FurShader

De Input:

Texture2D m\_TextureDiffuse;

Texture2D m\_TextureFurDensity;

Texture2D m\_TextureMask

float m\_Gravit;

float m\_FurLength;

float m\_FurUV;

De FurShader heeft 3 textures en 3 variabelen. De m\_TextureDiffuse is de normale diffuse texture van het object. De FurDensity is een texture die enkele bolletjes bevat. Deze zorgt voor de rangschikking van de haren. En de laatste texture is de mask en hiermee kun je kiezen welke stukken van de texture meer of minder haar bevatten.

Dan zijn er nog de drie floats. De gravity zorgt ervoor dat de haren naar beneden hangen, deze moet negatief zijn anders staan ze naar boven. De furLength zorgt voor de lengte van de haren en de furUV zorgt voor meer of minder haren.

struct GS\_DATA

{

float4 Position : SV\_POSITION;

float3 Normal : NORMAL;

float2 TexCoord : TEXCOORD0;

float ShadowStrength : FLOAT;

};

De GS\_DATA heeft nog een extra variable ShadowStrength. Dit is belangrijk omdat de geometry shader alle lagen in 1 keer maakt en de pixelshader moet weten welke laag donkerder moet worden gemaakt.

De Base geometry shader:

Er zijn 2 renderpasses in de FurShader. De eerste is voor de basis te maken zonder haar en de tweede is voor de haren zelf. Dit is omdat ze niet dezelfde geometry shader kunnen gebruiken. De BaseRenderer is de geomerty shader voor de basis van het model.

Deze geeft enkel de vertex informatie door en maakt de mesh opnieuw in de geometry shader en geeft een shadowStrength door van 0.5f waardoor de basis donkerder wordt gemaakt.

De Fur geometry shader:

De haren die worden gemaakt zijn eigenlijk allemaal lagen waar er bolletjes op worden getekend. Dus worden er 33 lagen gemaakt in de geometry shader. Elke vertex wordt volgens zijn normaal een beetje naar buiten verplaatst om op die manier de lagen te krijgen.

float length = ((float)i+1) \* m\_FurLength/maxLayers;

v1 = vertices[0].Position + vertices[0].Normal \* length;

Hierna zetten we er wat gravity op. Hiervoor moet je eerst de vertices in world space zetten. Met deze formule worden de haren met elke laag meer naar beneden getrokken als een parabool maar rekken ze wel een beetje uit. Daarna worden de vertices gemaakt op de geometry shader.

float3 g = float3(0,m\_Gravity,0);

float4 nV1 = mul( float4(v1,1) ,m\_MatrixWorldViewProj);

float4 h = float4(g,0)\*pow(length,2)\*i/maxLayers/m\_FurLength;

nV1 = nV1 + h;

De main pixel shader:

Er zijn naast 2 geometry shaders ook 2 pixel shaders. De main shader zorgt voor de texture van de basis vorm. De mask texture zorgt ervoor dat het onder de haren iets donkerder is.

De fur pixel shader:

Deze shader tekent enkel de bolletjes uit de m\_TextureFurDensity maar met de kleur van de diffuse texture. En die kleur wordt vermenigvuldigd met input.ShadowStrength waardoor de haren donker zijn naar binnen toe. Voor de TextureFurDensity worden de uv’s vermenigvuldigd met m\_FurUV om meer haren te maken. De mask wordt met de alpha van de density texture vermenigvuldigd om enkel op de witte delen van de mask haar te krijgen.

float alpha = m\_TextureFurDensity.Sample(samLinear,input.TexCoord\*m\_FurUV).a;

float mask = m\_TextureMask.Sample( samLinear,input.TexCoord ).r/3 + m\_TextureMask.Sample( samLinear,input.TexCoord ).g/3 + m\_TextureMask.Sample( samLinear,input.TexCoord ).b/3;

alpha \*= mask;

float3 color = m\_TextureDiffuse.Sample( samLinear,input.TexCoord ).rgb;

color = color \* input.ShadowStrength;