EURECOM

3D GRAPH COURSE

LAB N°3 : HAMMER

Author: **P. Gros,.**
Environment: **Hammer – Zhlt - BspViewer**

Introduction

The goal of this TP is to experiment the concepts presented in the course. During this TP you are going to use an interactive 3D modeling program called hammer, this program will allow you to describe 3D scenes that may be compiled into half-life 3Dgames maps. You will experiment how to model 3D objects using 2D projections. You will be able to navigate interactively in the scene, compile the scene and view the scene in a half-life map viewer. The compilation process is very interesting to demonstrate Radiosity.

The 3D Gaming market is very popular and leads the 3D market. There are a lot of 3D games types (sports, adventure, killing …). Quake and its derivatives (Half-life and Counterstrikes) are (or were ☺) very popular 3D games and they provide a tool that allows everybody to design a virtual world where the game may take place. This tool was called worldcraft and is now called hammer. Beside the fact that it is interesting to experiment with a gaming design tool, this TP will allow you to understand some course concepts (visibility through 3D sorting, texture mapping, radiosity and illumination maps), it will face you for the first time with an interactive modeler.

During this TP you are going to use 3 softwares :

- **hammer** is the half-life map modeling tool. It is a 3D scene editor that creates scene in the **RMF format**. The program also exports RMF files to **MAP files**. Hammer provides a graphical interface that allows you to interactively describe a scene.

  `(All Programs/Valve hammer/Valve hammer editor)`

- **Zhlt compiler** is a compiler that you have to use in order to transform MAP files to **BSP files**. BSP files may be directly used by half-life, or

counterstrikes games. The compilation process is very important to understand.

- **Bspviewer** is a BSP file viewer (it is not a game such as half-life), it allows you to navigate in the game map, but not to play with it.

```
(All Programs/BSPViewer/BSPViewer)

You have to drag and drop a BSP file onto the Bspviewer
executable
```

You may find a French tutorial about using hammer at the following address, other tutorial can be found on the web (youtube etc…) :

https://openclassrooms.com/courses/une-map-avec-hammer

The files described in that document are accessible at the following location :

\\datas\teaching\courses\3DGraph\Lab3_hammer

Subject

You will send your answers to each question by email to gros@eurecom.fr with the subject message message being "Hammer TP". You will attach the "rmf" (office and floor) files to your message, (note : these files should compile). The TP is structured in one big exercise split into several steps. The objective is to model the Eurecom building and navigate into it.

Step N°1 : build an office

Using hammer please create the model of an Eurecom office and save it as the **"office.rmf"** file. For this purpose, you should :

1. Create a box for the room **(~5x8x3 in 64 units)**, be careful the units are important (a too big or too small size will lead you to problems.
2. Save the file and give it a name with an rmf extension so that you don't loose your work, and save regularly after that.
3. Hollow the box (~8 units outside)
4. Add a door an a window (figure 1). Please just add a block in front of the wall and do not do a hole in the wall for the door and the window)
5. Map textures onto the ceiling, the ground the walls, the door and the window (figure 2).
6. Add two desks (with computers), two chairs and two cupboards copied from the "elements.rmf" file (figure 3).
7. Add four lights to the office file, one on each desk and two on the ceiling. Please choose the simplest light.

Please try to define a usable hierarchy so that things may be move or duplicate easily. Do you have any comments about hierarchy management ?

