VRET - Ready for the future!

Migrating the VRET system to a new development environment

Delft University of Technology
Siemen Roorda
Student number: 9706067
Delft University of Technology
Media and Knowledge Engineering
Supervisor: Charles van der Mast
June 2005
# Table of Contents

Table of contents.................................................................................................................. 3  
Introduction.......................................................................................................................... 5  
Research assignment............................................................................................................ 5  
Research outline.................................................................................................................. 5  
Time schedule...................................................................................................................... 5  
Outline of the current system................................................................................................. 7  
Requirements for the new system.......................................................................................... 8  
  Requirement list.................................................................................................................. 9  
  Research approach............................................................................................................. 9  
Overview tools and candidate applications........................................................................... 10  
Conversion tools.................................................................................................................. 10  
  Quick3D............................................................................................................................. 10  
  Crossroads3D................................................................................................................... 10  
  Polytrans.......................................................................................................................... 11  
  Writing a conversion tool............................................................................................... 11  
  Conclusion........................................................................................................................ 11  
Tools and add-ons.................................................................................................................. 12  
  DART................................................................................................................................. 12  
  Di-Guy............................................................................................................................... 12  
  Geo..................................................................................................................................... 12  
Virtual reality software.......................................................................................................... 13  
  CAVElib........................................................................................................................... 13  
  CaveUT and Unreal Engine............................................................................................. 13  
  EON Reality..................................................................................................................... 14  
  MetaVR............................................................................................................................. 14  
  MultiGen Creator and MultiGen Vega Prime.................................................................... 15  
  Quest3D........................................................................................................................... 15  
  Vizard............................................................................................................................... 17  
  Found but not reviewed................................................................................................. 19  
Overview.............................................................................................................................. 20  
Conclusion............................................................................................................................. 21  
Assigning a score to the programs......................................................................................... 22  
And the winner is................................................................................................................ 22  
Last remarks........................................................................................................................ 22  
References.............................................................................................................................. 23  
URLs...................................................................................................................................... 23
INTRODUCTION

About five years ago, the Delft University of Technology started in collaboration with the University of Amsterdam a project which goal is to treat patients suffering from several psychological disorders, such as acrophobia (fear of heights), claustrophobia and fear of flying. This project, called Virtual Reality Exposure Therapy (VRET) [8], is at the moment being extended with possibilities to treat people suffering from agoraphobia, a social disorder that is indicated mainly by fear in places where a lot of people are.

At this moment, VRET is changing from a mainly research-oriented project to an application that can be put into a commercial environment. This is proved by a Dutch health care organization, that is interested in the VRET-system to extend their market share by offering virtual reality as a new concept of treatment.

‘Going commercial’ means the application should be complete, robust, and maintainable. As a result of the research-aspect, the main focus never laid primarily on these characteristics, so in order to complete this, the application should be tested thoroughly and adapted where necessary. The big problem with this is the manufacturer of the formerly used VR-software, Sense8, has ended all its activities and can be contacted in no way anymore. WorldUp and WorldToolKit are no longer developed nor supported, and can not be installed on new systems due to the inability to get a new license code. We have one running system and a backup-copy of it, and these are the only systems we can use to access the worlds.

Taking into account these things, it is clear we should migrate to a new application to ensure continuation of the VRET-project. This research assignment is meant to make a decision which application to use. The new application will be chosen based on technical possibilities (including migration from the old WorldUp-system), quality and duration of support, and the balance between price and quality.

RESEARCH ASSIGNMENT

The research assignment is stated as follows:

- Make an investigation of the available basic virtual reality software, and examine which one is the best to migrate to. The main goal is ensuring continuation of support and technical quality, but the price is also an important aspect. Along the systems to study are Virtools, EON, GameEngine, DART and GreenDino. As the VRET application is not only used for developing new worlds but the environments should also be run in the patient’s home, simple systems must be looked at, too.

- The most promising systems should be tested, including the possibilities to convert the existing worlds to the new application. The ratio between price and quality must be judged.

- Start the research with an investigation of the current system and the current worlds.

RESEARCH OUTLINE

The first chapter in this research assignment report introduces the VRET application and the problems that should be solved and states the research assignment. Included is a time schedule.

Next, the current system is explained and the requirements for the new one are investigated.

The third chapter contains the main contents of this report, being the descriptions and comparison of the different virtual reality applications. Additionally, some tools (for making life easier or tools that really can be interesting for future research) are reviewed.

I conclude this report with a general conclusion in which I recommend a software application based on the found results.

TIME SCHEDULE

The research assignment starts at February 14, 2005. In short, the schedule will be: getting known to the current system, studying the general concepts of VRET, looking for promising applications to work with in the future (examining specifications and conversion possibilities) and implementing a conversion algorithm for one of the programs (in fact, the last point will be worked on for the course Design of Highly Interactive Systems).

A more detailed description of the time schedule is as shown in the table below:
<table>
<thead>
<tr>
<th>Week</th>
<th>Date start</th>
<th>Subjects</th>
</tr>
</thead>
</table>
| 1-3  | 14-02-2005 | - learning about the backgrounds of VRET and the current state of research  
- getting the system up and running (using Windows XP or an older version)  
- getting known to the current system (technical and practical)  
- stating requirements of the new application |
| 4    | 07-03-2005 | - looking for possible applications to migrate to  
- studying specifications and prices  
- getting demo versions, if possible |
| 5-6  | 14-03-2005 | - selecting the most promising applications  
- studying possibilities of these applications more in deep  
- getting familiar with these applications |
| 7-8  | 28-03-2005 | - studying possibilities of conversion from WorldUp to the new application  
- implementing a conversion algorithm for one of the applications |
| 9    | 11-04-2005 | - writing/finishing research assignment report |

Of course this schedule outruns: it took five weeks for the goals stated in the first row of the table, but this was mainly due to the complexity of getting the VRET system working on the new Windows XP-computers. Advantage of this was that I got complete insight in the working of the system, both VR and the linked libraries (DLLs).

The sixth week I really started looking at the available VR-applications, and due to the preparation mentioned above and the quick responses of most companies I could compare the different solutions in less time than expected.
### OUTLINE OF THE CURRENT SYSTEM

The current system, being in use for about five years, is outdated in both software and hardware. The initial version of the system was run on Windows 98 and Windows NT-systems. Via a conversion to Windows 2000, the final conversion is that to two Windows XP computers. This showed not to be as easy as just copying the right files and running the system: problems raised. Together with Frans Hooplot I solved these problems [3], resulting in a mostly working application again. At this moment, Parnassia, a health care organization from The Hague, can use VRET to really treat patients without having to wait until the system is migrated to another software environment.

