

# **The Firefighting Simulator**

By

Jason Benson

Submitted to the Faculty of the Information Engineering  
Technology Program Partial Fulfillment of the  
Requirements for the Degree of Bachelor of Science  
in Information Engineering Technology

# The Firefighting Simulator

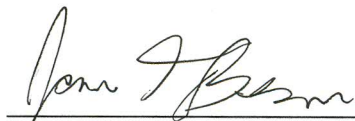
By

Jason Benson

Submitted to  
the Faculty of the Information Engineering Technology Program  
in Partial Fulfillment of the Requirements for  
the Degree of Bachelor of Science  
in Information Engineering Technology

© Copyright 2001 Jason Benson


The author grants the Information Engineering Technology Program permission to reproduce and distribute copies of this document in whole or in part.



Jason Benson

3/13/01

Date



Dr. Sam Geonetta

3-6-01

Date



Department Head: Lawrence G. Gilligan

3/13/01

Date

## **Abstract**

The Firefighting Simulator uses modern video game technology to create a realistic, three-dimensional environment. The environment consists of a two-floor office building set in an urban area. The user, through a first person perspective, navigates this environment to accomplish the mission objectives stated at the beginning of the simulation. To complete the objectives of the mission, the user must rescue anyone still in the burning building and extinguish the fire. The tools given to the user to complete the mission are a fire hose and a crowbar.

# Table of Contents

Section	Page
Acknowledgements	i
Table of Contents	ii
List of Figures	iii
Abstract	v
1. Statement of Problem	1
1.1 Definition of Need	1
2. Review of Literature	1
3. Description of Solution	2
3.1 User Profile	2
3.2 Design Protocols	3
3.2.1 Modeling	3
- Mesh	3
- Skinning	4
- Animation	5
3.2.2 Mapping	5
3.2.3 Interface	6
3.2.4 Programming	6
4. Deliverables	7
4.1 Complete Human Model	7
4.2 3D Environment	7
4.3 User Interface	7
5. Design and Development	8
5.1 Budget	8
5.2 Timeline	8
6. Proof of Design	9
6.1 Recommendations to the User	9
6.2 Files Used	10
6.3 Programs Used	12
7. Conclusions and Recommendations	16
7.1 Hardware Recommendations	16
7.2 Conclusion	17
Appendix A. Important Code	18
Appendix B. Simulator Screenshots	22
References	32

## List of Figures

	<b>Page</b>
Figure 1. Image of a Mesh	4
Figure 2. Image of a skin	4
Figure 3. Image of the joints in animation files	5
Figure 4. Image of the WorldCraft editor	6
Figure 5. Image of the Compilation Controller	13
Figure 6. Image of the FF_Sim Icon	13
Figure 7. Image of the PAK Explorer	14
Figure 8. Image of the WAD Viewer	14
Figure 9. Image of the Model Viewer	15
Figure 10. Image of the Flash .AVI	16
Figure 11. Image of the Sprite Viewer	16
Figure 12. Main Menu for the Simulator	22
Figure 13. Difficulty Menu	22
Figure 14. Starting point of the Simulator	23
Figure 15. The Fire Escape	23
Figure 16. Statement of the Situation	24
Figure 17. The main hallway on the first floor	24
Figure 18. View out of a window on the first floor	25
Figure 19. Office in the front of the building	25
Figure 20. Office in the back of the building	26
Figure 21. The Conference Room	26
Figure 22. The Men's Restroom	27

Figure 23. Extinguishing the first fire	27
Figure 24. The first fire is out	28
Figure 25. View from the second floor	28
Figure 26. Room on fire with no air source	29
Figure 27. Backdraft occurs when a window is broken, resulting in an explosion and death.	29
Figure 28. Fire in a storage area	30
Figure 29. One person rescued, one more left	30
Figure 30. The fire is out and everyone is safe	31

## **1. Statement of the Problem**

There are many jobs that require a person to work in hazardous situations. In firefighting more than 90 people are killed each year. Currently, the only training that a firefighter can receive is through studying books, classes, videos and real world experience. However, there are problems with all of these. Studying can give information needed to understand techniques but there is no real world experience. Real world training is the only way to experience situations that are necessary to practice and master techniques that are learned through study, but materials and preparations are expensive. Also, not all environments can be trained in due to lack of resources or manpower.

### **1.1 Definition of Need**

New technology has been only slowly introduced to the field of firefighting. Virtual Reality simulations could assist in the development of new training. Situations that are not seen in real world training exercises can be experienced. For example, firefighters who live in rural areas may not be able to train for hazardous waste spills or airplane emergencies. The experience that firefighters get from this training could prevent many injuries and deaths. This is a new technology and the development can be expensive. However, many fields currently benefit from this type of training.

## **2. Review of Literature**

To my knowledge, this is the first project that was designed with an existing game engine used for something other than a game (6). Modifications or Mods are games that are created from existing game engines. These are popular in the world of gaming because of the ability to create a new game without having to create it from scratch. Mods for the Half-Life game include Paintball, Team Fortress and Counter Strike. These,

and many other games, are based on the same technology that I used to create the Simulator for this project.

Many fields are benefiting from the use of training simulators. NASA uses a software package called Open Worlds. With this astronauts are able to train in simulations to prepare for work on the International Space Station (4). Flight Simulator have also been use to help train pilots for years. They are so common that anyone can buy one from a computer software store (1).

### **3. Description of Solution**

This project is designed to help train Firefighters. The program covers the basics of firefighting that new students of fire science will be learning. It helps the students by reinforcing the knowledge that they have learned in the classroom, without the risk of harm that could happen in real life training exercises. The final project involves an interactive three-dimensional scenario where users see through the eyes of an in-game firefighter who has the abilities and limitations of a normal human. The user has a specific task to complete in this environment. Timing and skill help the user complete the primary goal, the safety of the people inside. Through the use of this program, firefighters are able to anticipate future situations and how to resolve them.

