First-Person PacMan
by Brett Jones
TJHSST Computer Systems Lab 2007-2008

Abstract

The purpose of this project is to create a 3D, first-person version of the classic PacMan arcade game in order to learn more about the concepts of 3D graphics programming and rendering algorithms. The project will also include a basic AI to control the ghosts.

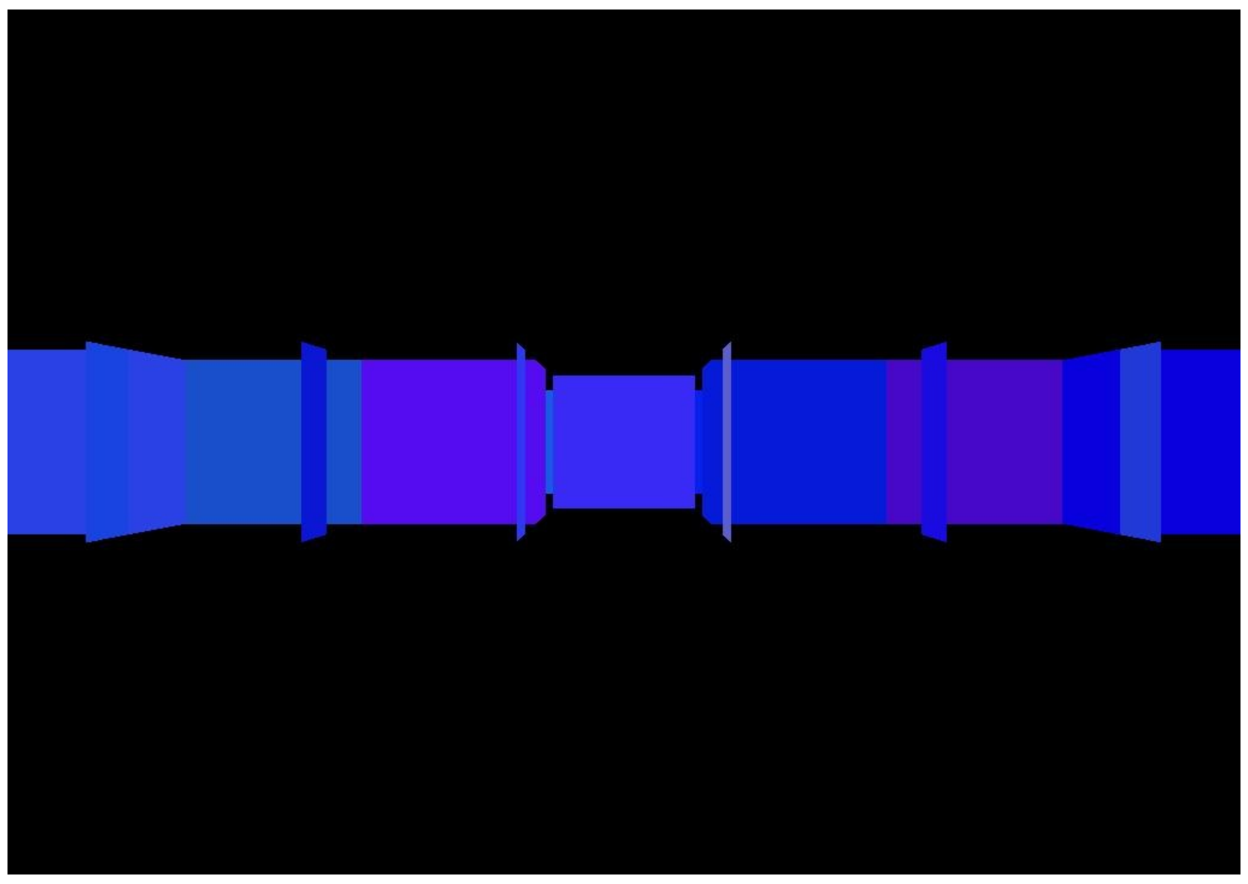
An initial view of the scene, slightly raised and with wall colors differentiated.