The current system setup (regarding software) is as shown on the left [4].

In order to get information from the software companies delivering the candidate applications, I set up a short description of the system to send along with the request for more details. This summary is quoted below and shows the functional behaviour of the system:

#### Technical overview Virtual Reality Exposure Therapy (VRET) application

The purpose of VRET is offering a VR-treatment to patients suffering from a phobia. To achieve this, a patient navigates or is navigated through a virtual world wearing a head-mounted display, while the therapist triggers events, which can be sounds being played or lights turning on or off. Database functionality is available to save and analyze the progress the patient makes.

![Illustration 1: Overview of the original VRET system software setup](image1.png)

![Illustration 2: Screenshot of the therapist user interface](image2.png)
The patient's user interface consists of a stereoscopic view, projected in respectively the left and right 'HMD-eye'. Position and orientation is detected using an Ascension Flock of Birds tracking device. In an advanced setup, a second Flock of Birds sensor is used for navigation by patient.

The therapist's user interface (see image above) contains both the same viewpoint as the patient (upper), as an additional, free viewpoint (lower). Also world controls and session information are showed.

The interaction between the two computers is established by a socket connection (this is achieved by integrating DLL-functionality in the VR-environments; the DLLs are written in Delphi). Ten times a second data is sent both ways. From patient to therapist, information regarding location and orientation is sent. From therapist to patient, information regarding status of the worlds parameters (lights, sounds) is sent. This new values for the environment variables are read by all running instances of the viewer and the views are adapted accordingly.

To make the computer setup clear: there will be one development system, and the therapist’s and patient’s computers just run the worlds.

After Martijn Schuemie finished his PhD project, several other master students have worked on it, either to extend the number of virtual worlds, or to extend the functionality. A very short overview of the students working and having worked on it is stated below, to give a complete idea of the VRET-project:

- Martijn Schuemie: developed the complete system as a PhD thesis project; he built the claustrophobia, acrophobia and airplane worlds and user interfaces and tested them with real patients. The main framework for all phobia treatments was laid.
- Lucy Trianawaty Gunawan: extended the airplane world and its functionality and adapted the therapist’s user interface.
- Fitri Nurdini Rahayu: developed the underground world and corresponding user interface.
- Rio A Sopacua: extended the underground world and its functionality.
- Iulia Dobai: overviewed the VRET-application as a ‘third party’, uninfluenced by former knowledge, and gave advice about usability.
- Frans Hooplot: is researching the ways in which agoraphobia can be treated and will create several new worlds for this topic.
- And of course at this moment I am investigating which software to use in the future.

Requirements for the new system

According to [4], WorldUp R4 was chosen for the VRET application for the following advantages:

- It is possible to design a new world within a few days
- WorldUp supports a wide range of input and output devices
- WorldUp offers the possibility of Dynamic Link Libraries (DLLs) to be called to extend the functionality of the VR-worlds

It is clear the new application to use should meet the same requirements, or at least approach them. Not completely necessary (for the hardware is subject to be renewed within a few years time) but favorable for a clean conversion, is the support for the currently used hardware by the new application. This includes Ascension’s tracking system Flock of Birds and the nVidia Quadro4 graphics card (in principle, every graphics card can be used, as long as it meets certain requirements, such as GenLock to be able to be used with the HMD). In the future there may be a slight shift towards treatment-at-home, at which the patient will have some hardware at home. In this case, it is preferable to support cheaper hardware components (both tracker and HMD).

DLL-functionality or similar is completely necessary: databases are built and data is sent from therapist to client or vice versa by a network connection, either local or via the Internet. Therapist-to-client communication includes change of variables (lights, sound); patient-to-therapist communication is needed to render the worlds the therapist sees exactly as the patient does (location and orientation). In the future more complex functionality will be added to VRET, for example use of artificial intelligence to make the worlds more realistic.
Rapid development possibilities will be useful, too, as a commercial system will profit from the ability to create and customize your own worlds in little time and thus for little money. The new application should therefore include a modeler, or contain sophisticated import functionality.

To use the currently developed worlds, it is necessary that these can be converted to the new application, either by direct import, or using a third-party converting utility. A significant part of this research assignment will consist of this aspect. The ideal conversion includes the 3D-data, as well as the interaction rules, triggers and the included scripts that handle among others the communication.

**Requirement List**

To be able to compare the software packages objectively, I've put seven clear statements below. Besides the ones explained above, I added the price/quality balance and the question if and how continuation is guaranteed.

- What hardware is supported? In principle, the only 'strange' hardware device is Ascension’s Flock of Birds tracker. The headmounted display will just function as a dual monitor setup, and additional driver support is not needed. If the software supports HMDs, it will support our Cybermind Visette 2.
- Can our setup (therapist-patient network connection, database functionality and future extensions) be implemented in the new application? Is it possible to re-use our existing DLLs?
- How are new worlds modeled? Does the application contain a modeler, or should we use an external program like 3D Studio Max or Maya?
- What are the import features of the application? The ultimate goal is of course direct import of WorldUp-files, but none of the applications offers this (as expected).
- How are the licensing conditions? Is an additional license needed for each computer running a world (i.e. besides the development system also the patient’s and therapist’s computers), how are support and upgrades included in the license?
- What are the prices? Are there educational licenses available, and under what conditions can we use these?
- And to prevent problems like we have experienced with Sense8: what rescue plans are implemented to ensure the software keeps working after the company behind it goes bankrupt or otherwise cannot continue its activities? To get an idea of the target market and scale, how many licenses are sold, and to what kind of companies or users?

**Research Approach**

The first selection of applications will be made by literature and Internet research, by which the most general information will be retrieved. For the applications that might be interesting I'll contact the developing companies for more detailed information, for example about technical features and licensing information. With this information, I select a few most promising applications to test for real, using a trial version (if available). Based on the results of this research, I’ll recommend one application to use in the future.
OVERVIEW TOOLS AND CANDIDATE APPLICATIONS

CONVERSION TOOLS

One of the primary goals is to migrate the currently existing worlds to the new development environment, in the ideal case retaining all graphics, interactions and other multimedia elements.

Without going into too much detail, a WorldUp virtual world consists of several .NFF-files containing the separate objects in the environment (geometry, materials and textures), a .UP-file gluing all objects (but also scripts, viewpoints, ...) together, and the physic files for textures, sounds, videos, etc. From this quick overview it is clear the .UP-file should be converted into the new file format.