#### **3.1 User Profile**

The intended users of this program are the students of Fire Science programs and Fire Service trainees. The program will need to be installed by someone with a moderate level of Information Technology experience. This person will need to know how to install programs from a CD-ROM and how to update files over the Internet. The user of the Simulator should have basic computing skills, such as how to use a mouse and keyboard.

Since this is how the Simulator is controlled, the better the user is at using these devices, the easier it will be to navigate. A brief introduction to the Simulator, by someone who has used it before, may be necessary for the users to understand how to run the program.

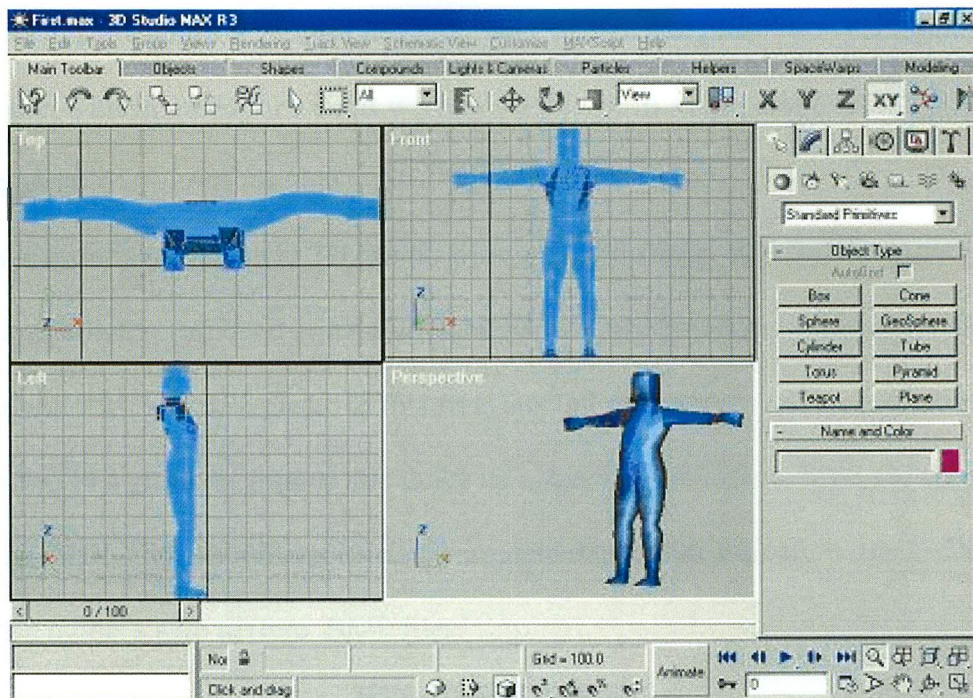
## **3.2 Design Protocols**

### **3.2.1 Modeling**

Modeling is the creation of anything that is three-dimensional and moves. It consists of three areas: building a mesh, skinning, and animation.

#### Building a Mesh

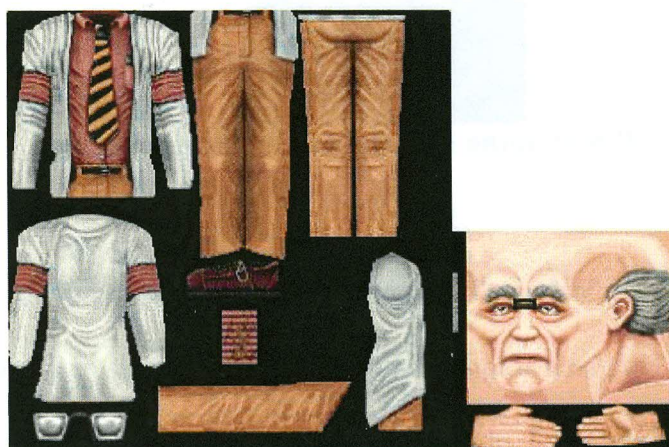
This process involves using a 3D editor. The two different programs used for this task are 3D Studio Max R3 and MilkShape. The first step is to decide the level of detail of the object (polygon budget). Too many polygons make the object's file size too large. Not enough polygons make the object indiscernible from other objects and the skin and animation of the object does not fit properly. For this project the polygon count is around 400. Once the degree of detail is decided, the actual model needs to be designed in the 3D editor. A finished mesh has the same shape as a complete model, but with no textures or animation.



**Figure 1. Image of a Mesh**

### Skinning

Skinning is a technique that takes a 2D image (the skin) and wraps it around a 3D image (the mesh) giving a realistic look. Because it is 2D, this image is designed using an image editor like Adobe Photoshop. The best way to describe the process of skinning is to imagine taking the skin off of a person and making it all face it one direction.

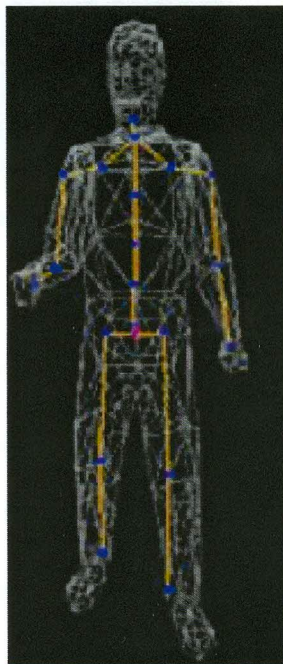


**Figure 2. Image of a skin**

Each skin is made for a specific mesh because of size and proportion changes. For example, a mesh made for a short woman would not fit a skin made for a large man.

