestamp	**1. Age**	**2. Gender**	**3. Please rate the quality of the grass in the image (image 1) above.**	**4. The image (image 2) above shows several patterns*, do you see a pattern in the image (image 3) below?**	**5. If you saw a pattern, what type of pattern did you see?**
5/22/2013 10:02:23	22	Male	3	No	
5/22/2013 10:12:26	22	Male	4	Yes, up close (The part of the image that is below the blue line.)	Height of the grass baldes.
5/22/2013 10:50:16	29	Male	4	No	
5/22/2013 15:45:54	21	Male	4	No	
5/24/2013 10:42:31	22	Male	2	No	
5/24/2013 10:42:48	23	Male	3	Yes, in the distance (The part of the image that is above the red line.)	The rotation (direction) of the grass blades., Positioning of the grass blades., Coloring of the grass blades., Height of the grass baldes.
5/24/2013 10:45:48	22	Male	2	Yes, up close (The part of the image that is below the blue line.)	Coloring of the grass blades., Height of the grass baldes.
5/24/2013 11:00:41	27	Male	3	Yes, up close (The part of the image that is below the	The rotation (direction) of the grass

				blue line.)	blades.
5/24/2013 11:00:59	22	Male	4	No	
5/24/2013 12:20:19	29	Female	3	No	The rotation (direction) of the grass blades., Positioning of the grass blades., Height of the grass baldes.
5/24/2013 12:36:35	21	Male	3	No	
5/24/2013 13:12:23	22	Male	4	Yes, up close (The part of the image that is below the blue line.)	Height of the grass baldes.
5/25/2013 9:47:54	22	Male	4	No	saw none
5/25/2013 9:52:36	17	Female	2	Yes, up close (The part of the image that is below the blue line.)	The rotation (direction) of the grass blades., Coloring of the grass blades.
5/27/2013 14:08:19	21	Male	3	No	
5/27/2013 16:35:39	25	Male	3	No	
5/27/2013 16:44:48	27	Male	3	Yes, in between (The part of the image that is in between the red and blue line.)	The rotation (direction) of the grass blades.
5/27/2013 16:48:18	25	Female	4	Yes, up close (The part of the image that is below the blue line.)	The rotation (direction) of the grass blades.
5/27/2013	25	Female	3	Yes, up close	The rotation

16:49:15				(The part of the image that is below the blue line.)	(direction) of the grass blades., Positioning of the grass blades.
5/27/2013 16:52:10	28	Female	3	Yes, in the distance (The part of the image that is above the red line.)	The rotation (direction) of the grass blades., Coloring of the grass blades.
5/27/2013 17:01:22	21	Male	3	Yes, up close (The part of the image that is below the blue line.)	The rotation (direction) of the grass blades., Positioning of the grass blades., Height of the grass baldes.
5/27/2013 17:09:05	27	Female	3	Yes, up close (The part of the image that is below the blue line.)	The rotation (direction) of the grass blades., Height of the grass baldes.
5/27/2013 17:24:24	31	Male	4	Yes, in between (The part of the image that is in between the red and blue line.)	The rotation (direction) of the grass blades., Positioning of the grass blades., Coloring of the grass blades., Height of the grass baldes.
5/27/2013 17:26:40	30	Female	3	Yes, in between (The part of the image that is in between the red and blue line.)	Coloring of the grass blades.

5/27/2013 17:43:56	24	Male	2	Yes, in between (The part of the image that is in between the red and blue line.)	Positioning of the grass blades., Height of the grass baldes.
5/27/2013 17:48:49	23	Female	3	Yes, up close (The part of the image that is below the blue line.)	The rotation (direction) of the grass blades.
5/27/2013 17:56:33	28	Male	4	No	The rotation (direction) of the grass blades.
5/27/2013 18:37:19	25	Female	4	Yes, up close (The part of the image that is below the blue line.)	Coloring of the grass blades.
5/27/2013 19:54:34	23	Male	2	No	
5/27/2013 20:08:44	56	Male	5	No	
5/27/2013 20:15:27	20	Male	3	Yes, up close (The part of the image that is below the blue line.)	The rotation (direction) of the grass blades., Positioning of the grass blades.
5/27/2013 23:19:46	28	Male	3	Yes, in between (The part of the image that is in between the red and blue line.)	The rotation (direction) of the grass blades., Coloring of the grass blades.
5/28/2013 9:24:07	20	Female	3	No	