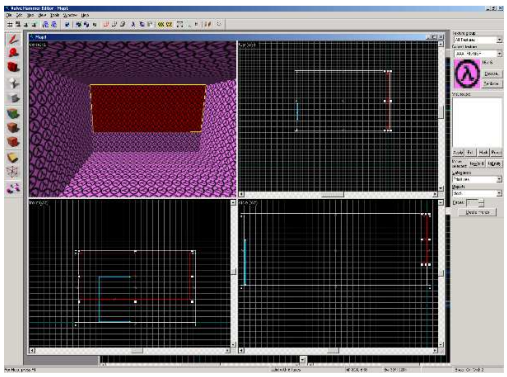


Figure 1

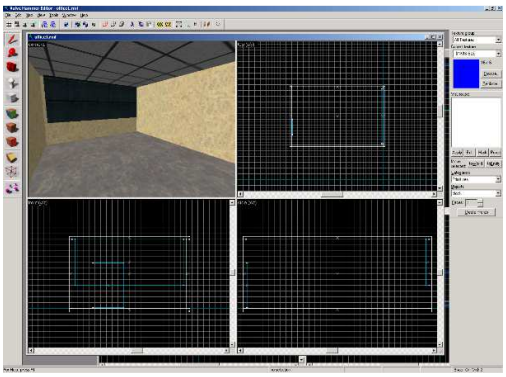


Figure 2

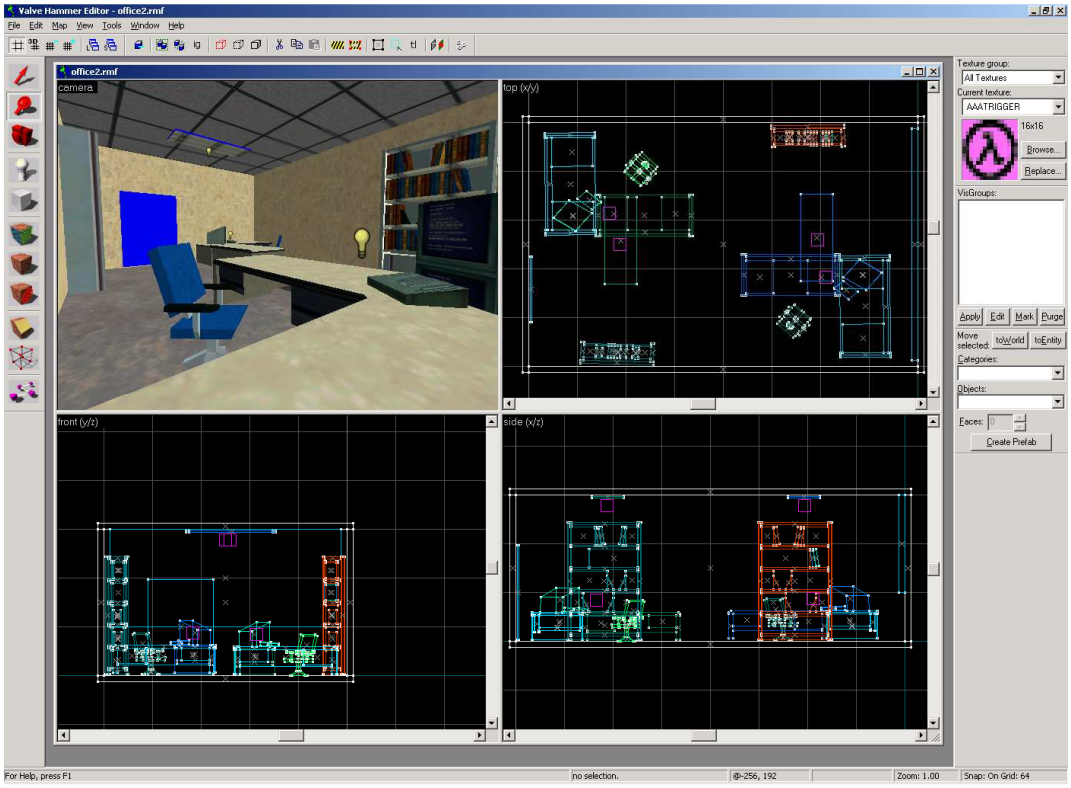


Figure 3

Step N°2 : compile an office

Export the map into a "`office.map`" file. Using Zhlt compiler please compile this file into a **"office.bsp"** compiled map. Take care during the compilation because the compiler is very error prone. Listen carefully to the professor's advises and make some attempts. View the map into the Bspviewer program.

You can play with the “render” menu in order to select different viewing parameter. Please explain the “Lightmaps” toogle item as well as the texture toogle item (like in figure 4 5 and 6). Explain the things you see.