However, although a lot of conversion tools exist, I found no program able to read .UP-files. In fact, only two programs claimed to support .NFF-files, which is a level lower: with this, you only can convert the separate objects, meaning you have to reorganize them again into a new environment. Below these two programs, Quick3D and Crossroads3D, are described.

Besides these ‘NFF-converters’ I reviewed a general conversion tool Polytrans, and I tried to convert the worlds myself. At last note that WorldUp provides an export to VRML, but I tried it on several worlds, and all of them failed to be correctly exported. This seems to be no solution.

Quick3D [9] is a viewing and conversion tool for a wide range of graphic formats, including WorldToolKit’s .nff-format and the often-used formats of Lightwave, 3D Studio and Maya.

This tool sounds ideal for conversion between the current WorldUp-files and the new application to be used, possibly in two steps (e.g., WTK to 3DS, and 3DS to the new application). However, after having installed a demo version (which is fully functional but without the possibility to save the new file easily), none of the .nff-files was read correctly: they all resulted in an error (typically saying no error was found...). Even the simplest NFF files (just a cube without textures) threw these errors. Further testing revealed other file formats (e.g. OpenFlight and Maya) also often result in errors, which makes Quick3D a very unreliable program.

Just to be complete: Quick3D comes in three versions: Viewer, Geometry and Professional. As only Quick3D Professional claims to be able to convert both geometry, materials and textures, the other versions are not interesting to look at. An additional advantage of the Professional package would be the availability of a command line interface, making automated conversion possible. The price of Quick3D Professional is $ 199.

Crossroads3D

The freeware program Crossroads3D is able to read and write a.o. .NFF-files. The website of the creator is down, but I found [10] a working download link to the application. .NFF-files are read smoothly (with a few exceptions, like the cloud in the airplane world), and a wireframe model is shown. This can be saved and opened by another program. Textures and all other information is not read, so only the wireframe rests. But although reading of the files seems to succeed, the problem indicated in the introductory paragraph remains: you only can convert the separate models, and not a complete world.

Illustration 3: Screenshot of the Quick3D interface

Illustration 4: Screenshot of the Crossroads3D interface
POLYTRANS
Although Polytrans [11] is not able to read or write WorldUp files, I decided to survey it as it is often mentioned by the different companies as ‘the ideal program to convert file format x to our native file format’. Except for WorldUp files, it is capable of converting a great lot of formats (including 3D Studio Max, Maya and OpenFlight), eventually by buying an add on. Polytrans is priced $ 395, but for the most important 3D graphics formats (a.o. 3D Studio Max and Maya) the DCC-addon is required, costing $ 245.

WRITING A CONVERSION TOOL
Due to the problems stated above, I did a little attempt to write my own conversion routines. The .NFF file format is very clear, the .UP file format less, but easily readable. I’d like to convert to the Quick3D file format, as this format is described on the websites documentation and is relatively simple. From the Quick3D file format a further conversion to for example 3D Studio Max should be possible.

I wrote the tool in PHP for I know that language very well, using a third party class [12] to show 3D images. Reading and showing the .NFF-files was completely no problem. It appeared to be easy to read Quick3D files, too. However, writing these files was more complicated, and I got stuck on calculating vector normals and texture coordinates due to a lack of knowledge of this matter. Accompanied by someone with more knowledge of this I’ll be able to implement it to a next stage, but this will be too much effort for the little gain we will have.

Additionally, I tried to read a complete .UP-file, but this neither succeeded without spending too much time on it as the file format is nowhere documented. What did succeed, however, was reading .WUP-files (the compiled .UP-files contain all information without using additional files), which can be useful whenever we want to convert a world that only exists in a compiled version.

CONCLUSION
WorldToolKit’s file format seems to be rather unknown among the different conversion tools. I found two applications, of which the first one crashes on all .NFF-files, and the second one only reads the wireframe information. Neither of them can read a complete .UP-file to convert it at once to another format.

In my opinion there are two solutions: either the worlds are rebuilt completely, or our own conversion utility should be written. With the first option we can make a little use of Crossroads3D (converting the separate objects), but a lot of manual finetuning will remain. Writing a conversion utility is maybe ‘shooting a mosquito with a cannon’ like we say in Dutch, regarding the limited number of worlds we have. Only when we foresee more interesting parties for such an application, it can be useful. In either case only the worlds structure is converted. Interaction and external scripting should be implemented again.

Taking in mind the relatively simple worlds used in VRET, my advice is to rebuild them. Testing Quest3D, my colleague Frans Hooplot built in a few hours the firestairs (using Maya), so with limited more time for a complete world, this will be the fastest way. A further advantage of building the worlds in a very general application is the 3D structure remains conserved whenever we want to switch to another application again. Also changes can be made easily.
In this section I describe a few additional tools I encountered when looking for the new software application. Although the focus of my research assignment is not upon this subject, it is a waste of time not to include them in my report, while in the future maybe these add-ons can be interesting in the view of further research.

**DART**
DART, the Designer’s Augmented Reality Toolkit [1, 12] is an educational project developed at Georgia Tech. DART is no VR application on its own, but is an extension to Macromedia Director and Shockwave3D [17] that enables one to create augmented reality applications: virtual content in the real world, or vice versa. DART uses the VRPN toolkit [18] to add tracker- and HMD-functionality to Director. The DART libraries are freely available from the website.

The question will arise why I discuss DART in the *Tools and add-ons* section rather than in the *Virtual reality software* section. The reason for this is DART will be very interesting from a researcher’s point of view, but as it is ‘just’ an educational project, it will not suit our requirements for the successor of WorldUp that are based on proved results, secured future and user friendly interfaces (although this last seems to be pretty good).

However, for future projects within Delft University of Technology, DART may be interesting for looking in what way augmented reality can improve treatment of phobia patients: does it enlarge the feeling of presence, is development of new worlds easier? Cooperation with one of the developers (Jay David Bolter) is accomplished yet in other subjects by Charles van der Mast.

**DI-Guy**
Di-Guy is [14] a general API that adds realistic people to virtual environments running OpenGL. Among the VR-programs that have integrated Di-Guy seamlessly are MultiGen Vega Prime and MetaVR. Almost 100 different characters are included, together with over 2,000 motion and transitions. Di-Guy is widely used, for example (like a lot of VR applications) in the US Army. Di-Guy costs $1,500, regarding the pricelist MetaVR offers.

In the scope of this research assignment, Di-Guy is not that interesting to look at any further, but maybe it can prove a very useful extension for treatment of social phobias like agoraphobia where environments containing other people are necessary. A trade-off should be made between the costs of this API and the effort it saves compared with manual creation and animation of humans (or, if available in the software, the built-in characters).

