Exploration of a 3-D Environment

By Zachary Greer

Introduction

Here's what it is.

Abstract

- Allows user to explore 3-dimensional world
- First person
- Mouse and keyboard input
- Fog and Lighting

Inspirations

- Proximity-based chat system
  - First employed in first-person shooters
  - Enhanced social element greatly (realism, intuitiveness)
  - Could be used for just a chat client
  - Could also be used for other teamwork activities
- Gaming experience
  - Explains the elements of an enjoyable game
- Google Lively, IMVU are 3-D chat clients, but are neither proximity nor audio.

Goals

- Realistic physics
- Loadable surface system
- Networked, first-person graphical chat system
- Chat: audio, based on proximity (modeling real life)
- Game element? (Maze? Capture the Flag?)

Development

Here's how it works.

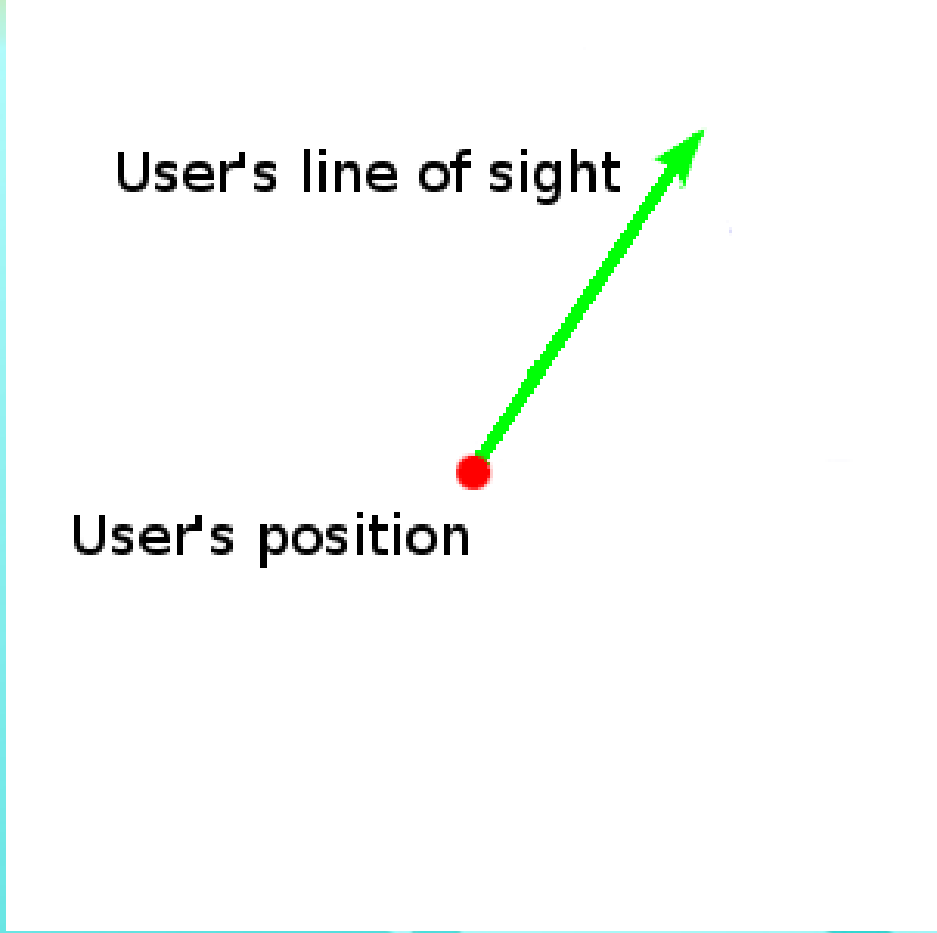
C, OpenGL, MPI

- Why C?
  - C is a powerful and fast language
  - OpenGL is a 3-D graphics library compatible with C
  - C does not change its syntax. C code is always valid
  - MPI is an interface to pass information over a network and is compatible with C
  - I have had experience and am familiar with all of these.

