

## Planet Universe!: A Procedurally Generated Java Game

Christine Lam, '22

Planet Universe! is a space themed Java game that uses procedural generation—the generation of data pseudo-randomly using computer algorithms. When launching the game, the user is brought to a standard launch menu where they can either start the game, see the credits menu, or exit the game. After clicking play, the screen scenery transforms into procedurally generated stars and nebulas.

All the current and future components such as the stars and nebulas are rendered in the same fashion: a class that manages a variety of abstract world tiles that come together to create a layer of the world. Randomly choosing from a list of star sprites, the program adds the stars to a world tile. World tiles come together to form a star layer which are also randomly layered on top of each other at different distances to create a depth effect. There are 128 pixels in each tile and the program generates and stores the tile if it's within twice the viewport width/height around the user's position in order to prevent the generation of unnecessary tiles.

To create the nebulae, a gradient noise was used to generate a visually appealing texture of the gas. A color palette is passed in to utilize its RGB color bytes to determine the color of the nebulae. This color palette is a 256x4px image of a color gradient

Using blend of simplex and Perlin noise, 12 gradients with values of only 0, 1, and -1 were distributed around the unit circle to generate smooth gradients. To create world tiles containing nebulae gas, a gradient buffer recursively evaluates the gradient functions to eventually return a float. This float is used as an image buffer value to determine the color and transparency taken from the color palette.

Creating the nebulae was super challenging. Getting a good understanding of noise was hard, as many sources I looked at contradicted each other. After finally finding a reliable source, I attempted to use the algorithm myself. The first attempt at creating the nebula was a failure as the nebula tiles were each the same and one could see where each tile started and ended. The tiles did not smoothly blend together like I had planned. Currently, I am working to fix the noise algorithm.

The end goal is a game that procedurally generates a space themed environment that includes stars, nebulas, and planets. The user would be able to move a spaceship in order to collect an infinite amount of unique planets. My next step would be to finish the generation of the nebula and create the planets, whose generation will be more straightforward as it will utilize a lot of the already made classes for star and nebula generation.

Peter Mawhorter

Computer Science

Funding source(s): Barbara Peterson Ruhlman '54 Fund for Interdisciplinary Studies