**GEO**
For a lot of applications do not have a modeler included, an external 3D program is required to create the virtual environments. Besides 3D Studio Max and Maya for the ‘default’ 3D objects, often MultiGen Creator (described later on in the report) is mentioned, its OpenFlight format being said to be the leading industry standard for virtual worlds. Geo (Pro) [15] is an alternative for MultiGen Creator, priced half the amount if the 3DS and VRML plugins are included ($10,170 vs $4,725).

Illustration 6: Di-Guy characters

Illustration 7: Screenshot of Geo
VIRTUAL REALITY SOFTWARE

This section contains descriptions of the actual software applications that may be the follow-up for WorldUp and WorldToolKit. The software packages are mainly found by searching and browsing the Internet, just a few hardcopy resources are used.

Starting with an initial list of software programs, I encountered ever more programs. Quiet a lot of them are ‘just’ 3D applications, without the possibility to extend its functionality to virtual reality. The ones that do, are described below (in alphabetical order).

CAVELib

The CAVELib [16], a product of VRCo, is not a scene graph, or rendering engine. It uses OpenGL or OGL based libraries for scene creation and rendering. The CAVELib is primarily the windowing and camera handling module for immersive displays. The CAVELib makes sure an application written on top of it displays with correct viewer centered perspective for a multi windowed display, like a cave or HMD. It also talks to the trackd daemon (small program running in the background) for obtaining tracker and input controller information from a.o. Flock of Birds.

For the purposes of this research assignment the CAVELib is of no interest, as completely no additional functionality is present: it is just a wrapper for a graphics library, where the latter should have all functionality as networking, modeling or anything else.

To be complete for the overview of applications, I mention the license costs of the CAVELib as VRCo sent me. The development system should have a full license, priced $ 14.900 ($ 8.900 educational); the patient’s computer needs an immersive and trackd license, costing $ 8.200 ($ 4.900 educational); and for the therapist’s computer a desktop license ($ 700 resp. $ 420) suffices. Additional costs are for annual maintenance (first year is free), being resp. $ 1.260, $ 490 and $ 42. This maintenance includes support and updates.

The CAVELib is licensed to VRCo from the University of Illinois, and depending on their policies the source code can be made public or not whenever the project stops.

CaveUT and Unreal Engine

CaveUT [6] is an open source modification of the Unreal Editor game development tool [31] which extends the functionality of the latter with possibilities to show a virtual world in a CAVE. For the fall of 2005 stereoscopic display support and motion tracker support are scheduled, which makes it an interesting project to look at for the VRET application. The core idea behind it is the multi-player functionality of the game engine: to be able to project the world in a two-wall cave, create two ‘players’, let them look in a perpendicular direction and project both views on the two screens. Stereoscopic viewing is likewise possible, by generating two views in one direction, slightly shifted, and project these views into a headmounted display.

Although Unreal Engine was primarily developed to create games, every world is possible. The only situation in which it cannot be used appropriate is for applications (e.g. scientific visualization) where shapes have to be generated on-the-fly from a data stream or some algorithm. At this moment, this is not used in the VRET application, but maybe in the future, for example when extending VRET to multiple application levels, it can be an issue.

Viewing the properties of the Unreal Editor, it appears to be a very complete program, offering a WYSIWYG (‘what you see is what you get’) editor, 3D Studio Max and Maya importers, C++ scripting language and native network functionality. Regarding usability, they are proud of having ‘all the other niceties you’d expect from a modern content editing tool: multi-level undo/redo, drag-and-drop, copy-and-paste, customizable key and color configuration, viewport management’.

As the gaming industry is a giant one, in which billions of euros are spent and earned, and multiple thousands, maybe millions licenses are sold, this is a certain security for the future. But it does not
guarantee the Unreal Editor will always remain open source. This is a requirement for the continuation of CaveUT.

The costs of CaveUT are low: only a Unreal Tournament license is needed, costing $ 40. The open source CaveUT-software is free downloadable, as is the VRGL-module used by it.

Configuring a cave-setup is quite difficult: one has to calculate all rotation, translation and field-of-view parameters and change them in the configuration file of CaveUT. If this is more or less difficult for setting up a HMD-configuration, cannot be said at this moment as this extension is not yet available. If tracker configuration is just as complicated, will be shown when tracker support becomes available. Note it is not known yet which motion trackers will be supported!

EON Reality [19] relies as different other programs on an external modeler: over 55 modeling formats, including 3D Studio Max and Lightwave, are importable. For integration with 3D Studio Max EON Raptor is developed. Writing our own conversion program is not possible, as the file format is closed. Virtual reality support is offered by EON iCatcher, supporting Flock of Birds and headmounted displays. For scripting (C++), including network support, the EON SDK should be bought.

In case of a bankruptcy or otherwise discontinuation of activities (although EON denies they plan to), there should be no problem as the license is perpetual. After asking details, it shows the license key is OS-independent, but reads the harddisk or network ID. Re-install on a new computer will therefore not be possible. The source code will not be released, and other certainties are not available, thus no real security is offered.

Licenses for the different applications are as stated below. These prices are for educational use; commercial licenses are twice the price. Annual maintenance includes support and upgrades and counts for all three modules together.

<table>
<thead>
<tr>
<th>Application</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>EON Studio, including Raptor</td>
<td>€ 2,500</td>
</tr>
<tr>
<td>EON iCatcher:</td>
<td>€ 12,500</td>
</tr>
<tr>
<td>EON SDK:</td>
<td>€ 5,500</td>
</tr>
<tr>
<td>Immersive viewer</td>
<td>€ 3,995</td>
</tr>
<tr>
<td>Annual maintenance:</td>
<td>€ 3,074</td>
</tr>
</tbody>
</table>

MetaVR's [20] main focus is clearly on military projects, especially for training purposes. The company delivers complete VR-solutions, including hardware and pre-assembled computer systems. Luckily not only the hardware offered by MetaVR is supported, but both the Flock of Birds we possess and general HMD’s, too [2].

MetaVR consists of different modules, of which the VR Scene Generator (VRSG) is the most important. This module loads a 3D-environment in MDX-format and makes it interactive. The 3D-environments can be created by either MetaVR’s WorldPerfect, or a program capable of creating OpenFlight models, followed by a conversion to MDX. Another possibility is to use provided (and mostly free) MDX-databases. However, in the scope of this
Overview tools and candidate applications

research assignment, this latter is no option as the number of objects and scenes will be limited at least to a certain scope.

To create landscapes, WorldPerfect is a great (and expensive) module. It is even possible to get a very detailed (0.3 metres) terrain database of the complete United States. Other, more complex 3D-objects cannot be modelled with this application, so you have to use another program, and after that convert the object to the OpenFlight format using a converter tool like Polytrans [11].

