

Exploring a 3-D World

Computer Systems Lab 2008-2009

by Zachary Greer

Abstract

The program displays a window which allows the user to explore an infinite 3-D world using OpenGL. A grid of spheres are provided as reference. Mouse input is taken to modify the direction in which the user is looking, and keyboard input is taken for movement across a plane. Movement occurs with respect to the direction the user is looking along the xy-plane. Spherical coordinates are used to achieve this intuitively and easily. This is a framework intended to be extended into a full world by someone who wants to load models into it. All variables are set up to be easily changed to fit the needs of those who want to use it.

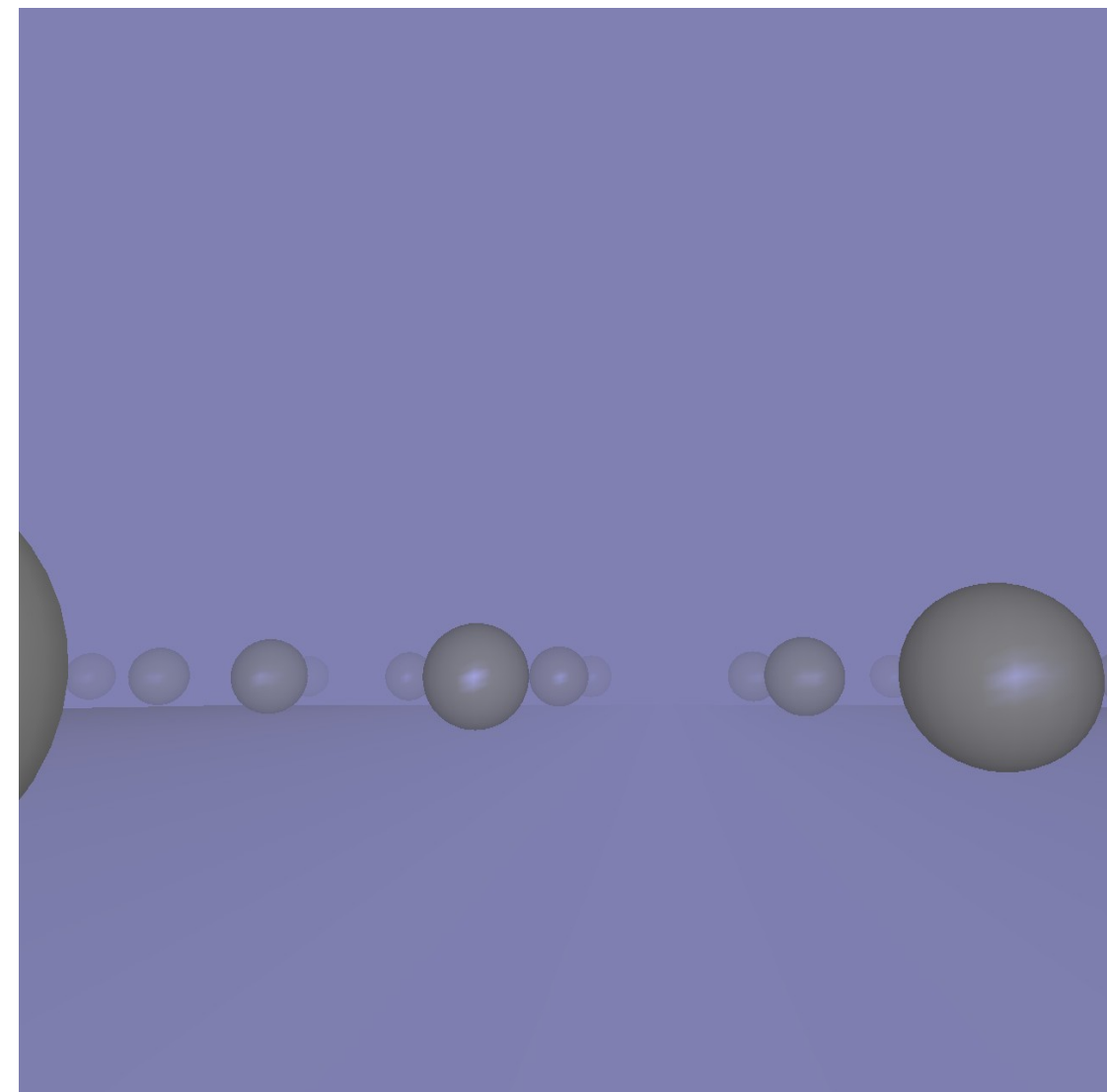


Fig. 1: A screenshot of the program running

Perspective and Spherical Coordinates

The first real issue presented to one trying to simulate a person exploring an environment is that of perspective. OpenGL handles some of that, in that it will look from one point towards another point. Still, how do we best represent this? We know the vector in which the player is looking, and the player's position. With the radius equal to 1, the spherical format can easily describe a point to look at. In (x,y,z) format, the point is $(x_position_of_player + \cos(\theta)\cos(\phi), y_position_of_player + \sin(\theta)\cos(\phi), z_position_of_player + \sin(\phi))$. Advantages of this: It translates easily to input (mouse motion in the x and y directions change θ and ϕ respectively), and the entire line of sight vector and motion vector can be described with two variables. These are quickly modified and also quick to produce either vector, as can be seen in Fig. 2.

