

Sam Erie  
serie@alaska.edu

Aisha Peters  
ampeters@alaska.edu

space.html builds on previous homeworks and the topics were a star particle system and procedurally generated textures on planets in the scene.

The planets textures consist of a heightmap for the terrain, and a second heightmap for the atmosphere. planetShader handles terrain, and atmosShader handles atmosphere. Heightmaps are made using perlin noise library written by Dr. Chappell. Terrain uses fnoise, atmosphere uses 1/f noise with greater frequency to give the clouds a more realistic appearance. The heightmaps are mapped to colors in their respective shaders. Planets are mapped with a mix from dark blue up to lighter blue on range [0., 0.5], then dark green up to lighter green on range [0.5, 1.]. Atmospheres are all white, with an alpha channel on the range [0., 1.]. This required the use of blending when atmosphere is drawn, which requires enabling blending, setting blend function and stopping from writing to the depth buffer.

Useful references for planets, are their shaders, function drawPlanet(planet), function genHeightMap(size), and other nearby functions depending on your level of curiosity.

The star particle system borrows from code written by Dr. Chappell to a draw white, fuzzy texture on squares and use the alpha channels to make them appear round. These "star" particles are then generated at random locations between -50 and 50 in the x, y, and z coordinates at random sizes. A random velocity and direction of travel is applied to each star in the idle function, with set bounds at 50 and -50 that move traveling stars to the opposite side of the scene to prevent all stars from disappearing over time. The depth filter is not applied to stars, thus they will not overlay any other object in the scene regardless of location. The number of stars generated is managed by the variable "starNum", and their rate of movement by "velfact". Modifying either of these variables will alter the respective attribute.