

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, but the movement of the scene has not yet been coded. 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 blue cubes (the wall objects), which are not yet connected in the fashion of contiguous walls, and accepts keyboard input for motion (the motion isn't yet coded, but the program still accepts the input) and returning to the main menu.

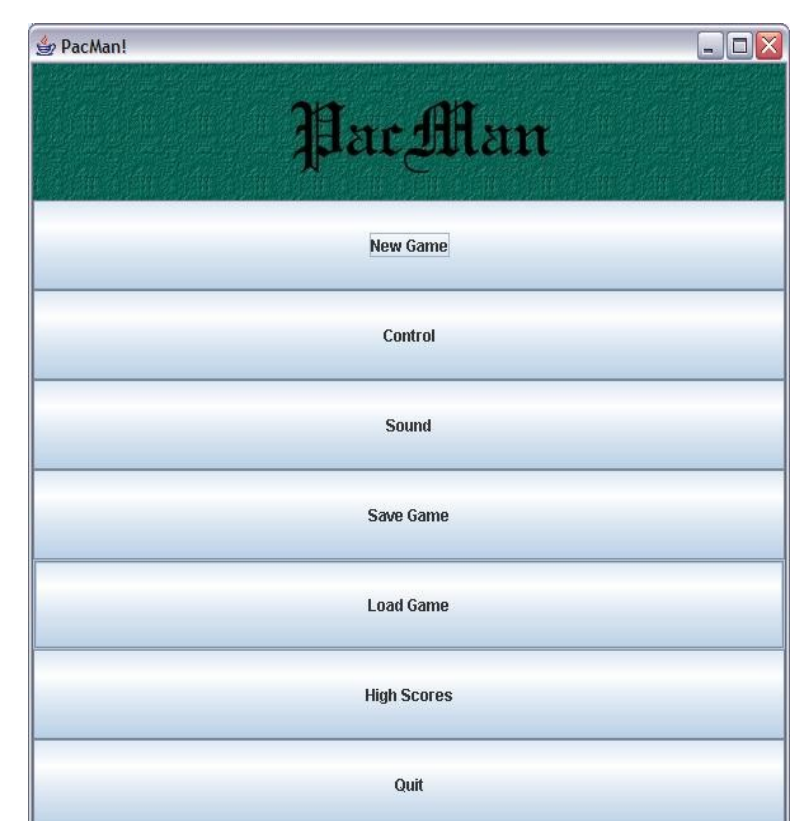
```

111111111111111111111111111111111111
100000000000001000000000000001
10111101111101011111011110111101
121 101 10101 101 121
1011110111110101111101111011101
100000000000000000000000000001
10111101101111111101101111011101
10111101101111111101101111011101
100000011000010000110000001
111111011111 1 111110111111
 1011111 1 1111101
 1011 a 1101
111111011 111 111 1101111111
 0 1b c d1 0
111111011 1111111 1101111111
 1011 1101
 1011 1111111 1101
111111011 1111111 1101111111
100000000000001000000000000001
1011110111110101111101111011101
1011110111110101111101111011101
12001100000000p00000000110021
11101101101111111101101101111
11101101101111111101101101111
100000011000010000110000001
10111111111110101111111111101
1011111111110101111111111101
100000000000000000000000000001
111111111111111111111111111111111111

```

A text representation of the map.

Key:
0 – A cookie
1 – A wall
2 – A powerup fruit
a, b, c, d – Ghosts
p – PacMan. Since Pacman is not visible to the user (he is the user), this is just the initial starting point of the view, which faces up relative to the text.
Blank - nothing



A screenshot of the game's main menu.