***Direct X Homework 4:***

***Growing Grass***

**Objective**

Today’s homework will cover the topics talked about in lecture. Specifically, we’ll be using the geometry shader to make a field of grass that can grow and be modified with the arrow keys.

**Homework Requirements**

Features you’re expected to implement:

* Grass must grow, and change width based off of arrow key input.
* Grass grows from a simple textured ground rectangle.
* All geometry is handled by the geometry shader.

**Homework Instructions**

**C++ code**

* In the C++ side of things, you’ll be handling the input, as well as things like position direction of grass growth then passing this along to be used in the geometry shader to create the geometry shader side.
* The position and direction can be stored in a vertex structure to be accessed later in the geometry shader.
* These two properties will be used for the position and Y direction of the grass blade’s world matrix, which you can create in the geometry shader.
* Other things we’ll need to pass are the grass width, and grass height.

**Geometry Shaders**

* Start Simple, take each incoming point and convert it into a triangle. After you get this working, then you can worry about getting your grass to work like the example.
* There will need to be two geometry shaders, one for the ground (which creates the ground quad) and one for the blades of grass. They will also need their own vertex and pixel shaders as well.
* For the grass geometry shader, we’ll be creating segmented blades of grass that we’ll scale and rotate.
* Each blade of grass is unique in that it is facing differently, has a possible different number of segments, and has a different curve than other blades of grass. If we pass our position and direction in the input structure as float4s we can take up the extra value for a scale (0-1 value that can be multiplied by the number of segments to give us a varying number of segments) and a curve value (the demo uses a random range from -4 degrees to -3 degrees).
* The grass is pretty simple in concept. What we’ll want to do is create a series of segments (the example uses 8 for a max value but this is modified by the scalar value mentioned above to give us a varying number of segments per balde). With our segments defined, we want to use the direction and position passed in to create the world matrix for each blade of grass by hand. With that made we’ll  need to use the curve (Remember passed as the w of the direction) value along with the length of a single segment (grassHeight/numberOfSegments) to create a transformation matrix that translates in the (local) y, and rotates slightly. For each segment, you’ll want to multiply the transformation matrix with the world matrix then apply it to the current segment’s vertices (Don’t forget to multiply it by the viewProjection before you send it!). What we’re trying to do here is make it so each segment is evenly spaced (segment 1 goes from 0-5, segment 2 from 5-10, etc) by multiplying the world matrix multiple times by our transformation matrix.
* Each segment is made up of 4 triangles. Another thing to remember is this is a strip so you’ll need to take that into account when you create the verts for your blade of grass.
* Try drawing it out on paper, then get that rendering, THEN try to get it growing. Don’t do everything at once.