### Animation

Animation is also done in 3D Studio Max and Milkshape. The model is given joints at which it can bend and move. It is then assigned specific poses according to the action that is to take place. In this final step all movements in the simulation will be linked to a group of frames that make up a specific animation. Files are compiled into an .MDL file. This file consists a complete mesh, 106 animation .SMD files and the model's skin texture .BMP files.



**Figure 3. Image of the joints in animation files**

### **3.2.2 Mapping**

Mapping is the process of creating the actual 3D environment. For this I used Worldcraft. Worldcraft is similar to other 3D editors, but is designed specifically to build and compile environments for the Half-Life game engine. Light entities, landscapes, structures, action triggers and textures are all incorporated into this program.

Mapping also creates the interactivity for the user with the environment. Objects are given characteristics that cause them to act in certain ways, like the ability to move, break or cause damage to the in-game character.

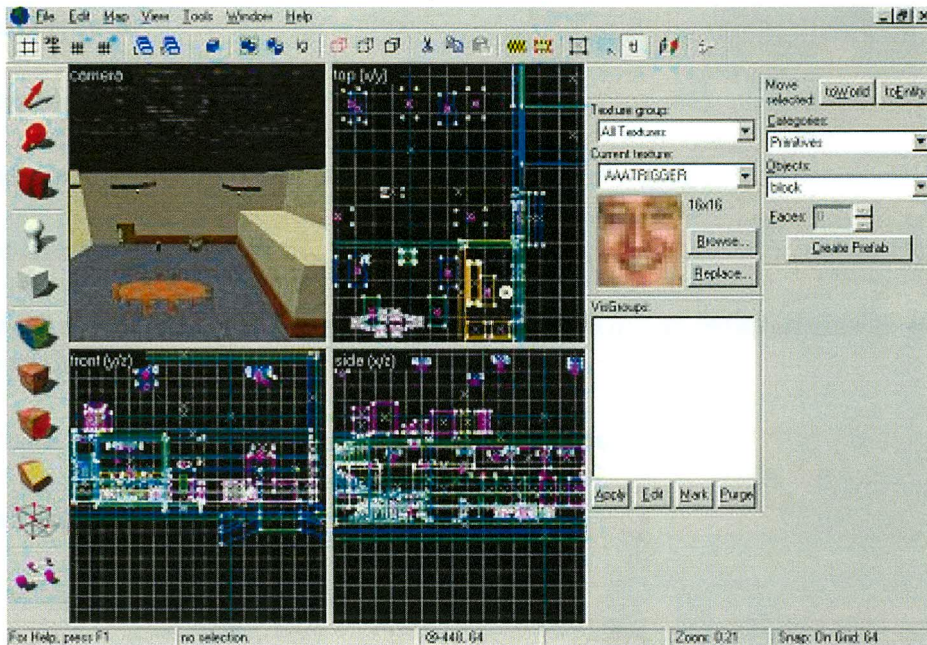


Figure 4. Image of the WorldCraft editor

### 3.2.3 Interface

The interface of the Simulator is what the user sees while in the program. This includes the splash screens and navigation menus. The menus include choices like Start a New Simulation, Change the Simulator controls, Load a Saved Simulation, and Go to the Simulator web site. When the Simulator is started, the user sees everything from a first-person point-of-view.

### 3.2.4 Programming

Visual Studio was used for the programming aspect of the project. Some of the code is already written since I am using the graphics engine of the video game Half-Life. However, the simulation is different from this game and because of this, code had to be re-written.

The code for this project controls the Artificial Intelligence (AI). The AI controls the activity of the fire. Triggers in the simulator make it act like a real fire. For example, water will decrease the size and intensity, while the introduction of a new air causes it to flare up and spread.

#### **4. Deliverables**

The Simulator is made up of the following objects.

##### **4.1 Complete human model**

- Mesh that contains less than 800 polygons, making an efficient model for the Simulator.
- Realistic skin designed specifically for this Mesh.
- All animations included.
- Packaged into the .MDL file format.

##### **4.2 3D Environments**

- Consists of several large office buildings. However, only one can be entered. The rest enclose the main environment.
- The building is fully furnished as a typical office setting, with desks, tables, chairs, doors, stairs, computers, lights, bathrooms, offices, filing cabinets and storage areas.
- Sound effects are associated with objects and actions such as doors opening and closing, footsteps and breaking windows.
- Ability to interact with environment, such as jumping, crawling and using objects.
- Creation of usable objects that are breakable and movable.

##### **4.3 User interface**

- Navigation menus allow the user to adjust settings, save and load simulators and change controls.
- Heads-Up-Display that shows user's health, current equipment used and spikes in temperature.

## 5. Design and Development

### 5.1 Budget

<i>Hardware used:</i>	<i>Price</i>
Pentium III	\$2700.00
<ul style="list-style-type: none"><li>• 900 MHz</li><li>• 128 MB RAM</li><li>• 64 MB NVIDIA Video Card</li><li>• 19" Monitor</li></ul>	
<i>Software Used:</i>	
Worldcraft	Freeware
3D Studio Max	\$3100.00
PhotoShop	\$170.95
Visual Studio	\$529.00
Licensed copy of Half-Life	\$29.99
SDK 2.0	Freeware
Half-Life Comp. Controller	Freeware
Icon Maker +	Freeware
Qped and PAKEplorer	Freeware
Wally	Freeware
MilkShape	\$30.00
Model Viewer	Freeware
Flash	\$200.00
SprViewer	Freeware
Total =	\$6759.94

### 5.2 Timeline

Fall 2000

- November 22  
Rough Draft for Final Map
- November 29  
Complete Firefighter model
- December 5  
Complete Model

December 6  
Documentation  
Presentation  
Prototype Simulator Complete

Winter 2001

January 18  
Incorporated Fire with Map

January 28  
Rough Draft of Complete Documentation

February 14  
Rough Copy or Final Paper

February 15  
Complete AI Programming

February 25  
Complete Final Simulation

March 2  
Complete Final Documentation  
Submit Final Project

March 13  
Final Presentation

## **6. Proof of Design**

### **6.1 Recommendations to the User**

A user of the Simulator will need to have familiarity with using a keyboard and mouse for navigation in the environment. Since the intended users are firefighters or those who are training to become a firefighter, knowledge of computer controls may be limited. Therefore, it is suggested that the user take the training course that is offered through the Half-Life menus. This will let a user practice jumping, using objects and moving. This familiarity will help the user succeed in the Simulator.

