Performance Oriented Massively Multi-player Online Role-Playing Game

Jing Chan

April 9, 2003

1 Abstract

This project is to create a prototype of a Massively-Multiplayer Online Role-Playing Game that's geared towards performance-oriented processes for the servers. This would allow the servers to operate on average machinery and save costs to the server-side users.

2 Background

Massively-multiplayer online games with persistant worlds often experience heavy server load and require large amounts bandwidth to run. This means that without purchasing expensive resources, the game would not run with optimal performance and speed. This is a lose-lose situation for the end-user who will either deal with sub-optimal performance in the game, or an expensive monthly bill to pay for the service.

3 Procedure

The project uses several programming libraries. It utilizes OpenGL, X11Lib, Windows API, and sockets. OpenGL is the library that handles 2 dimensional and 3 dimensional drawing routines for the program. All graphical elements of this project are handled by this library. X11Lib is used for handling the keyboard, mouse, windows management, and time management for the X11 graphical interface (Linux Operating System). Sockets are used for transferring data between computers over the internet. This is essential for any online multiplayer game. The Windows API does much the same as the X11Lib functinos, but for the Windows Operating System.

4 Development

This project has been done while taking time constraints into account. Though type of game usually takes a team of workers to finish, it can still be done if efficiently done. The project has been divided into several sub-projects such as sockets, server-client model, collisions, terrain handling, and graphical content such as particle systems. Another time-saving element is randomly generated content. If creative work is left up to a random-number generator, it becomes less of a burden on the creator.

5 Results

This project proposes to produce a playable "MMORPG" (Massively-Multiplayer Online Role-Playing Game) that uses several memory and processing tricks to lighten the load on the server, and thus produce a playable game that does not require expensive servers and connections to host. Another proposed aspect of this project is to automate as much game content as possible so that minimal work needs to be done to have a playable game by this project's deadline.

6 Appendix

6.1 Code

Code for this project can be found in a zip file located at:

http://www.tjhsst.edu/ jchan1/techlab/project.zip

6.2 Resources

ID Software, The Quake Source Code

ID Software, The Quake2 Source Code

Game Programming Gems vol 2

6.3 Acknowledgements

Thanks to:

Mr. Latimer, Computer Systems Techlab Director for support and guidence.

Yoyo Zhou and Yan Zhang for collaborating with me on previous projects, which have given me the experience needed to undertake this current task.