Figure 4

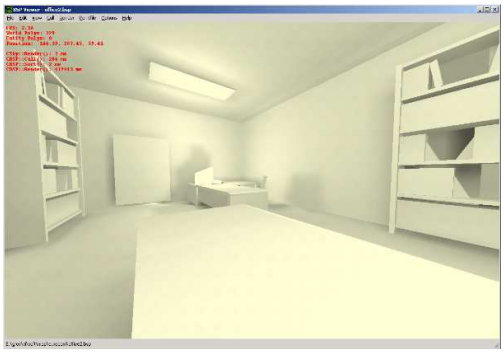


Figure 5

Figure 6

Step N°3 : build a floor

We are now going to model an Eurecom floor (the lab section) with offices and labs (see figure 7). You will have to generate a floor.rmf file and do the following :

1. Create a box for the full floor (~50x20x3 in 64 units)
2. Hollow the box (~8 units outside)

3. Save the file and give it a name with an rmf extension so that you don't loose your work, and save regularly after that.
4. Add the walls for the corridor and the walls for the offices and labs.
5. Map the textures on the ceiling, floor, and walls
6. Carve the doors in the corridor's walls.
7. Add the light, textured windows, desks, chairs, computers and cupboard in the offices and labs.
8. Add the lights in the corridor.

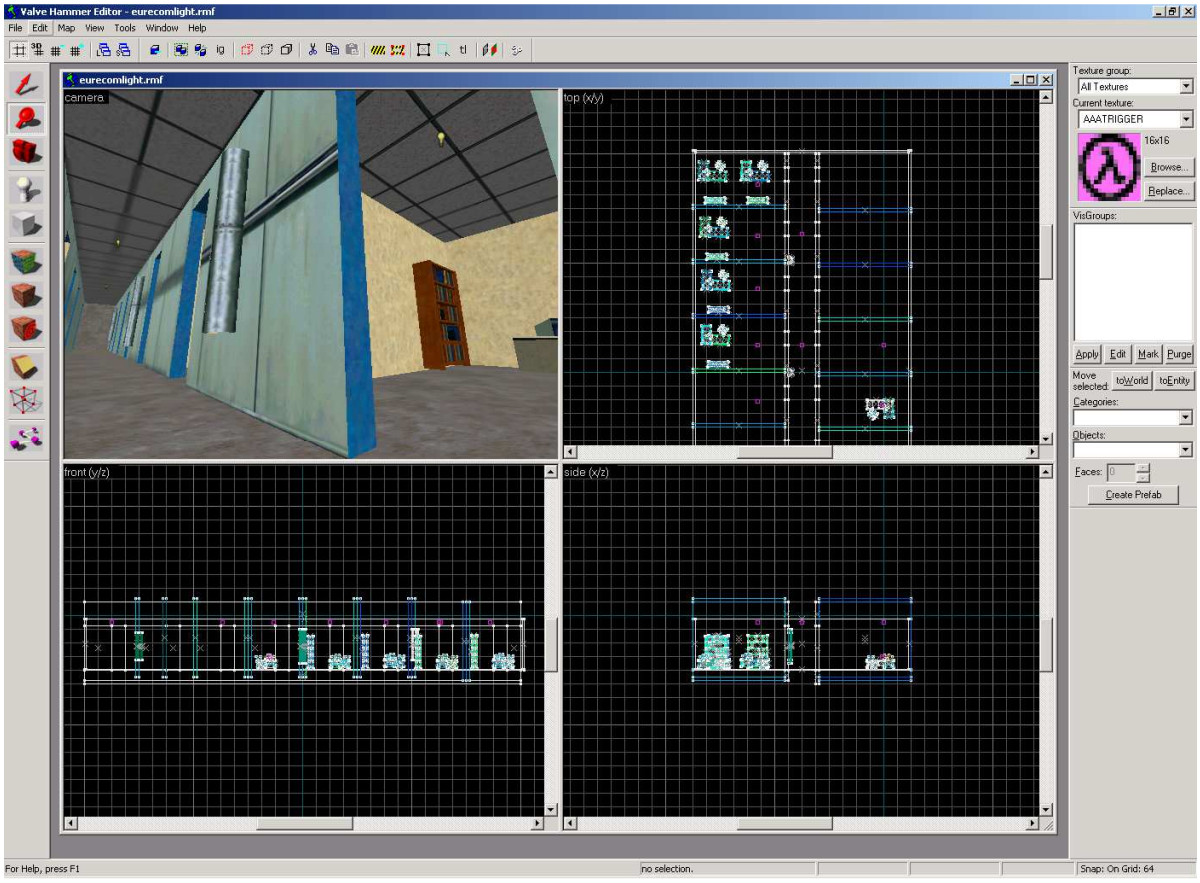


figure 7

Step N°4 : compile a floor

Compile and view the new scene.