The costs for the VRSG and WorldPerfect are listed below, as found on the MetaVR website. As MetaVR did not respond to my e-mails, I don’t know if educational licenses are possible and at what rate. Regarding the price of WorldPerfect, a better overall solution will be VRSG combined with MultiGen Creator or Geo, or just use another 3D modeler and buy Polytrans.

<table>
<thead>
<tr>
<th>Software</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetaVR VRSG:</td>
<td>$ 9,000</td>
</tr>
<tr>
<td>MetaVR WorldPerfect:</td>
<td>$ 19,500</td>
</tr>
</tbody>
</table>

As an additional remark, note that DI-Guy [14] can be implemented in MetaVR, resulting in real-like human objects.

**MULTI GEN Creator and Multi Gen Vega Prime**

MultiGen [21] produces a.o. the components MultiGen Creator and MultiGen Vega Prime; combined, this is a VR-environment as we need it. Creator is the modeler tool, and Vega is the rendering engine. Included in Vega Prime is a C++ API, which makes network functionality possible. As an extension, DI-Guy [14] can be seamlessly integrated into Vega Prime for realistic avatars.

As on the MultiGen website both Vega and Vega Prime are mentioned, a little explaining sidenote: MultiGen offers Vega and Vega Prime; the difference between these is that Vega is built on top of SGI’s OpenGL Performer [22]. This implies in principle only support for Unix-systems (the only operating system Performer worked at the time of development), but MultiGen ported the functionality to the Windows platform. Four years ago, MultiGen decided to create a new Vega, called Vega Prime, on top of their own software. The advantages are clear: platform independent and complete integration with all functionality of MultiGen products. Vega is still sold, for some customers are using it, but Vega Prime is the recommended product.

For hardware support, MultiGen uses VRCO’s [16] trackd module, which is also included in CAVELib. This module supports a lot of hardware, including headmounted displays and Ascension Flock of Birds.

Licenses are priced as follows, without difference between commercial and educational licenses:

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creator:</td>
<td>€ 10.170</td>
</tr>
<tr>
<td>Vega Prime:</td>
<td>€ 10.500</td>
</tr>
<tr>
<td>Vega Prime Runtime:</td>
<td>€ 1.100 each</td>
</tr>
<tr>
<td>2nd year maintenance:</td>
<td>€ 4.410</td>
</tr>
</tbody>
</table>

The maintenance includes all support and upgrades.

MultiGen has not taken measures against bankruptcy or something similar but, as MultiGen says, ‘the Creator and Vega Prime software are used on a large number of aerospace and defence contracts and many are on 5, 10 and 15 year maintenance programmes. This does not guarantee the software will be maintained and upgraded that long but it does provide a sound business reason for the software to be maintained for at least the current duration of programmes using the software.’ There are mechanisms to hold the software in security but they are very costly. If really wanted, we can get a quotation for this.

**Quest3D**

Quest3D [23] is a complete multimedia authoring tool developed by the Leiden company Act3D since 1998 as an in-company production tool, that showed such a potential to their customers that they invested in it and released a commercial version in 2000. As from version 3.0, launched just April 29 of this year, the product line is extended with a VR edition, exactly what I am looking for. The edition one level lower (and cheaper) is the Enterprise edition, but this lacks support for motion trackers and stereoscopic projection. As a result of the brand new version, not all hardware is supported yet. Among this unsupported hardware is the Ascension Flock of Birds tracker. In the contact I had with Act3D it showed that support was scheduled. Maybe coincidence, maybe as a result of my question, Flock of Birds support was mentioned to be in a next beta version and in either case in the final version. This deadline is not reached, however, and although Act3D claims to be working on it (‘ready within a few weeks’), some additional pressure from our side might speed up things a little,
especially when VRET goes ‘commercial’ and we buy the non-educational version.

The price of the Quest3D VR Edition is $9,000; this price is all-inclusive: there are no hidden costs for distribution or support, and all upgrades up to version 4.0 are included. Creation of stand alone executables is possible, for which no additional license is needed (still all functionality is included). A non-commercial license costs $229, which makes Quest3D one of the cheapest programs I looked at. As long as VRET is only used within the university, we can profit from this low price.

An interesting point in the view of continuation is the existence of a so-called escrow-contract: the source code is deposited in a central institute; if Quest3D goes bankrupt or otherwise discontinues its activities, the source code is released. For this a monthly insurance fee is required, being €1,000. Quiet high, but it is nice to see Quest3D has a rescue plan. It reduces the chance all knowledge and support will be lost. Information about exact numbers of sold licenses and types of customers can not be given, but Quest3D has ‘thousands of customers, mostly for the Lite Edition; a lot of Professional customers, too, and less Enterprise customers’. The number of Enterprise (and expectedly VR) customers is raising.

As mentioned before, version 3.0 will be the first version including a VR edition. Because of this, no one can tell of what quality this program will be, nor what the development stage is. However, several Internet resources praise the quality of the 3D products (Quest3D up to the Enterprise edition), giving confidence to what will come. A nice thing of the Quest3D website is the demo-section, where not only the ‘look how good Quest3D is’-demos are available, but also technical demos that show what is possible with the program. I planned a visit to Leiden with Frans to talk about some technical details that are necessary if VRET will be converted to Quest3D. In this meeting, I’ll try to get more clear information about Flock of Birds support, too.

Testing

Together with Frans Hooplot I tested Quest3D (fully functional, limited time license 3.0 beta version and final version), initially guided by a very simple online how-to and a lot of trial-and-error. This seemed however not to be the right way, for Quest3D has a steep learning curve, and luckily a comprehensive tutorial manual was included in the evaluation download. This manual introduces the basic and more advanced features of Quest3D in easy to follow, step by step examples.

At the first glance, as one can see by illustration 10, it is clear Quest3D is no real 3D modeler. It is possible to create scenes by placing primitives in the scene and adjusting their location, orientation and size; lightning and textures can be altered too. More complex objects will most likely be created in a more sophisticated program as 3D Studio Max, Maya or Lightwave. For these and a few other programs, Quest3D contains exporters to the Direct3D .X-format. Native Polytrans [11] import functionality is included, too. I tested importing from a Maya file, and everything was imported well.

