student ID: 10117339/UTS, 702544/UP

*Note: This is **not** the assignment I actually handed in but a shortened version. Its only purpose is to introduce you into the use and configuration of the binary files available on my homepage.*

? *Create a simple world. It has to be based on a BSP tree to efficiently represent the three-dimensional geometry. Your program for this assignment should be able to build such a BSP tree from raw geometry and store the information in a file.*
This is a lengthy process and usually done offline. Once you have this BSP file, read it and display the objects on the screen. Exploit the properties of a BSP tree to make the drawing operations fast and reliable.
Implement some ideas of your own to make the simple world a convincing experience.

INTRODUCTION

Binary Space Partitioning trees (BSP trees) became a very hot topic in the game development scene when the highly successful Quake I, published by id software in May 1996, totally changed the genre of First-Person-Shooters (FPS). Since then, almost every 3D game engine uses BSP trees to speed up the processes of rendering, collision detection, 3D sound, etc.

The file format of Quake III is very well documented, robust and wide-spread. Because the Quake III engine was used for several games, like Return To Caste Wolfenstein, Medal Of Honour, of course Quake III, and many more, there is a great variety of BSP files, called maps, available on the internet for free.

Thinking of a suitable BSP file format, I just had no choice – I had to go for the one made for Quake III ! And as it happens, the game was for sale in the local Dick Smith shops: so I bought a copy of the full game. Well, I grabbed Return To Castle Wolfenstein, too ☺. After playing for a few hours, I had the very determined feeling to write a comparable engine …

Maybe I should say that I have lots of experience in the field of 3D computer graphics. During my bachelor project, I rewrote parts of the OpenGL rendering code to meet high-performance demands of a 3D radio wave visualization tool. The whole engine was based on an open source scene graph called VRS (http://www.vrs3d.com) developed at the Hasso-Plattner-Institute, University of Potsdam, Germany. A year later, I created a tool to visualize demographic data sets of Berlin and its suburbs. Last but not least, I explored the design of the OpenGL Shading Language in early 2004, even before stable drivers were available for mainstream hardware. You can find more detailed information on my homepage: http://www.stephan-brumme.com

RUNNING THE PROGRAM

Just start the executable – you will see a simple house looking like this:

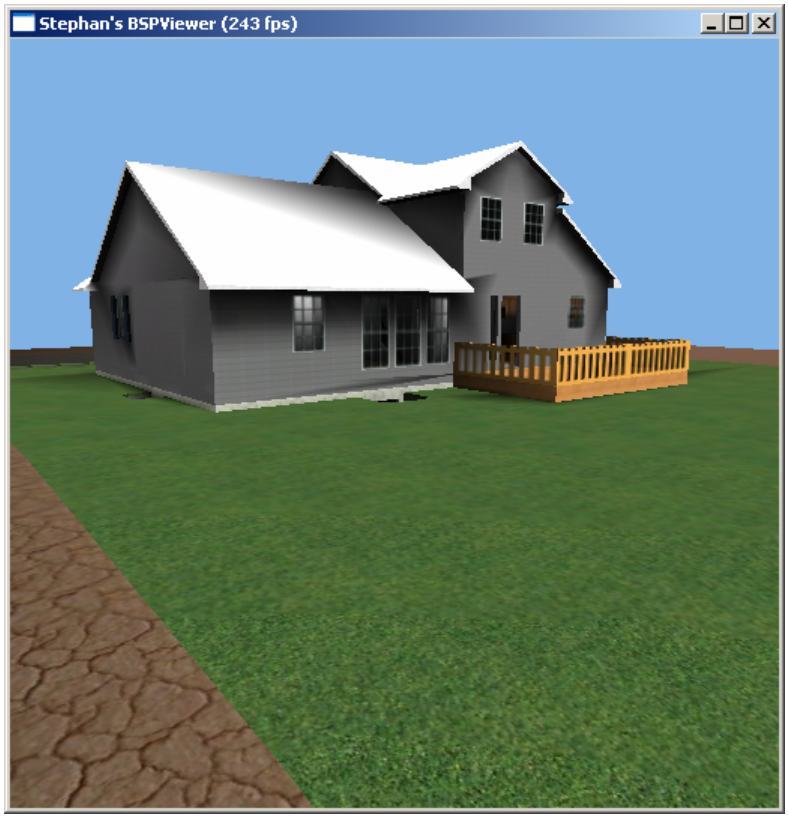

Figure 1: Screenshot House

You can explore the house by using your mouse and the cursor keys on your keyboard. There are several settings to improve and/or change the way the house is rendered:

Key	Action
Cursors	Move forward, backward, left, right
Mouse	Look around
W	Move up
S	Move down
Mouse Wheel	Move up/down
F5	Fullscreen / normal window size
F	BSP order: front to back
B	BSP order: back to front (default)
Z	Enable Z-buffer (default)
U	Disable Z-Buffer
M	Use multi-texturing → enable lightmaps (default)
N	Disable multi-texturing
P	Enable PVS
O	Disable PVS
L	Wireframe mode
T	Textured filled mode
D	Demo mode
Space	Pause demo

I recommend to press 'D' to enter the demo mode and get an impression of the whole scenery. Unfortunately, you cannot escape that mode unless you quit and restart the program. I definitely fix that issue for the next assignment.

The refresh rate on my Centrino 1.4 GHz / GeForce4MX-440 is almost always above 100 frames per second (fps). But the speed can still be improved by enabled the use of the potentially visible set (PVS). You can expect to gain between 30 and 100% performance, sometimes even much more. The effects of PVS are best visible when you switch to the wireframe mode (press 'L') because then you see some lines disappeared and popping up again.

Most settings can be changed by editing the file *config.ini*. It is plain text and should look like this:

```
[Program]
Title=Stephan's BSPViewer
width=512
height=512
statistics=0 # true => 1

[Rendering]
skipInvalid = 1
Z-Test = 1
FrontToBack = 0
PVS = 0

[Level]
package=house.pk3
map    =maps/house.bsp
#package=chartres.pk3
#map    =maps/chartres.bsp

# Quake 3 Demo files
#package=pak0.pk3
#map    =maps/q3dm1.bsp
#map    =maps/q3dm7.bsp
#map    =maps/q3dm17.bsp
#map    =maps/q3tourney2.bsp

# initial position of the camera
startx=400
starty=220
startz=-1850

[PlugIns]
renderer = OpenGLComponent.dll
engine = EngineComponent.dll
image = ImageComponent.dll
file  = FileComponent.dll
input = InputComponent.dll
sound = SoundComponent.dll
parser = ParserComponent.dll
```

If you want to explore another map then edit the appropriate lines in the [Level] section. A hash indicates a comment, whitespaces are automatically stripped. My program is not case-sensitive.

I liked to play with levels from Quake III. You can download the demo version of the game for free. To push your computer to the limit, download the chartres.pk3 map (ca. 12 MByte) from the internet and walk through the Chartres Cathedral which actually exists in France:

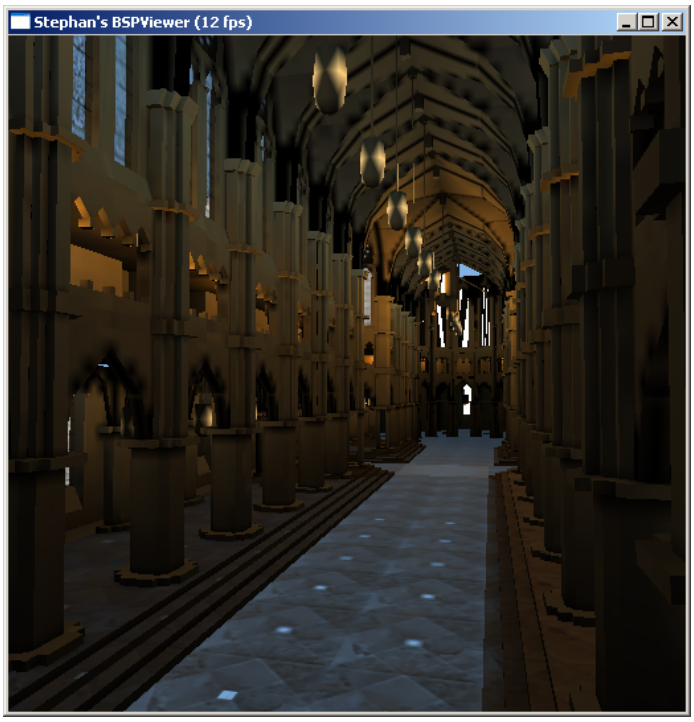

Figure 2: Screenshot Chartres Cathedral

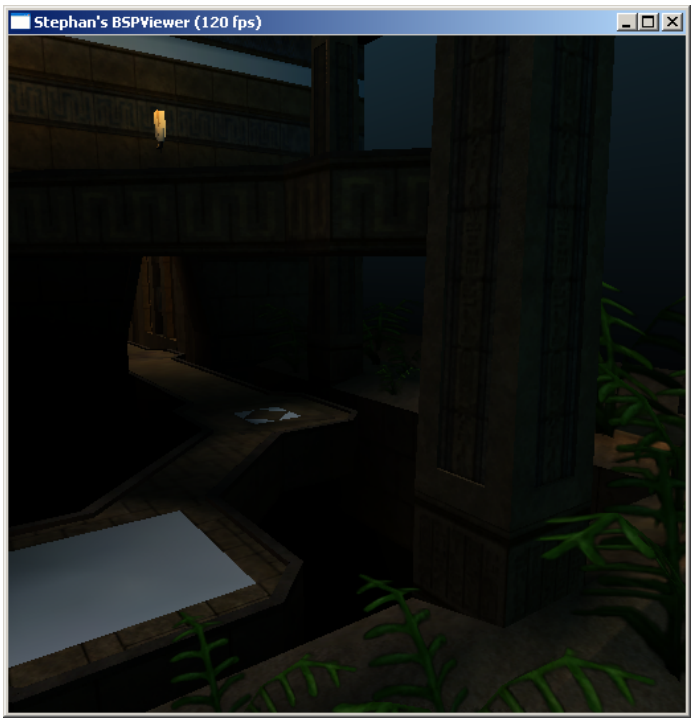

Figure 3: Screenshot AEDesert map

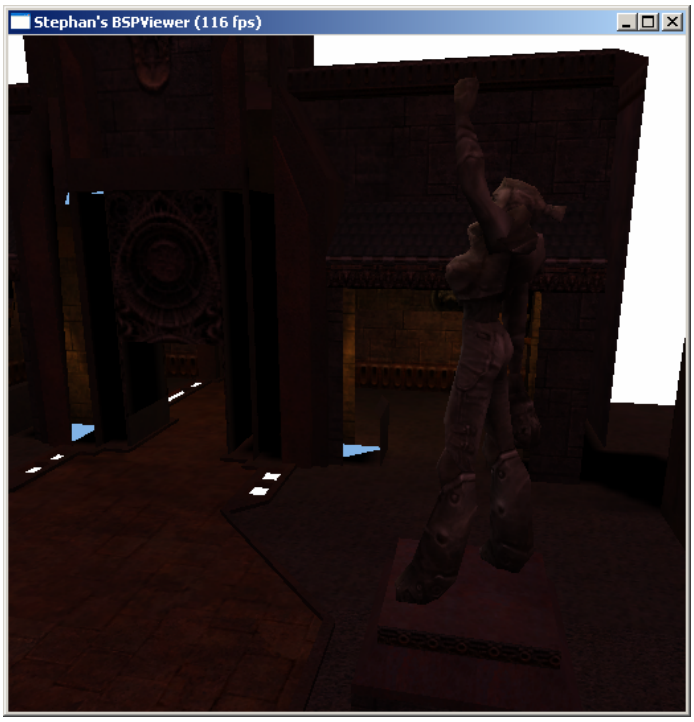

Figure 4: Quake III Arena Deathmatch1 map

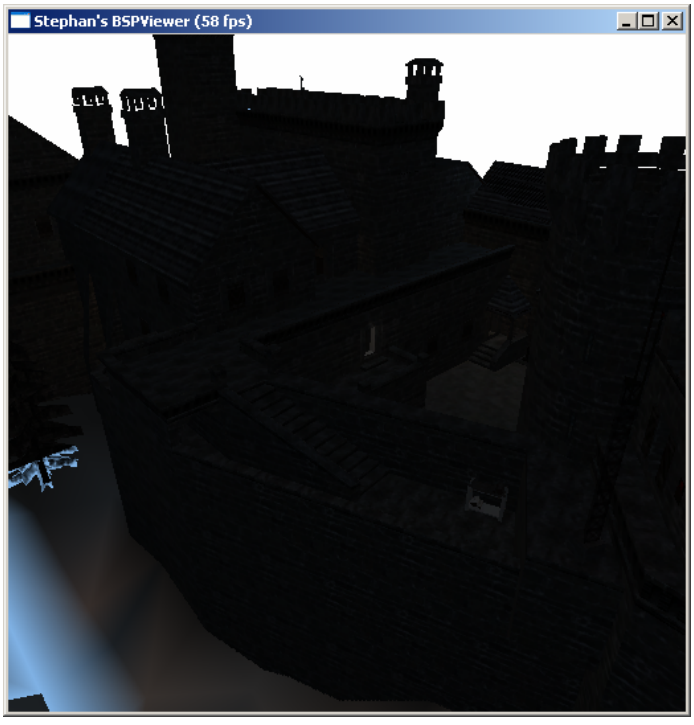

Figure 5: Return TO Castle Wolfenstein, map Escape1