## 6.2 Files Used

There are many file types of files that were created for this project. They are all organized in one Folder called "Firefighter". Every file is in a specific location that the game engine can read from. Many of the file types are special formats made for the Half-Life engine and they require specialty programs to edit or create. The most important files, beside the actual Half-Life engine, are the .BSP files. This type of file is just one of the eleven files created during the compiling process of the .RMF and .MAP files. This file is the actual environment that the Half-Life engine uses for the Simulator. Without this file there would be nowhere to put the other graphics and information files. This type of file is placed into the "Maps" Directory.

There are several .BSP files for the Simulator. That means that there are several environments. However the user may never notice this. Since the virtual space that the Simulator occupies is over 161,000 square feet and is 70 feet tall, the environment needs to be split into manageable sections. When certain areas of the environment are passed through, the .BSP file changes. The user will see the word "Loading" on the screen for less than one second, then the training continues. The disturbance is minimal and is necessary to keep the Simulator running with little latency. The target size for each .BSP file is between 1.5 and 2 MB.

This project was highly graphical. All of the 3D objects used were created from scratch through 3D Studio Max, MilkShape, Photoshop or WorldCraft. Some of these files are packaged into a file format called .PAK. These files are similar to the .ZIP file format because they compress large files into a smaller single package, hence the name .PAK. The difference is that the Half-Life engine is able to open these and use them

while the Simulator is running. In order to read and edit these files a .PAK Reader is necessary.

The Simulator is designed to use four .Pak files named Pak0, Pak1, Pak2 and Pak3. Many other files are stored inside of these files. The goal is to put as many files as possible into the .PAK files because it greatly reduces the size and the user will not be able to change or edit them, thereby causing in the Simulator to fail. The only files left outside of the .PAK files are those that are used frequently in the program.

The navigation for the menus is held in the “gfx” folder in the .PAK file. These files are bitmaps that create the hotspots used for navigation to different areas in the Simulator’s navigation.

Other files that are included in the .PAK files are the .MDL and the .SPR files. The MDL files make-up the models of human figures and the .SPR files are used to create the fires and smoke used in the Simulator.

There are two important .DLL files used in this project. The first is the client.dll. This file is used to customize the user health display (HUD) and the type of equipment (Crowbar or Fire Hose) that is currently in use while in the Simulator. This file is stored in the cl\_dll folder. The second is the einar.dll file. This file lets the Half-Life engine know there is a modification of the game engine and to use the new, customized files.

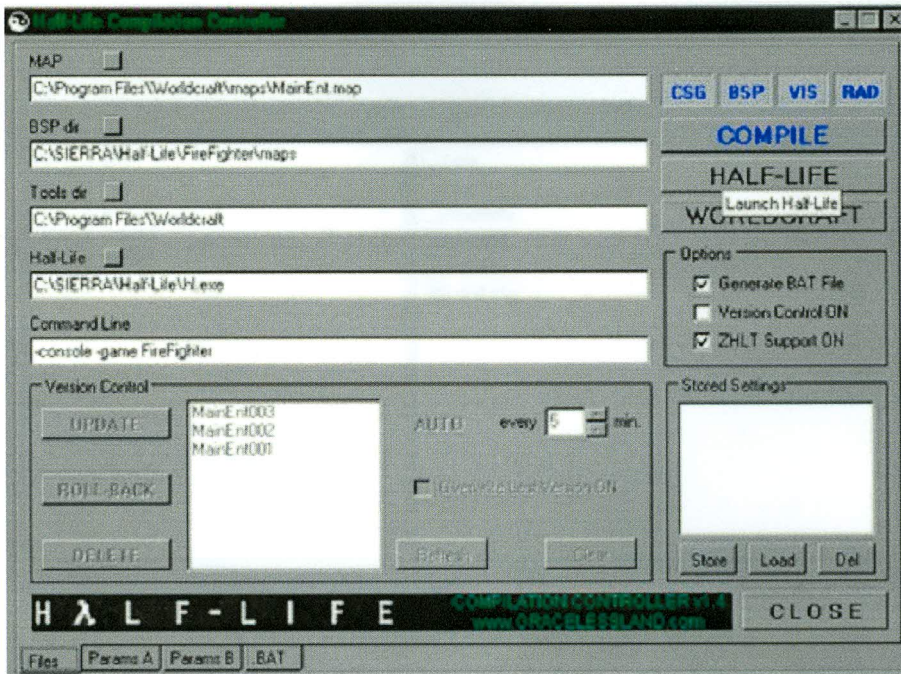
There are several files that allow the customization of the Simulator through the Half-Life engine. The config.cfg file gives the user the ability to assign specific buttons for different actions. This allows the user to create a configuration that is most comfortable. This file can be accessed and changed inside the Simulator’s navigation

under the “Controls” section. Another file that customizes the engine is the liblist.gam. This file does several things. First, it tells the Half-Life engine what the modification is. Second, it allows input for information such as the Web Site associated with the Simulator, which is accessible through the Navigation under “Previews”. Third, it tells what .BSP should be displayed at startup.

Finally, the last file for the Simulator is the FF\_Sim.lnk. This file links to the hl.exe which is the Half-Life program. However, this icon does more than just link, it tells the hl.exe to use the FireFighter directory as the home directory, making it use all of the customized files. This is the icon that will always be used to start the Simulator.