figure 8

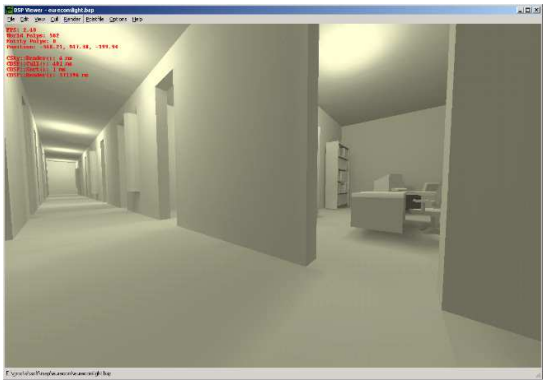


figure 9

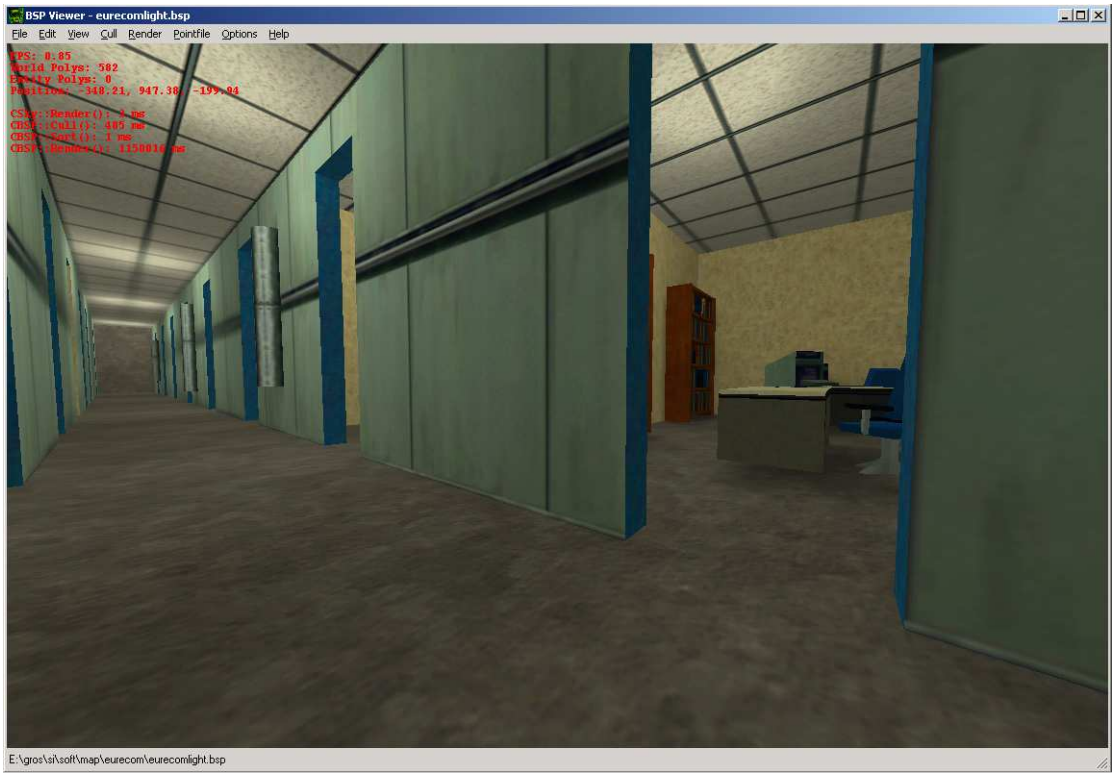


figure 10

Step N°6 : analyse the scene

Please answer the following questions :

1. Using the floor.bsp file, play with the 3 “Cull” menu items and explains what you see in the scene info according to what you have learned during the course (note: the frustum means view volume). As par of these explanations you will give a matrix showing the number of world polygons according to the selected item (please try different view directions (face the whole scene, put most of the scene outside the

view frustum etc…). The scene info may be toggled on and off using the View -> scene info menu.

