MAPWAD - A 3D Modeling Language

Group Leader: Avrum Tilman amt77@columbia.edu
Ben Smith bhs16@columbia.edu
Josh Weinberg jmw211@columbia.edu
Ron Weiss ronw@cs.columbia.edu

September 23, 2003

Abstract

Traditionally, 3D modeling tools have taken the form of graphical editors. Advances in graphics technology have made it possible to create ever more detailed models, greatly increasing the complexity of the modeling process. We will develop a computer language, called MAPWAD, which can be used to algorithmically generate virtual environments. A small number of MAPWAD instructions will create quite complex 3D environments.

1 Introduction

Beginning in the early 1990's with games such as Wolfenstein 3D, 3D video games, especially first person shooters (FPS), have become enormously popular. DOOM, an FPS written by *id Software*, was probably the most popular game of the first half of the 90's. The immense popularity of DOOM led to a stream of copycats trying to cash in on the FPS craze. Slightly more advanced games, such as Heretic, Duke Nukem 3D, and Descent, were released shortly after DOOM. However, *id* would not be outdone. In 1996 *id* released Quake, a far more advanced and visually appealing game. Aside from its success on the graphics front, Quake revolutionized the concept of multiplayer gaming by allowing people to play "deathmatch" games against each other over the Internet. The ability for people to challenge friends remotely also served to boost Quake's popularity. The game was so successful that even today, 7 years after its initial release, one can still find Quake game servers hosting large multiplayer games. *id* went on to release two very successful sequels to Quake, and even licensed their 3D engine technology to other game developers.

Communities revolving around modifications of *id Software*'s games have existed for years on the Internet. File formats and level specifications were made widely available shortly after the release of DOOM and Quake. Enthusiasts around the world used this information to create level editors and work on creating new content for use in the games. Their efforts received a boost in 1997 when *id* released the source code for DOOM under the GNU General Public License. Two years later, they followed by releasing the Quake source as well. The tremendous online following originally created by the multiplayer capabilities of Quake still exist today precisely because of *id*'s willingness to allow fans to modify and extend their games.

Despite the interest in modifying 3D games, newcomers are put off by the large learning curve they need to cross to be able to create interesting modifications. The available tools, mostly in the form of graphical editors, are very complex to learn. In addition, for certain applications, a graphical editor might not be the best way to build game levels. The motivation for MAPWAD is to create a straightforward, yet powerful language for creating levels for games such as Quake.

2 Goals

The goal of this project is to design a language that will generate levels for a 3D game such as DOOM or Quake. We will be using the Quake rendering engine to represent the MAPWAP created models so as to take advantage of the open source code and tools that are available. Beyond gaming, the ability to algorithmically generate 3D interactive environments is useful for any virtual reality application. Using MAPWAD, designers will be able to take advantage of algorithmic programming constructs to create such environments. In other words, MAPWAD will make a simple description yield a complex result - what used to take an hour to do by hand will be done in just a few lines of MAPWAD.

Ease-of-use

MAPWAD will provide a high-level approach to virtual environment design. Instead of defining polygons based on their coordinates in 3D-space, MAPWAD will describe complicated scenes as collections of objects.

Automation

MAPWAD will harness the power of loops and conditionals to provide the user with the ability to generate regular structures. For example, instead of having to define every stair as a separate platform, MAPWAD will allow for the use of loops to create a staircase iteratively.

Flexibility

MAPWAD will not be limited to the production of video game levels. The generic nature of the modeling language will allow MAPWAD to be used for other tasks, such as: interior modeling/design, landscape modeling, or any other task that would require a 3D modeling solution.

Extendability

MAPWAD will be as generic as possible. With no direct connection to a specific game, it should be possible to port MAPWAD to a new rendering engine's specifications at a later date.

3 Example Applications

Simple levels

The obvious application of MAPWAD would be as a tool for generating game levels. The most common type of game that could take advantage of MAPWAD would be a 3D first person shooter such as DOOM or Quake. MAPWAD could be used to generate the layout and dimensions of the various rooms and corridors found in a typical first person shooter game level. The appropriate MAPWAD compiler would then translate the code into a level file for a specific game engine.

Advanced levels

More interesting applications of MAPWAD would take advantage of its control flow abilities. A random maze algorithm could be written in MAPWAD and used to generate mazes in the form of DOOM levels. Maps could, in fact, be generated based on any mathematical function that can be visualized in two or three dimensions.

MAPWAD could also be useful for describing geometric features that would be difficult or tedious to design manually. These might include complicated architectural features such as spiral staircases.

Interior modeling

MAPWAD's capabilities could extend beyond DOOM level creation. Although it is not intended to replace CAD systems, MAPWAD would be an excellent tool for initial concept designs by a builder and/or floor layout specialist. MAPWAD interactive designs could be produced more quickly than hand designed versions; because the models can be viewed on a widely available and inexpensive 3D engine, clients would be able to view and explore the models on their own hardware.

Simulation

The carefully hand designed game world may be a source of pride to the devoted DOOM or Quake fan. Other applications of interactive virtual environments however, will have different requirements and constraints. For military training simulations, for example, it might be desirable to be able to quickly generate many different interactive environments matching certain specifications, but that are not identical. A MAPWAD program could be written which would generate new levels with a random map layout, but always matching the requirements of the mission being trained for.

4 Sample Code

```
//a basic four walled room
room FourWaller {
    FourWaller(length, width, height) {
        //define a four wall room...
    }

    plane wall1,wall2,wall3,wall4,floor,ceiling;
    entry door; // door coresponds to the fourth wall
}

//a corridor with a door in the middle
room Corridor {
    //numDoors specifies number of doors opening out of
```

```
    //the middle of the corridor
    //doors[0] and doors[numDoors-1] are doors at either
    //end of corridor
    Corridor(length, width, height, numDoors)   {
        //define a corridor...
    }

    plane ceiling,floor,wall1,wall2;
    entry doors[numDoors+2];
}

//A simple circular map
//A circular corridor forms a loop, with rooms opening off
//of the corridor segments
map_start {
    FourWaller aRoom=FourWaller(200,200,50);

    Corridor aCorridor = Corridor(400,50,60,1);

    entry first=aCorridor.doors[0];

    attach(aCorridor.doors[1],aRoom.door);

    entry prev=aCorridor.doors[2];

    for(i=0;i < 10;i++) {
        aCorridor = Corridor(400,50,60,1);
        attach(aCorridor.doors[0],prev);

        aRoom=FourWaller(100,100,50);

        attach(aCorridor.doors[1],aRoom.door);

        prev=aCorridor.doors[2];
    }

    attach(prev,first);
}
```