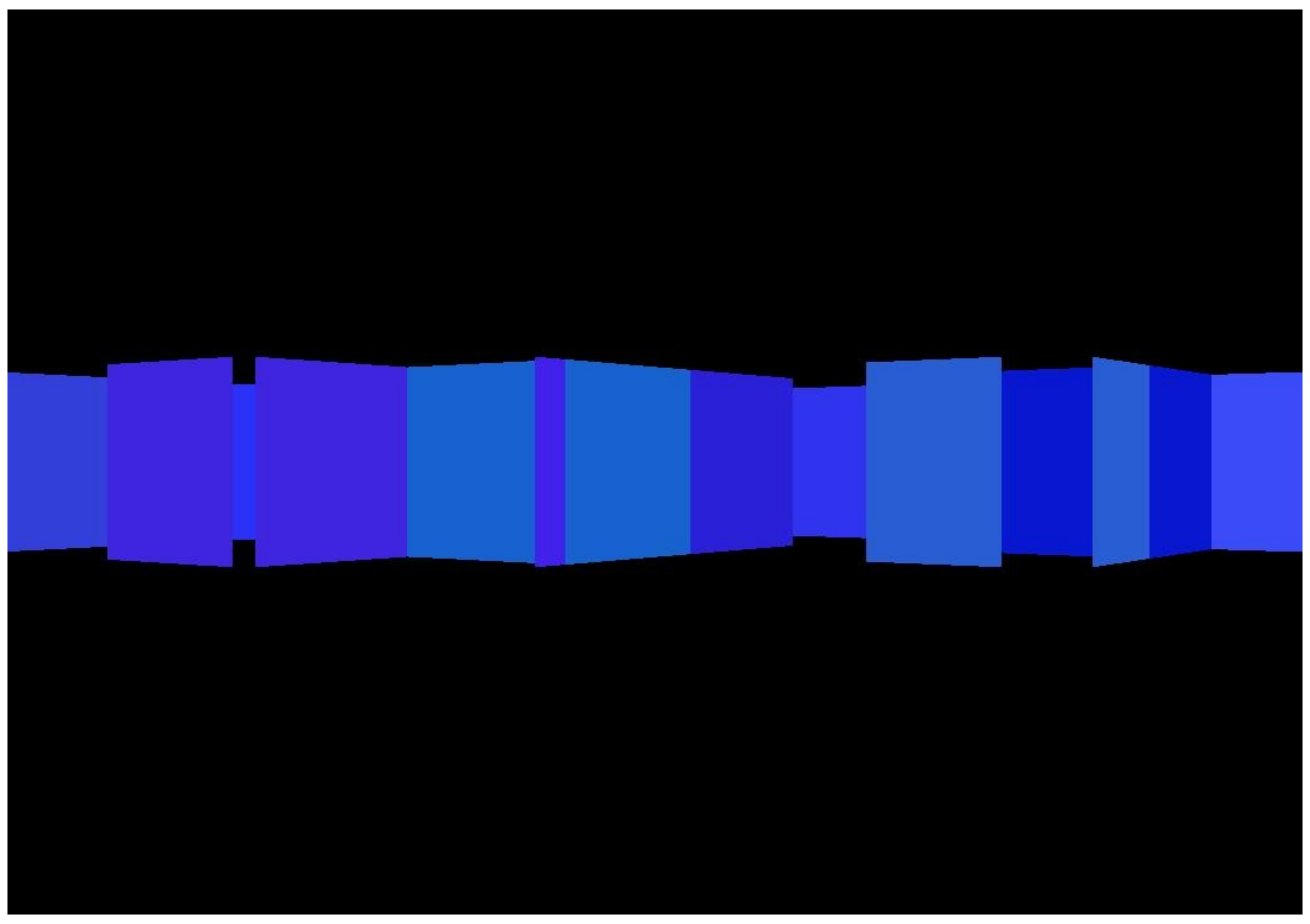
Background

The field of 3D computer graphics has been explored quite extensively, and comprises of three major parts: 3D modeling, animation, and 3D rendering. The first part, 3D modeling, refers to creating a 3D representation of an object. Animation, the second part, is moving the object through time. The final part, 3D rendering, is drawing the animated 3D model to the screen. 3D rendering is the most complex of the three parts, and is accomplished through several algorithms: polygon modeling, ray tracing, ray casting, or scanline rendering. This project will use the ray tracing algorithm, which casts a ray from the eye through each pixel of the virtual screen to the environment, calculating the length of the ray and using that to determine view distance, and using the piece of the environment the ray intersects to determine what to display.

Progress

Currently, the program is coded to run in fullscreen exclusive mode (FSEM) in order to display the game over the entire screen. The program runs without errors and displays the scene objects, and the view can be rotated. The menu consists of a title image and seven function buttons: New Game, Control, Sound, Save Game, Load Game, High Scores, and Quit. Quit exits the program, New Game creates an instance of the World class (which extends Frame) and sets the program to run in FSEM with the World class as the viewable display, and the other buttons do not have any coded functionality. The program displays a black background with randomly shaded blue cubes (the wall objects) connected in the fashion of contiguous walls, and accepts keyboard input for motion and returning to the main menu. The move method currently generates runtime errors, but the turnLeft and turnRight methods work appropriately.


A screenshot of the scene's initial view, as of 04/02/08.


A screenshot of a rotated view in the scene.