2. Experiment with the freeze tool (in the same menu) : it is very convenient to understand what frustum, backface or viz optimization are actually doing, which polygons are removed according the freezed cull choice. Explain what does the freeze tool, some screenshots would be a reel plus to your explanations (choose extreme situations where the optimizations are very effective. Give some explanation to the strange image you may obtain using this tool.
3. Using the floor.bsp file, play with “render” menu items (mainly texture and lightmaps and explain what you see).
4. What are lightmaps ? (how many of them, how are they computed ? what is there usage ? how are they used etc…)
5. Can you define the resolution of a lightmap and the resulting problems, can you show an image with an artifacts due to that problem ?

ANNEX (software quick howto)

1-The hammer GUI

The application has a

- A menu bar (file, edit etc…)
- An horizontal tool bar (for specific actions)
- A vertical tool bar (for defining the mode)
- The working area is splitted in 4 main windows (3D, top side and left).

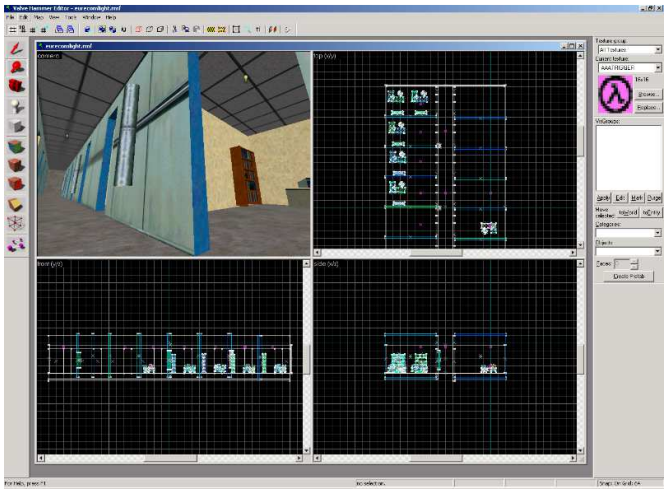
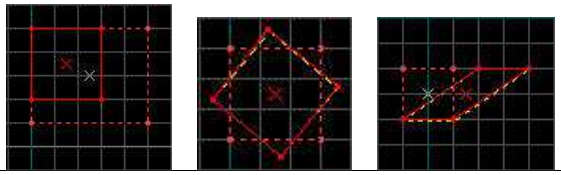
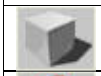
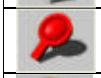


Figure 11

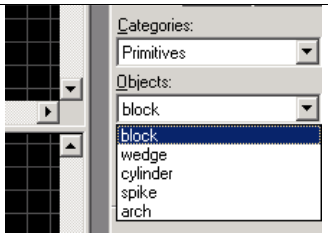
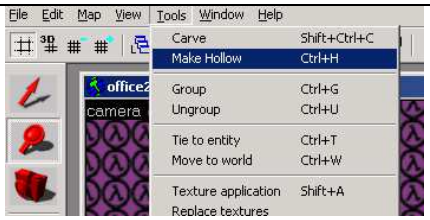
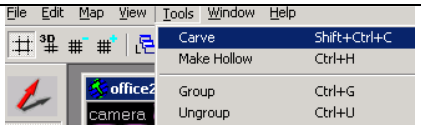
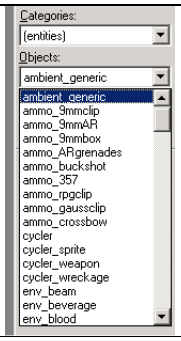
The vertical tool bar allows you to choose the mode of interaction. The following table gives you some explanation about the tools to use for this TP.

<table>
<tr><td>  </td><td>The “selection tool” for selecting objects
- the ctrl key allows you to select several blocs
- you can copy/paste the selection using Ctrl+C Ctrl+V
- you can scale (one click), rotate (two clicks), and shear (three clicks) th bloc
  </td></tr>
<tr><td>  </td><td>The “bloc tool” for creating blocs (just draw the bloc and press enter)</td></tr>
<tr><td>  </td><td>The “magnify tool” for zooming (left click) or unzoom (right click)</td></tr>
<tr><td>  </td><td>“The texture tool”
 
Opens the texture popup
- browse allows you to choose a texture
- The mode allows you to :
○ Lift : select a texture from a face
○ Apply texture only : apply the selected texture to the face you are going to click on</td></tr>
<tr><td>  </td><td>“The entity tool” for creating lights and other entities (see the lower right part of the screen for selecting the entity you want to add)</td></tr>
</table>

In order to move in the 3D scene, put the mouse in the 3D window and :