### **6.3 Programs Used**

- a. WorldCraft: WorldCraft is the most important of all programs used in the Simulator. It is used to create the environment. It creates the lighting, objects, buildings, sprites, textures, entities and triggers. This is where all of the files are combined into a final product. It produces its files in .RMF and .MAP formats, both of which are necessary to compile.
- b. Half-Life Compilation Controller: Program that inputs the WorldCraft’s .RMF and .MAP files. It uses several other tools (Zoner’s Tools) to control the compilation process and output the usable file .BSP.



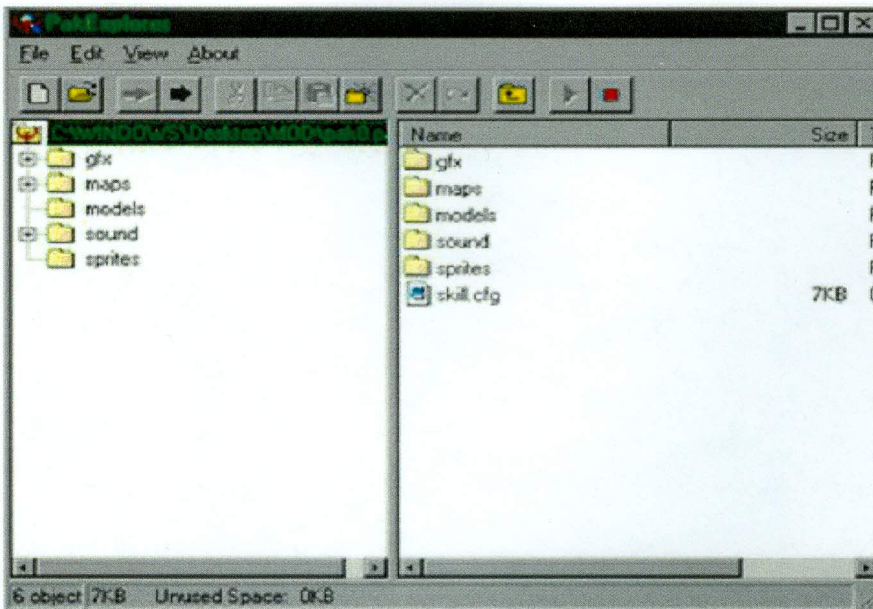
**Figure 5. Image of the Compilation Controller**

- c. Zoner's Tools: Consists of four files that are used with the Compiler program.
  1. hlRad: Controls the scaling and placement of objects in the .MAP file.
  2. hlCSG: Controls the textures of the objects used.
  3. hlVIS: Controls the lighting.
  4. hlBSP: Controls the creation of the .BSP files
- d. Icon Maker +: Used to design icons. Created the Fire.ico that links to the Half-Life engine and starts the Simulator.



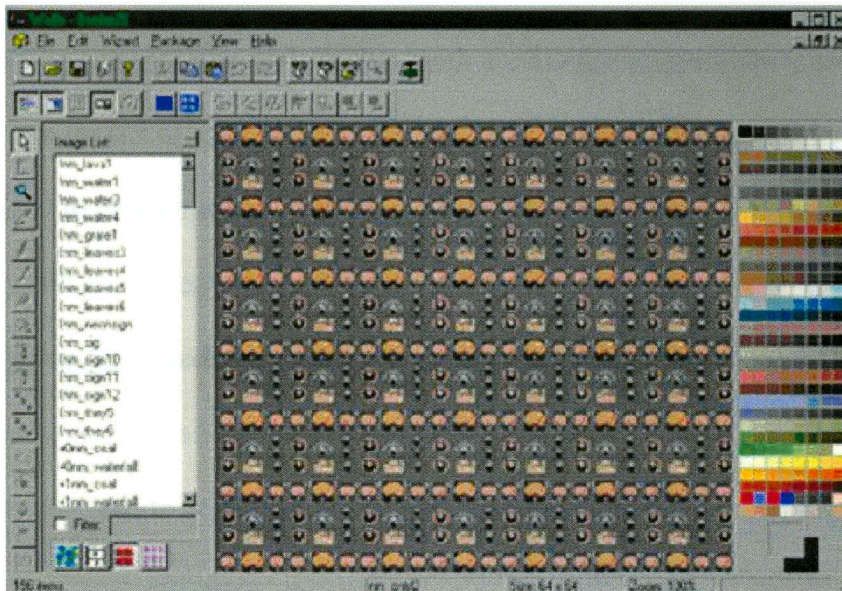
**Figure 6. Image of The FF\_Sim Icon**

- e. Qped and PAKExplorer: Both were used to view and edit the .PAK files. The use of the Qped program was needed to rename files that had names longer than ten characters.



