Increasing realism of animated grass in real-time game environments

Benjamin Knowles,[*] Oleg Fryazinov[†]
The National Centre for Computer Animation, Bournemouth University, UK

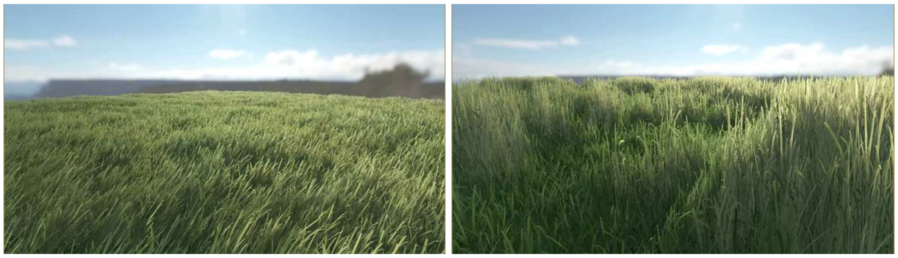

Figure 1: *Resulting grass animation in a real-time game environment.*

1 Introduction

With the increasing quality of real-time graphics it is vital to make sure assets move in a convincing manner otherwise the players immersion can be broken. Grass is an important area as it can move substantially and often takes up a large portion of screen space in games. Animation of grass is a subject to academic research [Fernando 2004; Perbet and Cani 2001] as well as recent examples implemented in games. The list includes, but is not limited to, games such as Far Cry 4, Battlefield 4, Dear Esther and Unigine Valley. Comparing video game assets with reality, it can be seen that the current methods have a number of problems which decrease the realism of the resulting grass animation. These problems include: 1) the visible planar nature of grass geometry and 2) problems with the grass movement which include over-connectivity of grass blades in respect to their neighbours, no obvious wind direction and exaggerated swaying motions.

In this paper we propose to increase realism of the grass by focusing on its movement. The main contributions of this work are: 1) Distinguishing ambient and directional components of the wind and how they interact and 2) The method for calculating directional wind by using a greyscale map and wind vector. The grass was implemented with vertex shaders in line with the majority of methods described in academic literature (e.g. [Fernando 2004]) and implemented in modern games.

2 Our method

The grass asset was modelled with some bending and distorting to the faces to lessen the issue of revealing obvious planes. It is procedurally replicated in space with additional modifications of height driven by a tiled greyscale map in the vertex shader. As in current state of the art papers, the grass is animated in the vertex shader by using a displacement function. The displacements are only applied to the vertices that are higher up in UV space with a smooth falloff. In this work we distinguish directional wind and ambient wind. Both are driven by texture maps that are panned and rotated in world space.

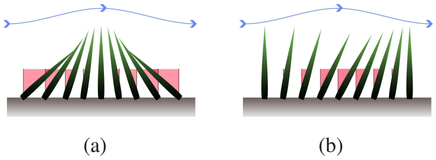

Figure 2: *a) Grass behaviour from normal map, b) Our method;*

The directional displacement component is limited to the wind vector. Instead of using a normal map or algorithm to move the grass in all directions (see figure 2a) - established methods - we are using a greyscale map and multiplying its intensity by the wind vector (see figure 2b). This method requires an additional function to calculate the height of the grass. The ambient displacement component uses a normal map to add noise to the movement, lessening the effect of movement being locked to a vector. Its intensity is linked to the directional wind so as the wind blows stronger more noise is applied. This component can also be used to resemble the movement of grass under no strong wind.

The shader developed can be applied to any mesh providing its UVs are suitable and the maps and parameters of the wind are controllable by an artist to achieve the desired effect. The grass was implemented by using Unreal Engine 4 and is rendered with 60+ FPS on the patch of land and on 30+ FPS on the full landscape. The results show to be an improvement on what games are currently offering. No level of details were used grass so potential for further optimisation exists.

References

Fernando, R. 2004. *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. Pearson Higher Education.

Perbet, F., and Cani, M.-P. 2001. Animating prairies in real-time. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, ACM, New York, NY, USA, I3D '01, 103–110.

[*] e-mail:benmcknowles@gmail.com
[†] e-mail:ofryazinov@bournemouth.ac.uk