- space bar + left button : turn around
- space bar + right button : pan move
- space bar + shift + right button : front and back move

Other usefull things to know :

Upper left		The snap buttons (on/off, 3D view, more and less)
Middle right		The type of blocs you can add : - Blocks - prisms - cylinders - cones - arch (some parameters required)
Tool menu		Hollow a block (positive inside, negative outside)
Tool menu		Carve with the current bloc (you have then to delete the bloc)
Tool menu		Group ungroup selected objects
Lower right (while in entity mode)		Choose an entity

1-The ZHLT Compiler

In order to compile a map file into a bsp file, one solution is to use ZHLT compiler. This program use a 4 stage compilation process (CSG-BSP-VIS-RAD). There are o One command for each step :

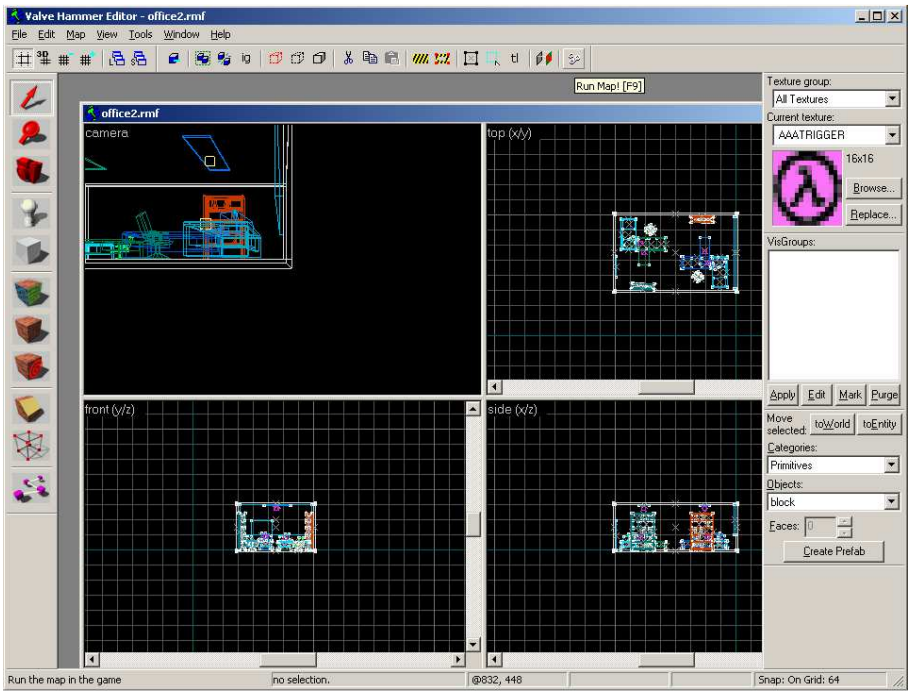


Figure 12

- **C:\PACKAGES\zhlt3.4\hlcsg.exe <yourfile> -nowadtextures** :
  CSG transforms and verifies the data (geometry). You should use it using the "–nowadtextures" option
- **C:\PACKAGES\zhlt3.4\hlbsp.exe <yourfile>**
  BSP transforms the previous data to a bsp file, verify that there are no leaks
- **C:\PACKAGES\zhlt3.4\hlvis.exe <yourfile>**
  VIS optimize the visualization process (defines what is visible from where.
- **C:\PACKAGES\zhlt3.4\hlrad.exe <yourfile> -bounce 3**
  RAD do the radiosity computation and adds the Lightsmaps. You should use it using the "–bounce 3" option.

There is a possibility to directly run the four commands in sequence from hammer (see figure 12). When you click on the "Run Map!" button then you should fall onto figure 13. If a window shown of figure 14 appears, please choose the expert mode and you should fall on a window shown on figure 13. Verify that the compiler options are set as shown in figure 15 and 16 (specially the "parameters" text box for the second and fifth compile commands (-nowadtextures and –bounce 3).

It is also possible to directly run each command from the cmd tool and thus to have a better feedback on the progression of the compiler.

Be careful that the compiler is very sensible to errors and that it sometimes does not compile because it thinks that nothing should be done, changing some flags generally solves the problem). Figure 17 through 20 gives you the configuration of the compiler.

There are a lot of issue that we will not cover during that TP, the main point is that your world should be closed (no leak) for the radiosity algorithm to converge.