The basic building blocks in Quest3D are the so-called ‘channels’. A channel is a sort of black box, requiring some input, and producing some output. Quest3D contains a lot of default channels, making it possible to create a wide range of programs. If these default channels are not enough, the C-based Lua scripting language [32] can be used to create new channels with all dedicated functionality that specific program needs. But, as said, most functionality is available by default: camera’s, lights, pre-rendered objects, particles, logics (conditional rules, binary operators, user input, pathfinding), timers, sound, physics, database and networking capacities. In illustration 11 the basic setup of a program can be seen: from the Start-channel a render loop is called, containing a camera, a light, and a primitive object (in this case a sphere). Two logical IF-channels are added for user input. The more complex a program becomes, the more jungle-like the channel window becomes, each channel linked to another one, having a lot of children, etc.. Luckily it is possible to create multiple channel
Overview tools and candidate applications

Testing is very easy: Quest3D is a ‘wysiwyg’ editor: what you see is what you get. All changes in the program are immediately visible in the scene, even if the program is running. After the program is finished and tested, it is ready to compile into a standalone executable. The only additional requirement to run it on another computer is the availability of DirectX, but each Windows version has it by default. Other ways of publication are: a player file (the Quest3D player can be downloaded for free), a web application or a screensaver. Neither of the distributions is bound to additional licenses.

Virtools [24] is a complete 3D development application (the first version is released in 1999), consisting of different modules. Virtools Dev is the core element for creating the 3D environments, including all actions and scripts. Virtools VR Pack is necessary for support of trackers and HMDs. Based upon the open source standard VRPN [18], Flock of Birds and a lot more hardware VR-devices are supported. Other modules include a.o. an AI Pack and a Physics Pack, but are less interesting to look at at this particular moment. Maybe in the future these are interesting extensions. If we do not want to rewrite our communication libraries into C++, the multi-user package (Virtools Server) is available for this functionality. However, regarding the high costs of it this is no option and we will rewrite the code.

Virtools is capable of importing three 3D image formats: the native .nmo-format, 3D Studio Max and DirectX. CAD-import is possible, too, but an additional module is required for it. The .nmo-format is not open for writing our own conversion tool, but the source code of the 3D Studio Max exporter is. Frans Hooplot has had contact with Virtools earlier and was told maybe Virtools would implement a WorldUp-conversion tool. However, in reply to my questions, this proved not to be true.

A free webplayer is available for viewing simple Virtools objects and scenes, but for actually running the programs offline, and for viewing applications made with the VR Pack, a licensed player must be bought. The educational prices are as follows:

- Virtools Dev: € 6.000 (10 licenses)
- Virtools VR Pack Dev: € 4.000 (10 licenses)
- Virtools VR Player: € 2.000 each
- Virtools Server: € 12.000 (10 licenses)
- Virtools Publishing Server: € 10.000

The last two modules (Server pack) are used for multi-user environments. However, writing our own communication classes, these modules are not needed. The educational licenses can only be bought in multiples of ten.

Virtools aims at three main markets, being VR and visual simulation (55%), games (25%) and multimedia and marketing (20%) [7]. Among the customers of the 10,000 sold licenses are large companies as Electronic Arts, Microsoft and Warner Bros, but also the educational market is represented with 20% of the sold licenses. More than half of the total 3 million dollar revenues is generated in Europe (headquarters is located in Paris, France); Asia and America (north and south) follow with respectively 30% and 15%. To support their customers whenever Virtools stops developing its software, the source code will become public.

Illustration 11: Structured channels of a simple Quest3D program
As an interesting fact, on the Internet Virtools and Quest3D are often compared, even on the Virtools forum [25] itself (though mainly for their 3D capacities). In most reactions Quest3D appears to be faster and even better, and a very often mentioned difference is the licensing structure: Quest3D is free for distribution, whereas for Virtools a large amount should be payed.

Testing

A demo version of Virtools, including the VR Pack, is sent to me for testing the program. After having followed one of the two tutorials included in the user’s manual, Virtools appears to have a similar way of working as Quest3D. Maybe this is the reason these two programs are compared so often, as stated above.

But of course there are differences, starting at the look and feel: Quest3D uses the Windows look and feel, which is (or at least has become) very intuitive, and navigating in a 3D world is performed like most programs do (scrolling to zoom, alt+mouse for rotation, control+mouse for moving). Virtools however requires to select from the tools panel the ‘rotate camera’ icon, and then rotate the view; moving and zooming requires another tool to be selected. A second difference is the user’s manual: this is far more abstract and therefore makes the threshold to actually learn and use Virtools a lot higher. As a last difference I re-mention the way of publishing your program: the only way to play your program on another computer is by installing the Virtools player. This is an additional action to be performed, and again the cost aspect is a disadvantage.

Like in Quest3D, Virtools enables the possibility to adapt the scene while running. This makes testing and debugging easy. Like the channels in Quest3D, Virtools uses ‘building blocks’ that can be linked to eachother to more complex actions. The C-like Virtools Scripting Language (VSL) can be used to create more advanced building blocks.

VIZARD

Vizard [26] is developed as part of huge research projects at the University of California and MIT and has a strong link with NASA. Advantage of the binding with these universities is that in case WorldViz ever stops the development of Vizard, all rights fall back to these institutes which have a strong interest in making the software available to the general public.

Vizard doesn’t include a modeler, but makes use of existing models which can be imported, for example VRML, 3D Studio Max and Maya. Regarding implementation of virtual worlds, graphical options are very rare, and everything (configuration, interaction, etc.) is done by programming a script in the Python language, although soon a C++ SDK will be available, too. This sounds difficult, but (partially due to a few
very good tutorials) I managed to get used to Vizard really fast (see a little further where I discuss my experiences with Vizard). Advantages of this ‘hardcore programming’ is that all functionality of the Python programming language is available in all virtual worlds, so networking, database functionality and the AI-elements to be developed in the future can all be integrated seemlessly. Hardware support is very complete: different libraries are pre-configured, as is the Flock of Birds, and with use of the VRPN-library all supported hardware can be used (although it requires a little scripting). Headmounted displays are no problem: it is very easy to choose for stereo displays or even anaglyphic (‘red/green’) projection. This last type of rendering is maybe interesting for the treatments at home, where the costs of the application should be reasonably low. An additional research should make clear if this monochromatic projection does not lower the feeling of presence too much.

An educational license is priced $5,199, runtime licenses cost $449 each. However, these runtime licenses can be bought in bundles, lowering the price down to approximately $75 each (an average set of 5 licenses costs $745). A multi-seat license of the enterprise edition is also available, priced $8,575 for 5 seats. Support can be bought for 12% of the total amount spent to the licenses (yearly) and includes all upgrades.