Perspective

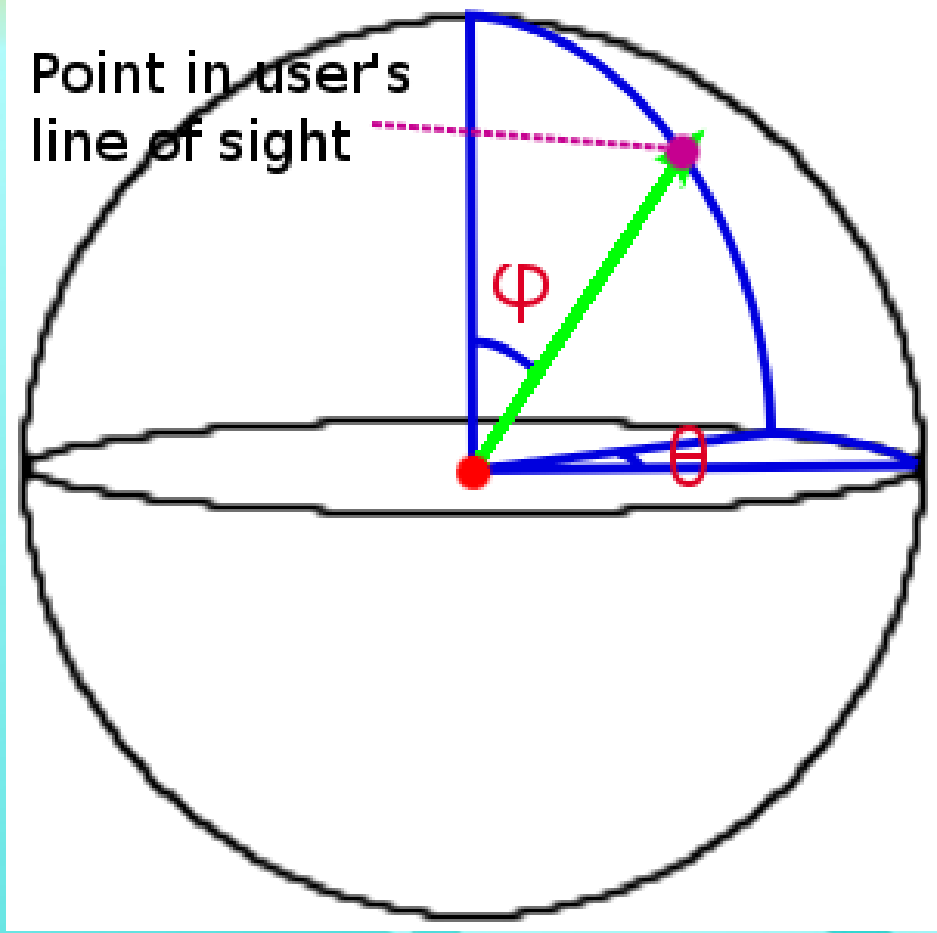
- Spherical coordinate system
- Angle measurements in radians
- Mouse control of looking around
- Mouse recenters to the middle of the screen, so the screen does not limit where you look.

Why Spherical?



- OpenGL provides a command to look at a certain point from another point
- User's position is easy, but how to give point on line of sight?

Why Spherical?



- Apply a spherical coordinate system, and this is simple
- In (x,y,z) format, the point is (x_position_of_player + cos(θ)*cos(φ), y_position_of_player + sin(θ)*cos(φ), z_position_of_player + sin(φ))

Why Spherical?

- Mouse system intuitive, looking up or down changes φ, left and right changes θ
- Movement simple, obtaining a unit vector in the plane of horizontal movement just <cos(θ), sin(θ)>

Movement

- Movement occurs with respect to the direction the user is facing
- Movement is controlled with keyboard input
- The unit vector in the direction of the line of sight projected onto the xy plane is multiplied by speed, added to position. The result is motion.