6. Should there be more variation in the color of the grass?	7. Should there be more variation in the height of the grass?	8. Should the width of the grass blades change?	9. Should the overall color of the grass blades change?	10. Should the overall height of the grass blades change?	11. What is the quality of the grass in close view?
No	No	No	No	No	3
No	Yes	Yes, they should be thinner	No	No	3
Yes	Yes	Yes, they should be thinner	No	No	5
No	No	Yes, they should be thinner	No	No	2
No	No	Yes, they should be thinner	No	Yes, they should be shorter	3
Yes	Yes	Yes, they should be thinner	No	No	2
Yes	Yes	Yes, they should be thinner	No	No	1
Yes	Yes	Yes, they should be thinner	Yes, it should be lighter	Yes, they should be shorter	2
No	No	Yes, they should be thinner	Yes, it should be lighter	Yes, they should be shorter	4
No	Yes	No	No	No	5
Yes	No	Yes, they should be thinner	No	No	2
No	Yes	Yes, they should be thinner	No	No	3
Yes	Yes	No	No	No	3
No	Yes	Yes, they should be thinner	No	No	2
No	No	No	No	No	3

No	Yes	Yes, they should be thinner	No	No	4
No	No	Yes, they should be thinner	No	Yes, they should be taller	3
Yes	Yes	Yes, they should be thinner	No	No	3
Yes	No	Yes, they should be thinner	No	No	2
No	Yes	Yes, they should be thinner	No	No	3
No	Yes	Yes, they should be thinner	No	No	4
No	No	Yes, they should be thinner	Yes, it should be lighter	Yes, they should be shorter	2
No	No	Yes, they should be thinner	No	Yes, they should be shorter	3
Yes	Yes	Yes, they should be thinner	No	No	3
Yes	Yes	Yes, they should be thinner	Yes, it should be lighter	Yes, they should be shorter	3
Yes	Yes	Yes, they should be thinner	No	No	3
Yes	Yes	Yes, they should be thinner	Yes, it should be darker	Yes, they should be taller	4
Yes	No	No	Yes, it should be lighter	Yes, they should be taller	4
No	No	Yes, they should be thinner	No	No	2
No	No	No	No	No	5

No	No	Yes, they should be thinner	Yes, it should be darker	No	3
No	No	Yes, they should be thinner	No	Yes, they should be shorter	3
Yes	Yes	Yes, they should be thinner	No	No	2

12. What is the quality of the grass in the distance?	13. What is the quality of the grass in between?	14. What is the overall quality of the grass?	15. Does the grass overall look realistic?	16. Would you have liked to see variation in density and grass species?	17. Other feedback
4	2	3	3	Yes, both	
4	4	4	4	Yes, both	"Q12 and Q13 would be 5/5 if FXAA or another AA is applied. Added flowers and moss and other stuff that usually grows with grass would be nice and add realism."
5	5	5	5	Yes, both	
5	5	4	3	No	
4	4	4	4	No	När man ser gräset väldigt nära ser det lite kantigt ut, men från avstånd är det snyggt!
4	3	4	3	Yes, variation in density	
4	4	2	2	Yes, both	
4	3	3	3	Yes, both	
4	5	4	3	No	
3	4	5	4	No	
2	4	3	1	Yes, both	"Grouping different varations of grass into seperate but large spots might help in increasing

					realism somewhat. Depending on what type of terrain one is looking for, it might be worth looking into. The grass shown in the far distance and up close does not look very nice, as it essentially gives the impression the landscape has grown spiky. The grass in the distance should be far less pronounced, perhaps replaced with something less eyecatching. The middle one looks great, though. While it perhaps lack some detail in some areas, I think continued work on this could yeld some very interesting results. I would also like to say that I wouldn't actually mind seeing this

					implemented in a game. As far as rendered grass is concerned, I like it."
3	4	4	3	Yes, both	The grass looks very good, but it doesn't look like real grass, that doesn't really change the fact that it looks really good though :D
5	4	4	3	Yes, both	myah
2	2	2	2	Yes, both	Be-leaf in your grass!
3	4	3	2	Yes, variation in grass species	"To look more realistic grass should be bigger on flat fields and smaller on hills, perhaps follow the terrain normal. Hard to see how big the grass actually is, should have something compareable next to it. Is there sunlight on the grass? Is there sunlight on the terrain? I can't tell see it, having those will make it look

					more realistic.
2	2	3	3	Yes, both	
4	5	3	3	Yes, variation in density	
4	5	4	3	Yes, both	
5	5	4	2	Yes, both	
4	4	4	3	Yes, both	
2	5	4	2	Yes, both	
4	3	3	2	Yes, variation in density	
5	5	4	3	Yes, both	
4	3	3	2	Yes, both	
4	3	3	2	Yes, both	
2	4	4	3	Yes, both	To make it more lively, I would've added some brown colours to it as well, real grass is seldomly just green.
5	4	5	3	Yes, both	
3	5	4	3	Yes, variation in grass species	
3	3	3	3	Yes, variation in density	
4	4	5	5	No	
4	4	3	2	No	
3	4	4	3	Yes, variation in grass species	
4	3	4	3	Yes, both	