Testing
Vizard is one of the applications I actually tested (fully functional trial version valid for 90 days) and within a day I was able to create the type of worlds we need to have. One day further doing tutorials and reading documentation delivered an ‘almost-ready’ VRET application, of course not completely finetuned, but reasonable working including Flock of Birds support and therapist-client communication. Maybe it is an advantage my background is mainly code-based, so that I could find my way in the Python language very quickly.

Compared to Virtools and Quest3D, Vizard is far more ‘back to basic’ as no fancy user interface is available. Where the other two programs offer basic tools to create or adapt the 3D scene, Vizard requires coding to do so. Textures, rotation or position of objects can be changed, new primitives or simple objects can be created and managed, but not as intuitive as in the other programs.

A disadvantage, or maybe an advantage?, of Vizard is that it in some cases seems a ‘test project’: the support is really good and fast (Vizard misunderstood a question of mine regarding the Flock of Birds, and immediately bought a second Flock of Birds to add multi-bird support), but provided by a very limited number of users. Additional features requested by users are implemented in no-time, or studied with real interest (as upon my question if the titlebar could be removed from a running instance).

FOUND BUT NOT REVIEWED
Besides the programs described above, I found some more applications that seemed interesting at first sight, but proved to be not exactly what we need. To complete the software overview, I list these programs very quickly below:

- Quantum3D Mantis and Quantum3D VTree [27] - upon reply to my e-mail for more information it seemed these products were not the right applications for VRET, for ‘it is aimed at the Visual Simulation and Training markets, rather than virtual reality, and hence is missing some important functionality you may require’.
- VR Juggler [28] - like DART a research project of the Iowa State University. As the program is very low-level, both developing and running (e.g., a compiler, CppDOM, GMTL should be pre-installed), it is not the right application for a user-friendly VRET-application.
- MÄK [29] - like Quantum3D more a simulation and training environment, aiming mainly/only on the military market.
- Apex DAS [30] - the same reason as Quantum3D and MÄK
## Overview

The table below shows a quick summarizing overview of all reviewed software applications.

<table>
<thead>
<tr>
<th>Name</th>
<th>Modules needed</th>
<th>Price1</th>
<th>Distribution2</th>
<th>Modeler</th>
<th>Continuation guarantee</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAVELib</td>
<td>CAVELib trackd</td>
<td>E: $8.900 C: $14.900</td>
<td>$4.900 each</td>
<td>Not included (OpenInventor)</td>
<td>Depends on University of Illinois</td>
</tr>
<tr>
<td>CaveUT</td>
<td>CaveUT Unreal Editor</td>
<td>E/C: $40</td>
<td>Free</td>
<td>Unreal Editor</td>
<td>-</td>
</tr>
<tr>
<td>EON Reality</td>
<td>Studio ICatcher 3D</td>
<td>E: €23.500 C: €47.000</td>
<td>€3.995 each</td>
<td>Not included</td>
<td>Perpetual license</td>
</tr>
<tr>
<td>MetaVR</td>
<td>VRSG</td>
<td>E/C: $9.000</td>
<td>?</td>
<td>WorldPerfect $19.500</td>
<td>-</td>
</tr>
<tr>
<td>MultiGen</td>
<td>Creator Vega Prime</td>
<td>E/C: €20.670 C: €1.100 each</td>
<td>Included</td>
<td>None</td>
<td>-</td>
</tr>
<tr>
<td>Quest3D</td>
<td>VR Edition</td>
<td>E: €229 C: €9.000</td>
<td>Free</td>
<td>Included</td>
<td>Escrow contract</td>
</tr>
<tr>
<td>Virtools</td>
<td>Dev VR Pack</td>
<td>E: €10.000 C: €2.000 each</td>
<td>Included</td>
<td>Source becomes public</td>
<td>-</td>
</tr>
<tr>
<td>Vizard</td>
<td>Enterprise</td>
<td>E: $5.200 C: $8.000</td>
<td>$75 - $450 each</td>
<td>Not included</td>
<td>Probably source becomes public</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Platforms</th>
<th>Importable formats3</th>
<th>Hardware support4</th>
<th>Scripting language</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAVELib</td>
<td>Windows Unix Irix</td>
<td>OpenInventor</td>
<td>trackd</td>
<td>C++</td>
<td>Just wrapper for other GL’s, not what we need</td>
</tr>
<tr>
<td>CaveUT</td>
<td>Windows Linux MacOS</td>
<td>3D Studio Max Maya</td>
<td>Currently neither HMD nor FoB</td>
<td>C++</td>
<td>-</td>
</tr>
<tr>
<td>EON Reality</td>
<td>Windows 3D Studio Max Lightwave</td>
<td>HMD Flock of Birds</td>
<td>Visual C++</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>MetaVR</td>
<td>Windows OpenFlight</td>
<td>HMD Flock of Birds</td>
<td>-</td>
<td>Di-Guy can be integrated</td>
<td>-</td>
</tr>
<tr>
<td>MultiGen</td>
<td>Windows Linux Irix</td>
<td>-</td>
<td>trackd</td>
<td>C++</td>
<td>Di-Guy can be integrated</td>
</tr>
<tr>
<td>Quest3D</td>
<td>Windows 3D Studio Max Lightwave</td>
<td>HMD Flock of Birds</td>
<td>C++</td>
<td>No FoB support yet</td>
<td>-</td>
</tr>
<tr>
<td>Virtools</td>
<td>Windows 3D Studio Max DirectX</td>
<td>VRPN</td>
<td>C++</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Vizard</td>
<td>Windows 3D Studio Max Lightwave</td>
<td>HMD Flock of Birds</td>
<td>Python C++</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

---

1 Price is for an all-inclusive development environment (modeler excluded, if not part of the application). The first noted price is the educational license; the second one is a commercial license.

2 Costs of distribution of the applications. Either free, or the costs of additional licenses.

3 Noted a maximum of 2 formats; most applications can handle more.

4 Trackd [16] or VRPN [18] means the program makes use of the respective general library and can handle multiple trackers, a.o. Flock of Birds and HMDs.

5 Flock of Birds support is not finished yet, but will be soon (‘within several weeks’).

6 In the future, Virtools will support multiple platforms.
CONCLUSION

Looking for virtual reality software environments, a whole lot of applications can be found. Most of them are not interesting to look at (for example, because they define just 3D as being VR), but a few deserve a closer look. To be able to compare them, I looked in what way they meet the criteria stated in the requirements chapter. First a general overview of these criteria is given, after that I try to fit the separate applications into these requirements to be able to compare them in an objective way.

