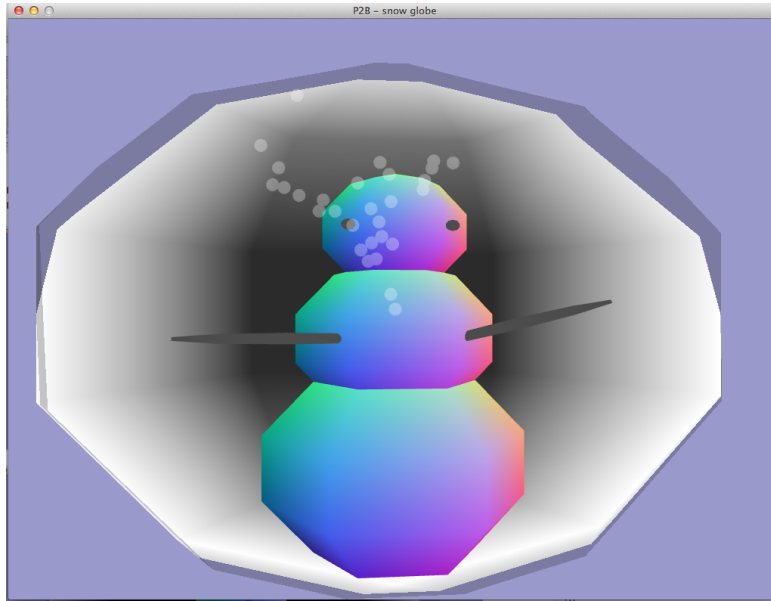


Program 2B – CSC/CPE 471 Snow Globe



Due Wednesday, October 26th, 11:59pm. (7% of your final grade)

The goal of this assignment is to practice:

- using transforms in order to create a more complex scene from one simple input mesh, specifically a sphere
- using vector math to control rendering
- passing data to our shaders

For this assignment, you must build an animated scene of a snow globe. Consider starting from your lab 5 code. The tasks required for this assignment include:

1) Building a complex model from an input sphere. The model must include at least seven copies of the sphere, each, which are, transformed differently than the others. For example, the snowman above was built with 7 spheres. Your transforms must include:

- One non-uniform scale
- One rotation
- One animating element (that animates with respect to the other parts of the model – you will likely want to look at the hierarchical modeling slides and lab for this component of the program).

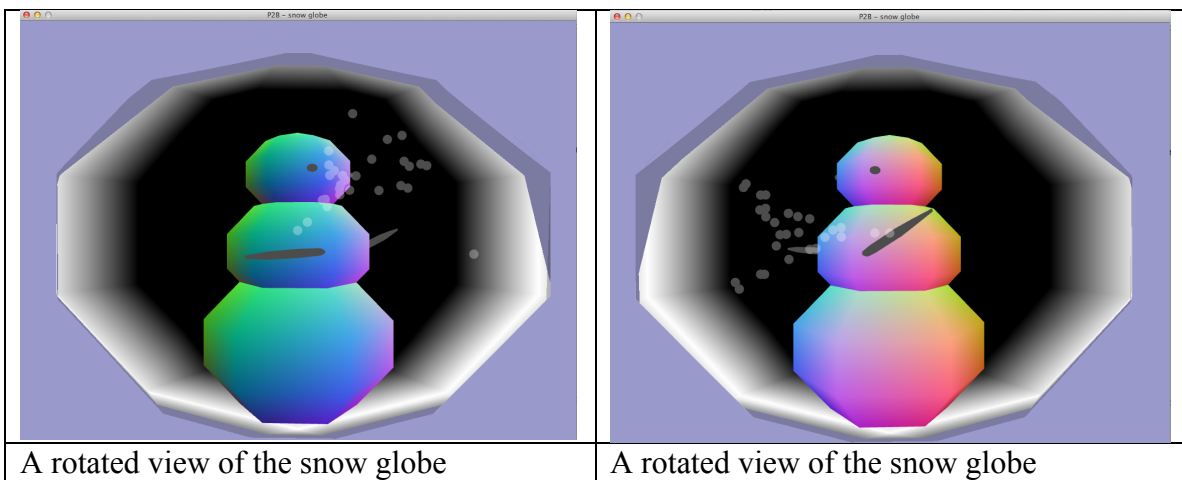
Please color your model using the normal as done in lab 5. An example sphere.obj file with normal will be provided. Make your scene look interesting.

2) Encapsulate the entire scene in one large sphere. You will need to write a shader such that most of the front facing fragments are not drawn (in order to simulate looking into the snow globe), however the back faces must be drawn with some color. Consider col-

oring them such that the fragments that are further away are darker. Play with the shading and your vector math in order to make the snow globe look as realistic as possible. For the snowman example above, I kept some of the front facing fragments but colored them with a transparent component.

3) Add keyboard events for the “a” and “d” key which rotate the entire snow globe, so that the user can look at the model from different angles. Make sure to update the normals in order to correctly cull out the faces that now “face” the user.

4) Add snow to your globe by integrating your program 2A code with some modifications to make the points travel in a circle from their current location. Use the parametric equation for a circle (and uTime) in order to animate the snow. Add the snow with a different shader (vertex and fragment).



Percentage point break down:

- 30% animated model
- 20% snow globe render
- 20% user interaction (global rotate and correct render update)
- 20% snow
- 10% general look of model and code sanity