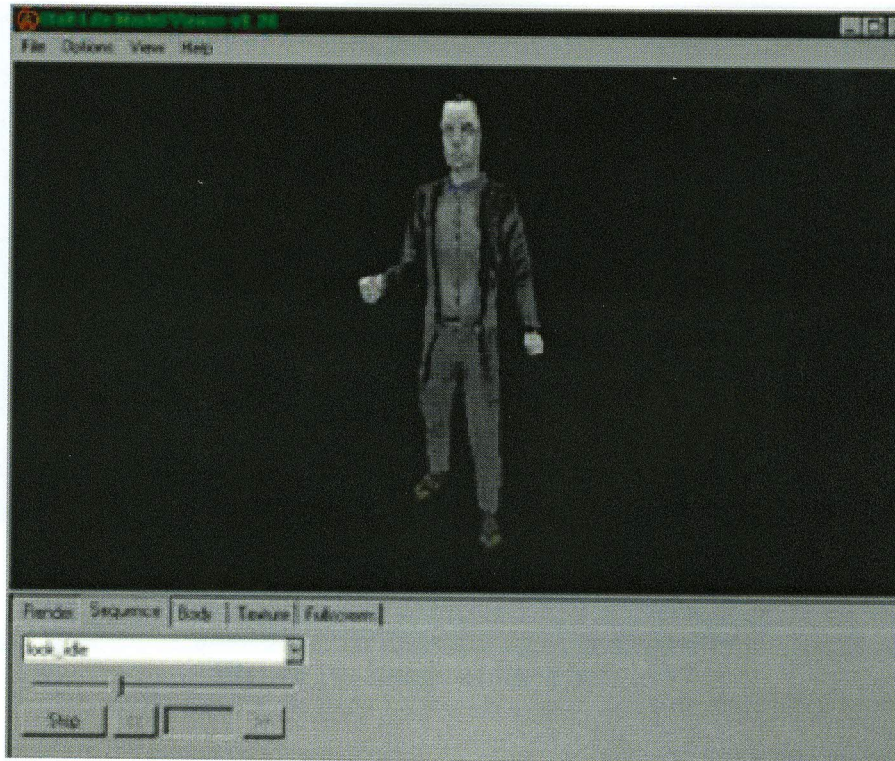
**Figure 7. Image of the PAK Explorer**

- f. Wally: The .WAD file reader and editor. The .WAD contains all of the textures for the Simulator that can be placed in the environment through



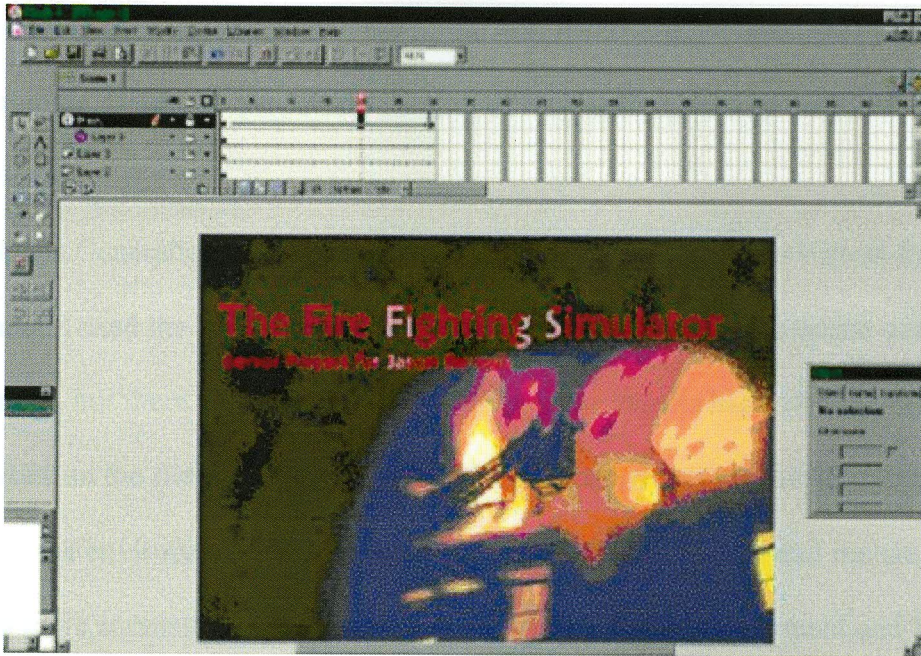
**Figure 8. Image of The WAD Viewer**

- g. Half-Life Model Viewer: Opens a viewable representation of models (human figures) and can view skins and animations in a test environment.



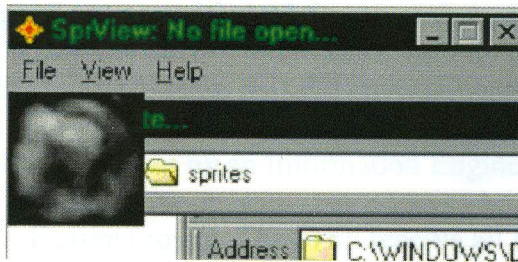
**Figure 9. Image of the Model Viewer**

- h. 3D Studio Max R3 and MilkShape: Both are 3D design programs, used to create the mesh for the models. MilkShape was used in certain situations because it was designed to create models for Half-Life. It had several specialty tools that 3D Studio did not have.
- i. Photoshop 5.5: Used to create the skins for the models. Also, to edit bitmaps used for the navigation and the splash screens.
- j. Flash 4.0: Used to create the intro .AVI in the Simulator.



**Figure 10. Image of the Flash .AVI**

- k. SprView: Used to view sprites (.SPR files)



**Figure 11. Image of the Sprite Viewer**

- l. Visual InterDev: Used to open and edit files with .GAM, .CFG and any other file that required text editing.

## 7. Conclusions and Recommendations

### 7.1 Hardware Recommendations

This project was designed on a 900 MHz computer with 128 MB of RAM and a 32 MB video card. The operating system was Windows 98. I recommend that the Simulator be used on a 450 MHz machine that has 96 MB of RAM and a 16 MB video card with a full version of the game Half-Life. Anything less than this may cause latency