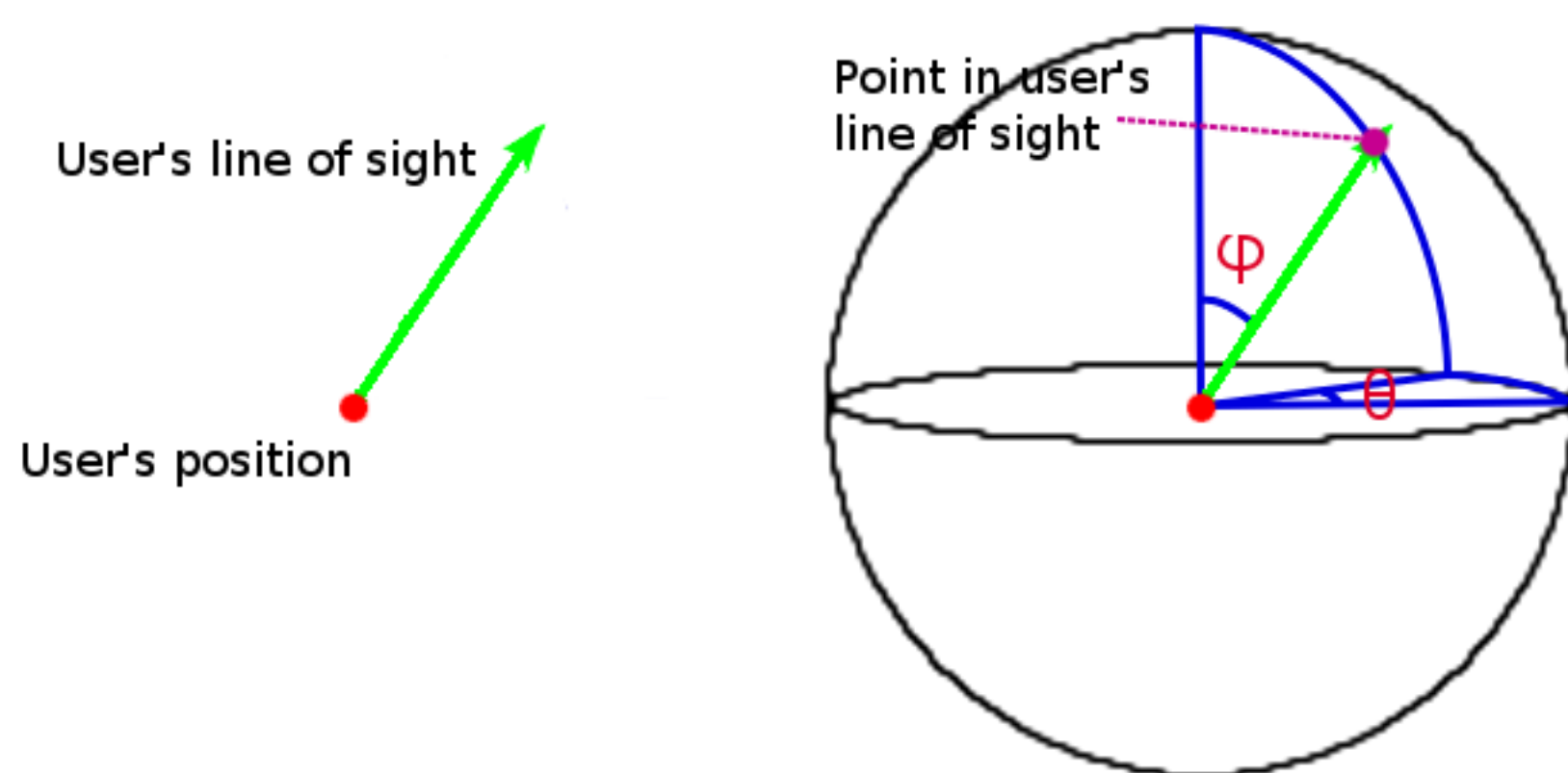


Fig. 2: Demonstration of spherical coordinates

Extendability

The program is equipped with variable names that are intuitive, and is commented and organized in such a way that the program is easily changed to fit one's needs. This software can be included in a larger program that causes models one loads to move, and the program will display the new situation with no added code. This program essentially handles all of the graphics problems one would run into.

Lighting and Fog Effects

The program handles all of the difficulties of setup and intricacies of fog and lighting. Anyone who wants light need only determine what color it is, and where they want it. OpenGL supports up to eight lights, so that's what the program can do. All lighting acts very similarly to how it would in the real world, as Fig. 1 displays. Fog is difficult to use properly due to many different choices, but the program includes all of them so that one may pick which one to use. Other than that, it reduces fog use to being a simple variable: intensity.