Freestyle Assignment : Snow Scene

Julian Spittel

December 21, 2018

Contents

1 Introduction **2**

2 Rendering Techniques **2**

3 Implementation Details **2**

3.1 Grading 3

1 Introduction

The main goal of my freestyle assignment is to implement a snowy winter scene.

To achieve this I will improve the already existing project from the last assignment by adding vertex displacement. Vertex displacement will simulate snow that fell on the terrain. This is done with an heightmap representing the offset in positive y-direction(upwards). This also leads to a new normal of the terrain which needs to be calculated and handled. Snow will only fall above a variable height.

The full geometry trees in the scene will additionally be gradually colored whiter depending on the vertex normal and height of the tree/vertex.

The already existing distance based terrain tessellation will also be used for height displacement and normal calculation. This will optimize the rendering time.

Another new feature will be the implementation of a very basic collision system that will be used to leave footprints in the snow as the camera walks across the terrain.

The footprints will gradually fade out as new snow falls that will (re-)fill the heightmap. Additionally, the falling snow will be displayed by a basic particle system rendering falling snowflakes of random size and position. These particles will slowly fall towards the ground and will be affected by a random velocity displaying the wind. Particle amount and refill rate will be dependent on the current height.

2 Rendering Techniques

Tessellation was already used in the previous assignments and will be used again for the heightmaps. The tessellation evaluation shader will now additionally add the vertex displacement and update the normal calculation. Vertices that are moved by the vertex displacement will be colored white (maybe by using a texture). However, the standard lighting equation (Phong) will be used.

Vertex displacement is commonly used technique in physically based rendering. It uses a texture to define how to move a vertex in model-space within a shader. Snow fall will depend on the terrain slope and height. [1]

Snowflakes will be rendered with a textured quad displaying a given texture. Additionally snowflakes will only be visible within a certain range from the viewer. This will dramatically increase performance. The flakes will be affected by wind blowing with random strength and direction. [2]

3 Implementation Details

The heightmap requires multiple new textures that will be packed into a texture array and mapped onto the terrain.

The particle system will surround the viewer with a given cube and moving will also move through the particles. Particles that will fall out or will be left

outside of the sphere will be set to the opposing side of the cube. To use instanced drawing a new buffer for the particle data must be allocated.

The implementation of footprints will use a mask that will be applied to the vertex displacement heightmap on a specific position. Multiple textures in a texture array will be needed to provide a reasonable resolution to actually view footprints.

3.1 Grading

Points	Task
4.0	Implement vertex displacement with multiple heightmaps on terrain.
2.0	Texture the snow and calculate new normals in the tessellation evaluation shader.
2.0	Footprints in the snow will be left while walking over the terrain and they will refill over time.
2.0	Adjust snow height and refill rate depending on the height.
3.0	Create a snowflake particle system that follows the viewer, moves particles and repeats them wi
1.0	Let the number of particles depend on the current height.
1.0	Move particles with random wind.
1.0	Texture trees white depending on the vertex normal and the height.
1.0	Create a user interface to adjust values for vertex displacement and the particle system.

References

[1] Per Ohlsson and Stefan Seipel. Real-time rendering of accumulated snow. *Proceedings of SIGRAD*, 01 2004.

[2] Changbo Wang, Zhangye Wang, Tian Xia, and Qunsheng Peng. Real-time snowing simulation. *The Visual Computer*, 22(5):315–323, May 2006.