## Appendix A. Important Files

### Config.cfg

```
unbindall
bind "TAB" "+showscores"
bind "ENTER" "+use"
bind "ESCAPE" "cancelselect"
bind "" "+moveup"
bind "+" "sizeup"
bind "," "+moveleft"
bind "-" "sizedown"
bind "." "+moveright"
bind "/" "+movedown"
bind "1" "slot1"
bind "2" "slot2"
bind "3" "slot3"
bind "4" "slot4"
bind "5" "slot5"
bind ";" "+mlook"
bind "=" "sizeup"
bind "a" "+moveleft"
bind "c" "messagemode"
bind "f" "impulse 100"
bind "q" "lastinv"
bind "s" "+back"
bind "t" "messagemode2"
bind "v" "+moveup"
bind "w" "+forward"
bind "[" "invprev"
bind "]" "invnext"
bind "`" "toggleconsole"
bind "~" "toggleconsole"
bind "UPARROW" "+forward"
bind "DOWNARROW" "+back"
bind "LEFTARROW" "+moveleft"
bind "RIGHTARROW" "+moveright"
bind "ALT" "+strafe"
bind "CTRL" "+reload"
bind "SHIFT" "+speed"
bind "F5" "snapshot"
bind "F6" "save quick"
bind "F7" "load quick"
bind "F10" "quit prompt"
bind "INS" "+klook"
bind "PGDN" "+lookdown"
```

```
bind "PGUP" "+lookup"  
bind "END" "centerview"  
bind "KP_END" "+jump"  
bind "KP_INS" "+duck"  
bind "MWHEELDOWN" "invnext"  
bind "MWHEELUP" "invprev"  
bind "MOUSE1" "+attack"  
bind "MOUSE2" "+attack2"  
bind "PAUSE" "pause"  
hud_takeshots "0"  
hud_classautokill "1"  
pushlatency "-1000"  
hud_centerid "0"  
hud_fastswitch "0"  
hud_capturemouse "1"  
joystick "0.000000"  
sensitivity "9.800000"  
m_filter "0.000000"  
m_side "0.8"  
m_forward "1"  
m_yaw "0.022"  
m_pitch "0.022000"  
cl_vsmoothing "0.05"  
cl_backspeed "400"  
cl_forwardspeed "400"  
lookspring "0.000000"  
lookstrafe "0.000000"  
cl_cmdrate "30"  
cl_allowupload "1"  
cl_allowdownload "1"  
cl_download_ingame "1"  
cl_cmdbackup "2"  
cl_timeout "35"  
cl_idealpitchscale "0.8"  
cl_himodels "1.000000"  
cl_dlmax "128"  
tracker "0"  
cl_lc "1"  
cl_lw "1"  
cl_updaterate "20"  
rate "9999.000000"  
bottomcolor "6"  
topcolor "30"  
model "gordon"  
name "Rash"  
_snd_mixahead "0.1"
```

bgmvolume "1.000000"  
hisound "1.000000"  
suitvolume "0.250000"  
volume "0.800000"  
s\_eax "0.000000"  
s\_a3d "0.000000"  
s\_verbwet "0.25"  
s\_bloat "2.0"  
s\_numpolys "200"  
s\_polysize "10000000"  
s\_polykeep "1000000000"  
s\_refdelay "4"  
s\_refgain "0.4"  
s\_leafnum "0"  
s\_max\_distance "1000.0"  
s\_min\_distance "8.0"  
s\_automax\_distance "30.0"  
s\_automin\_distance "2.0"  
s\_distance "60"  
s\_doppler "0.0"  
s\_rolloff "1.0"  
gl\_monolights "0"  
gl\_flipmatrix "0"  
gl\_overbright "1"  
gl\_polyoffset "-0.001"  
gl\_dither "1"  
mp\_decals "300"  
r\_bmodelhighfrac "5.0"  
viewsize "120.000000"  
sv\_aim "0.000000"  
net\_graphsolid "1"  
net\_graphpos "1"  
net\_scale "5"  
net\_graphwidth "192"  
net\_graph "0"  
con\_color "255 155 50"  
fps\_modem "0.0"  
fps\_max "72.0"  
brightness "1.000000"  
gamma "2.500000"  
crosshair "1.000000"  
console "1.0"  
+jlook  
+mlook

## Liblist.gam

```
// Fire Fighting Simulator Setup File
game "Fire Fighting Simulator"
gamedir "FireFighter"
url_info "www.FireFightingSimulator.com"
url_dl ""
version "1.0"
hlversion "1100"
size "90000000"
type "Single"
startmap "Outside"
trainmap "basic"
gamedll "dlls\einar.dll"
cldll "1"
svonly "0"
```

## Strings.lst

```
// String patches
```

```
// #define IDS_MAIN_PREVIEWSHHELP      400
// #define IDS_MAIN_CUSTOMHELP        530
// #define IDS_MAIN_RETURNHELP        188
// #define IDS_MAIN_NEWGAMEHELP       189
// #define IDS_MAIN_TRAININGHELP      190
// #define IDS_MAIN_LOADHELP          191
// #define IDS_MAIN_LOADSAVEHELP      192
// #define IDS_MAIN_CONFIGUREHELP     193
// #define IDS_MAIN_READMEHELP        194
// #define IDS_MAIN_ORDERHELP         195
// #define IDS_MAIN_QUITHELP          196
// #define IDS_MAIN_QUICKHELP         197
// #define IDS_MAIN_MULTIPLAYERHELP   198
// #define IDS_MAIN_QUITPROMPTINGAME  235
// #define IDS_MAIN_QUITPROMPT        236
```

```
402 "www.FireFighterSimulator.com"
```

```
400 "Check for new Simulations in the Future"
```

```
189 "Try to pass all of the Simulators for Certification"
```

```
190 "Skip to another Simulation (you must start New Game first to choose skill)."
```

```
234 "Starting another Simulation will terminate your current Simulation, OK to exit
without saving?"
```