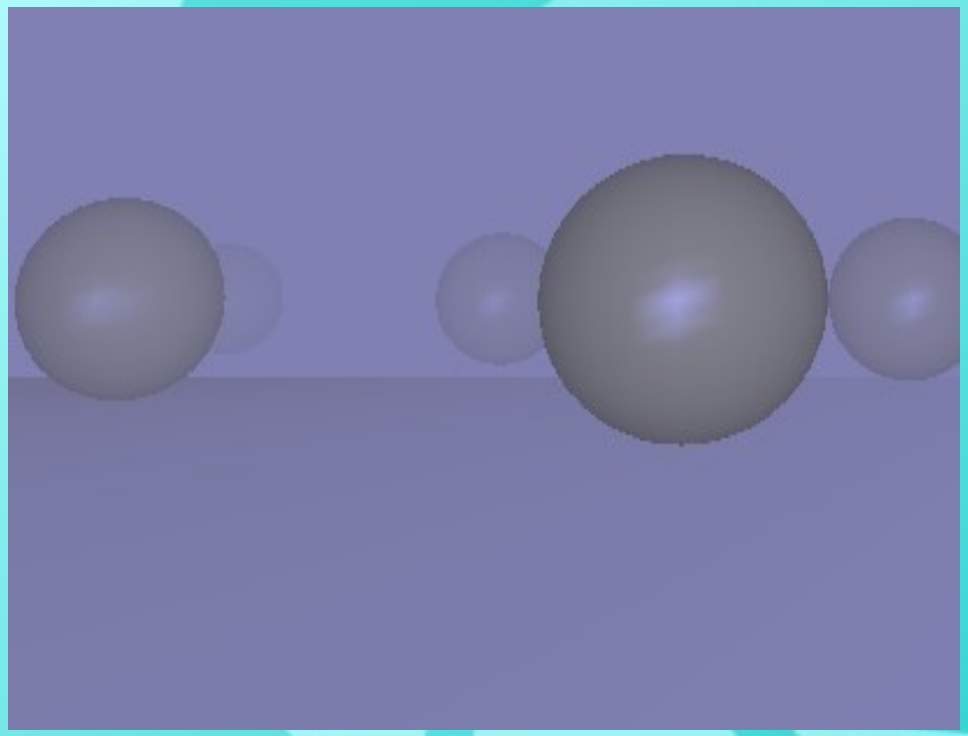
Working with X11

- Problems encountered:
  - Screen size is not a simple thing to find
  - Little documentation on how X functions work in C
  - Mouse warp works inconsistently
  - Very careful of how many connections you open
- Mouse warp now works every time, as I found out how to check if it's warped. So now, despite inconsistency, I can do it until it works.
- Pieced together most X11 from forums.

Results

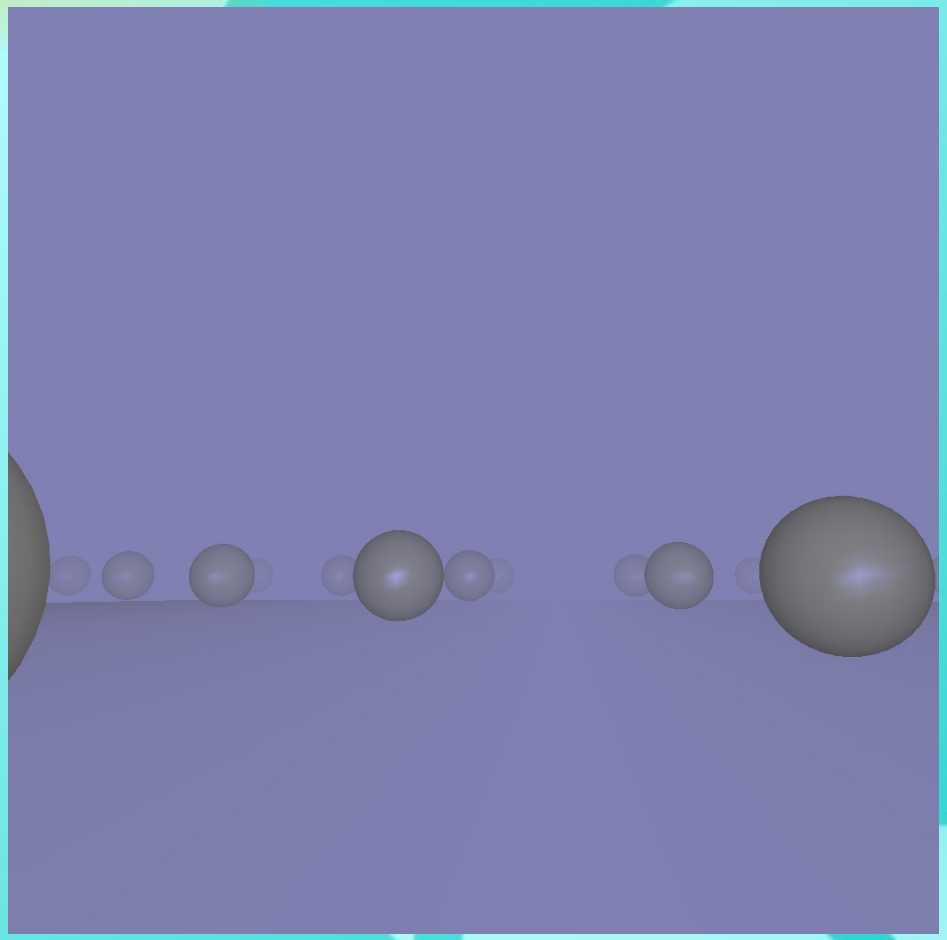
Here's what it does.

Current State



- The user may move along the xy plane
- A grid of spheres are provided for reference
- A floor and horizon are displayed
- Realistic lighting and fog

Current State



- The user may move freely throughout the environment with arrow keys
- The user may use the mouse to look wherever they wish
- Speed can be changed with +/- keys.

Current Obstacles

- OpenGL is *very* unkind about loading images
- No libraries for audio seem to exist for C...
- Networking over internet is not trivial

Changes of Plan

- Originally meant to be a simple first-person shooter
- Now the plan is a framework for networked chat system/maybe puzzle game/team activity
- Of these, puzzle game looks most feasible