Real-time shader-based rendering of grass

Kin Liu
Manuel Reinfurt

Abstract

The grass is the best.

CR Categories: I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Display Algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Radiosity;

Keywords: radiosity, global illumination, constant time

1 Introduction

Best

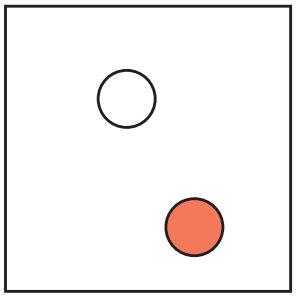

Figure 1: *Sample illustration.*

2 Methods

The pipeline is set up very easily. For a realistic scene, we also need terrain.

CPU
The CPU will generate the root points out of the density map and create a single vertex buffer, that contains all roots. It will also slice the full grass grid into smaller patches, which can be controlled through constant buffers on the GPU.

Since we also have static terrain, the root points will already contain the correct displaced Y position.

Vertex shader
The vertex shader is, in essence, a pass-through shader.

Geometry shader
All the hard work is done in the geometry shader. Since the geometry shader gets a single point as an input, it's job is to create a grass blade with a specified number of vertices. The geometry shader will also take care of calculating normals, level of detail, animation.

Pixel shader
To calculate the color, we use the basic Pong BDRF, combined with a texture and a randomized tint.

3 Randomization

The scene looks very artificial if the grass is laid out with exact distances, when each grass blade has the same tint, height or width. However, it is fairly difficult to generate random numbers on the GPU, which is why we have to use certain tricks to have fairly random grass properties. When laying out the root points, we can generate random positions using the CPU. Since these points are given to the geometry shader, and we kow that these positions are random, we can generate a random number between -1 and 1 using the following equation.

$$r = sin(\frac{\pi}{2} * frac(root.x) + \frac{\pi}{2} * frac(root.z)) \qquad (1)$$

There is still a problem, though. Since we now only have one random number, the grass properties will be related to each other. If the grass is a bit higher, it will also be a bit browner.

INSERT non-randomized and randomized picture

4 Animation

Simulating wind.

5 Wheat differences

It's not about grass, it's about sending a message.

6 Benchmarks

FPS with different machines, root counts, level of detail settings and so on. Grapshs!

7 Conclusion

As you can see, shaders are... best.