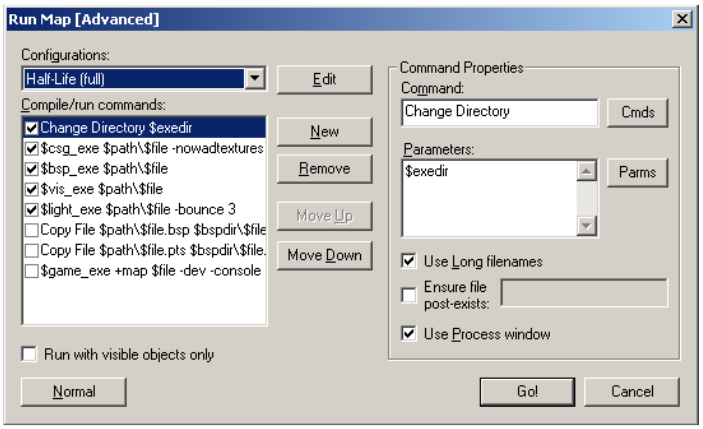


Figure 13

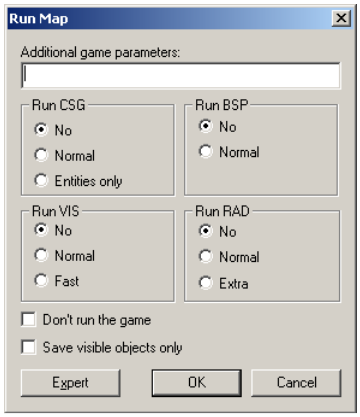


Figure 14

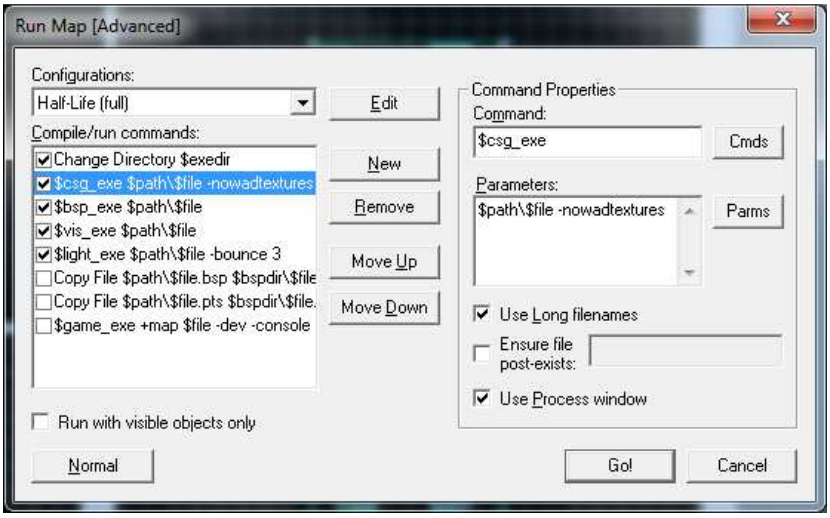


Figure 15

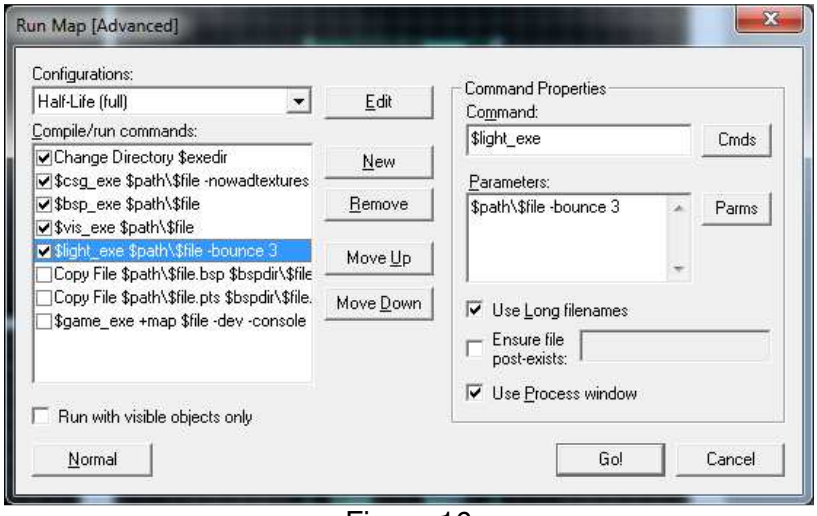


Figure 16