Appendix B. Simulator Screenshots

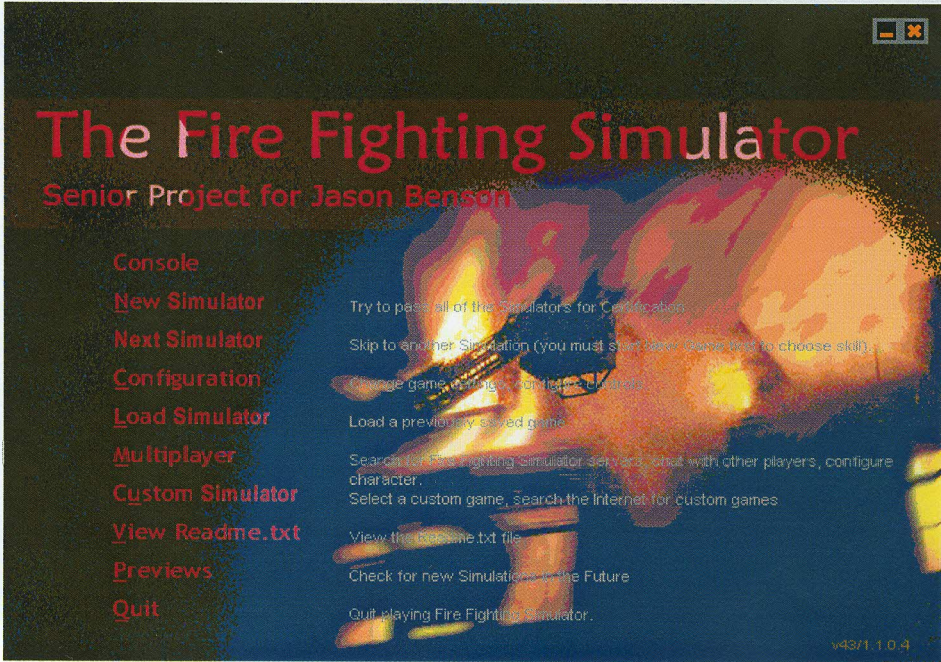


Figure 12. Main Menu of the Simulator



Figure 13. Difficulty Menu



**Figure 14. Starting point of the Simulator**



**Figure 15. The Fire Escape**

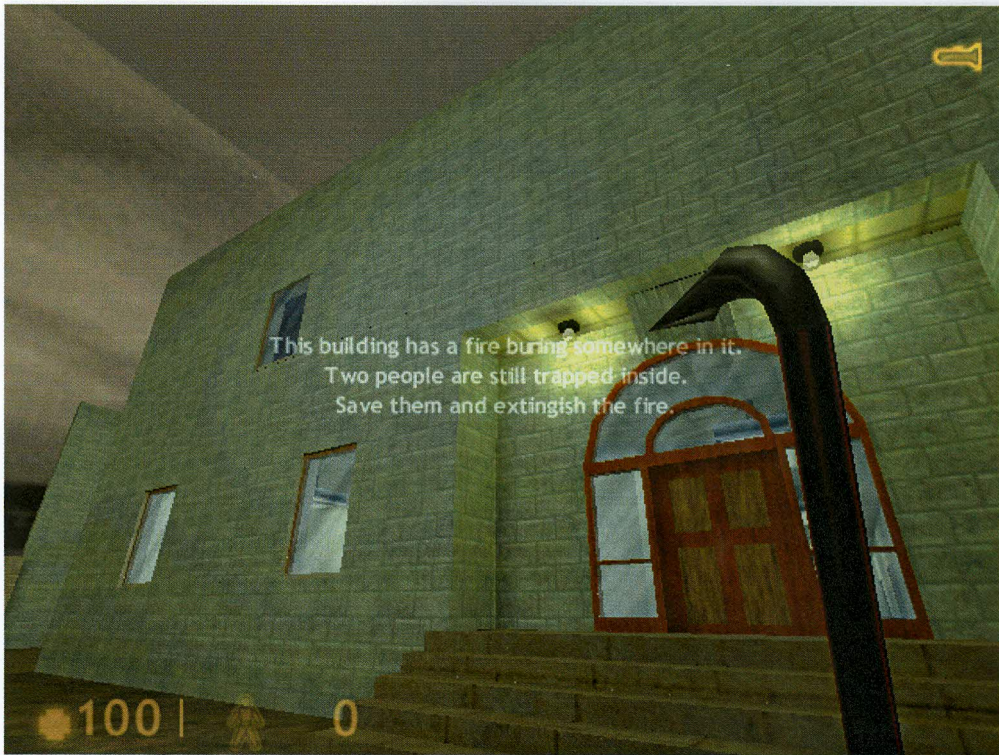


Figure 16. Statement of the Situation



Figure 17. The main hallway on the first floor

Figure 19. Office in the front of the building



**Figure 18. View out of a window on the first floor**

*Figure 18. Office in the back of the building*



**Figure 19. Office in the front of the building**



**Figure 20. Office in the back of the building**



**Figure 21. The Conference Room**



Figure 22. The Men's Restroom



Figure 23. Extinguishing the first fire



Figure 24. The first fire is out



Figure 25. View from the second floor



Figure 26. Room on fire with no air source

Figure 28. Fire in a storage area



Figure 27. Flashover occurs when a window is broke, resulting in an explosion and death.

Figure 29. One person died, one more left



Figure 28. Fire in a storage area



Figure 29. One person rescued, one more left



**Figure 30. The fire is out and everyone is safe**

## References

1. AETI. "SimPilot"  
<http://simpilot.com/>. February 2001.
2. CNet Networks. "Cnet.com".  
<http://www.cnet.com/>. November 4, 2000.
3. Fingr. "Half-Life Programming Tutorials".  
<http://www.planethalflife.com/hlprogramming/>. October 2, 2000.
4. Marshal Space Flight Center. "Virtual Reality Software Developed For The International Space Station Comes Down To Earth".  
<http://www.nasasolutions.com>. February 2001.
5. Scarecrow. "Planet Half-life Model Resource".  
<http://www.planethalflife.com/coldfusion/>. September 15, 2000.
6. Sluggo. "Mod Central".  
<http://www.planethalflife.com/modcentral/>. October 10, 2000.