Hardware support
The Ascension Flock of Birds seems a very general tracking system: almost every application supports it. Headmounted displays are no problem at all, which is predictable as it is ‘just’ an output device like a monitor. Most applications are able to handle multiple hardware devices, and sometimes third party libraries are used: VRCO’s trackd and VRPN are used by different programs. Often support for your own hardware can be programmed by yourself with the use of the SDKs. (Switching of) hardware will be no problem, neither now, nor in the future.

Only Quest3D does not support the Flock of Birds yet, but are willingly to add this as soon as possible, especially if we become an interesting party.

Software development
As stated before, it is completely necessary to be able to use our own extensions to the application, as the VRET-system is far more complex than the average VR application. Luckily all applications include a software development toolkit (SDK) or something similar. The solely used programming language is C++, with sometimes an additional language (Java, Python); Delphi is nowhere mentioned. Usage of our existing DLLs will therefore not be possible, in neither of the applications, which implies a rewriting of this code.

Rapid development
It is very difficult to make an estimation of the time it will take to develop a brand new world. Some applications include a modeler, but most rely on third party software such as 3D Studio Max or MultiGen Creator. Depending on the skills of the designer, I don’t think there are significant differences in design time, especially as the scenes are relatively simple (i.e., at this moment; maybe in the future more sophisticated worlds will be built, but for this, one should keep in mind mostly import from other 3D programs is possible). After the world modeling interaction should be added and scripts written or at least adapted. The limited number of practical tests I was able to perform is insufficient to say which one is the quickest to implement, and I guess after a few worlds creation is done smoother and faster. Most of the time will be in designing a new world, the stage before even touching a computer.

Conversion of existing worlds
The WorldUp file format is by neither of the programs supported as importable format. Export from WorldUp to another format is also not possible (VRML is supported, but only a sometimes wrong rendered wireframe is exported), and the conversion tools are not working in an appropriate way. Two options remain: write our own import module, or recreate the virtual environments. Maybe a combination of these two is the best way.

Pricing and licensing conditions
When looking at the costs of the solutions, big differences are seen: the cheapest application is priced $ 40, the most expensive $ 43.750 (all educational licenses - the relative differences between commercial licenses is large, too, but not this exaggerated). The latter even lacks a modeler: if the price for MultiGen Creator is added, almost $ 55.000 is the total amount, which is a factor 1.400 (!) more. The price of the overall solution is certainly an important issue in deciding the software to use.

Besides that, it differs how end products can be distributed. Several applications can create stand alone executables that can be used for free, others require a player and/or additional player-licenses priced up to $ 5.000 each. This is completely overdone, as in the nearby future patients will be treated at home. Additional costs of multiple thousands of euros above the price of the hardware are not possible.

Continuation insurances
Most companies are very reserved about their future prospections. They all show how business is doing now, how fast they are growing and what important names they have in their customers range.
Conclusion

A few companies do have a sort of insurance when they stop development and support, either by making the source code open, or by offering a so-called escrow-contract. The costs of the latter are too high to be a serious option for us at this moment, but there will be a chance another company takes over the activities or support of this program.

Assigning a score to the programs

In the table below I try to assign numerical scores to the different criteria. As this is very difficult (for a real comparison one should know all programs thoroughly), I'll not blindly choose the program with the highest score but base my conclusion on more weighted decision. The possible scores range from 1 ("bad") to 5 ("very good").

<table>
<thead>
<tr>
<th></th>
<th>CAVElib</th>
<th>CaveUT</th>
<th>EON Reality</th>
<th>MetaVR</th>
<th>MultiGen</th>
<th>Quest3D</th>
<th>Virtools</th>
<th>Vizard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware support</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Software development</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Scene development</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Conversion and import</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Price</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Licensing conditions</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Continuation</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>21</td>
<td>29</td>
<td>19</td>
<td>19</td>
<td>23</td>
<td>31</td>
<td>29</td>
<td>31</td>
</tr>
</tbody>
</table>

And the winner is...

Starting my research assignment, my idea was VRET is a very complex application, which requires very high-level development software. After going deeper into the matter, it proved not that complex at all: the virtual reality was not top of the bill, just functional (just as it has to be). And the interaction and scripting around it neither is that difficult: all programs support extension development in a higher level language.

Taken this into account, together with the price and licensing information, I dropped the most expensive (and complex) applications, being CAVElib, EON Reality and MultiGen. MetaVR is not interesting according to the details mentioned in the former chapter, leaving Quest3D, Virtools and Vizard.

Mainly due to its expensive distribution licenses making it unsuitable for use at home at larger scale, I drop Virtools as possible candidate. Besides this, the most similar application is Quest3D, and this application seems easier to work with, regarding to the limited testing time I spent on both.

Quest3D and Vizard are priced equally if used commercial, but the difference in educational licenses is large, in favour of Quest3D. However, this latter does not support the Flock of Birds yet, which makes it a little guess.

Personally, at this moment Vizard is with a small lead my favourite, mainly because I could find my way into it very fast. But due to three aspects I'll advice a migration to Quest3D: it appears to be a more complete application, its interface is easier to learn which is important as multiple students will work on VRET, and the costs for educational use are so low it can be bought without having to wait until new subsidiary is received, which will be only next year. Parnassia is helped with this, other master students can really work on VRET, and maybe in the background the fact Act3D is a Dutch company plays a little role...

Last remarks

Maybe you wonder why not all software packages mentioned in the research assignment itself are reviewed. To be more precise: I 'skipped' GreenDino and GameEngine. The first one only develops VR software and not the development environments for it; GameEngine too aims at applications and also offers possibilities for online gaming.
REFERENCES


URLs

Virtual Reality and Phobias (February 2005)

Quick3D Professional (March 2005)

Crossroads3D (March 2005)

Okino Computer Graphics, Inc’s Polytrans (March 2005)

3DLib PHP class (April 2005)

Designer’s Augmented Reality Toolkit (February 2005)

Boston Dynamics DI-Guy (March 2005)

Carbon Graphics’ Geo (April 2005)

VRCO - CAVELib and trackd (March 2005)

Macromedia (March 2005)
Virtual Reality Peripheral Network (February 2005)

EON Reality (March 2005)

MetaVR (March 2005)

MultiGen Creator (March 2005)

OpenGL Performer (March 2005)

[23] www.quest3d.com
Quest3D (March 2005)

Virtools (March 2005)

Virtools Community Swap-Meet Forum (April 2005)

WorldViz’ Vizard (April 2005)

[27] http://www.quantum3d.com
Quantum3D (March 2005)

MÅK (April 2005)

VR Juggler (April 2005)

Dynamic Animations Systems, Inc. (March 2005)

Unreal Technology (April 2005)

The Lua programming language (June 2005)