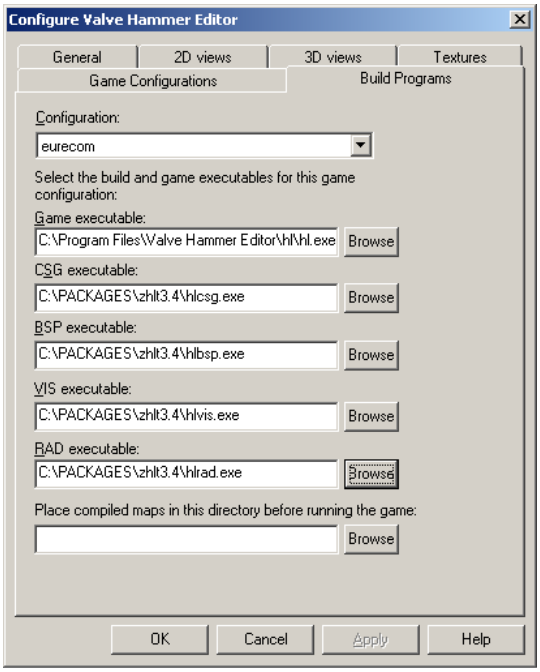


Figure 17

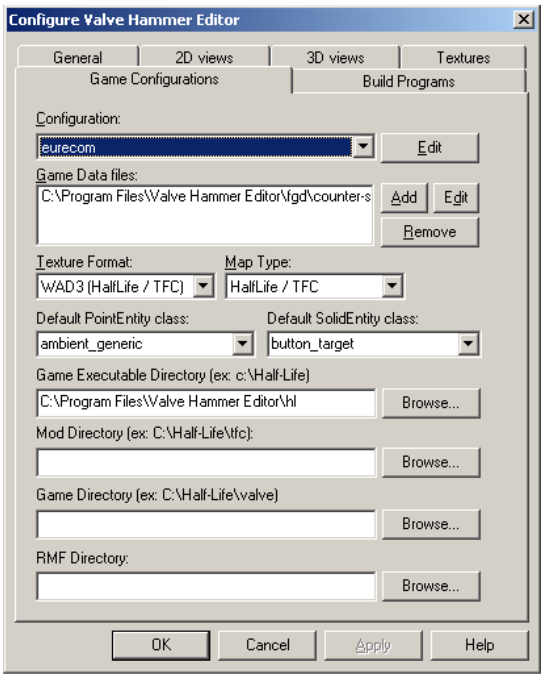


Figure 18

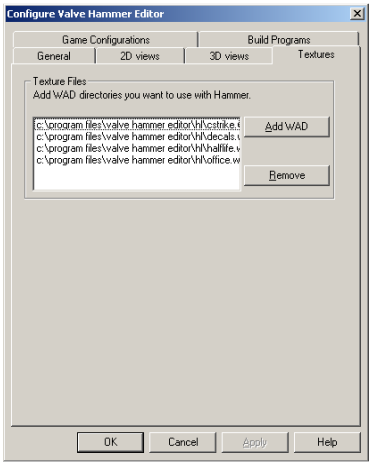


Figure 19

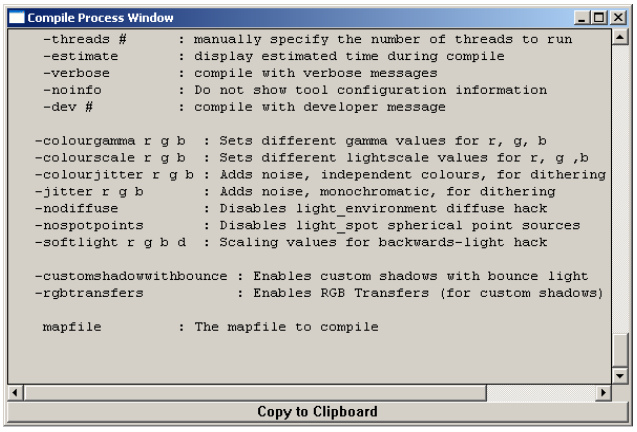


Figure 20