Sam Skolnekovich
11/6/16
Project Final

## Things you need to know:

You can freeze the application by going off the map and trying to turn. Don't do this.

**Usage:**
r =switch to rain/fog scene
s =switch to snow scene
w= move forward
W=move backward
arrow keys= look around
y= lower fog density

Y= lower fog density more
t= take fog density back to default value
H = switch off help statement
h = switch on help statement
x = get rid of all grass polygons. (grass polygons expensive computationally, will go into more detail later)
a= low number of grass polygons in center of map
d=medium number of grass polygons in center
g= fill map with grass polygons (slow)
P= more total particles ~90000
p= less particles - 10000
, = slow down particles
. = speed up particles

This will build the program:

```
make clean
make
./FinalProject
```

## Things I would like you to notice:

I made different textures depending on the scene you are in. If you go into snow you will notice the grass has a different texture applied. Along with different ground and particle effects.

There are a ton of controls(including fog density because it's set pretty dense!) and lot of them can add pretty interesting changes.

The fog/rain scene is supposed to simulate a very subtle water particle effect. It should be barely noticeable. The idea is that these are small water droplets forming in the fog. The snow scene has a faster downward motion.

Sam Skolnekovich
11/6/16
Project Final

There are three different tree obj's and the fog color changes based on which scene you are in.

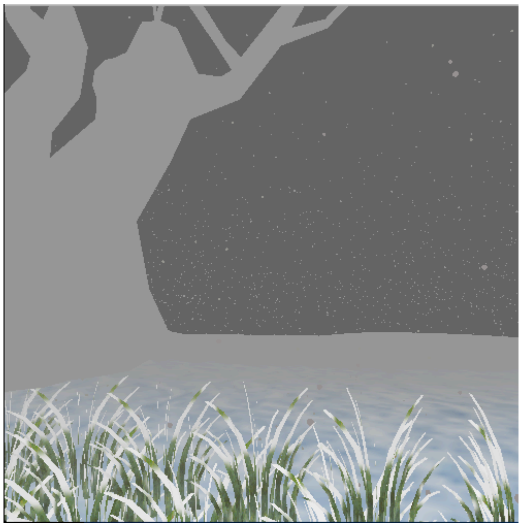
I am proud of the camera. I think it works very well. You can look up and down and the particles actually tilt correctly even though this is too small of a change to notice. The z value of the camera is tethered to the dem height map. So it should simulate you walking along the plane.

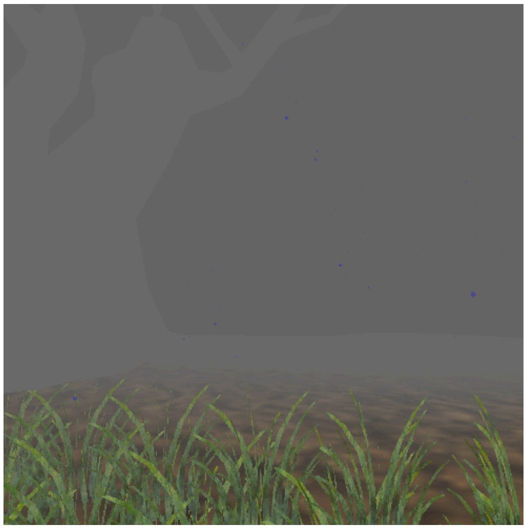
## Things you should know:

I added the sphere function into my code for the particle shapes and it ran slower. I am planning on following up on this with you. I do not know why but glutSphere ran ALOT faster on my machine. I may just have a really out of date gpu.

The actual controls for the particle systems I adapted from a github repository. You can find all of my sources at the bottom of the c file and this readme.

**Things you should see if this rendered correctly for you:**





Problem I had:
I could not load my alpha grass textures. SOIL and a few other solutions just did not work for me. I tried converting it to 32 bit bmp and alpha blending on a white background as well. You can see more of this problem in my project review readme. So instead I used a grass obj. It is computationally very expensive. That is why I give options on how many grass polygons you render.

Sam Skolnekovich
11/6/16
Project Final

Resources:
Besides the tutorials, I did get help with obj loading, particle systems and initiating fog with code. I used the glm.c .h program. I also found a rain simulation and adapted it.

```
/*
*  Resources:
*
*  http://www.lighthouse3d.com/tutorials/glut-tutorial/keyboard-example-moving-around-the-world/
*
*  OBJECT LOADER glm.c glm.h
*
*  http://www.swiftless.com/tutorials/opengl/camera2.html
*
* http://www.swiftless.com/tutorials/opengl/fog.html
*
*  http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/
*
*  RAIN SIMULATION https://gist.github.com/thaenor/4d9531cc9a7d1c34b998
*
*  Taken work from ex17 ex7 from class.
*
*  All of the obj's were free models, usually from turbosquid.
*
*
*/
```