3D Wizard: A Tool to Easily Build 3D Worlds

Paul Juell and Jasmeet Bhatia
Department of Computer Science
North Dakota State University
Fargo, ND 58105
email: paul.juell@ndsu.nodak.edu
web: http://www.cs.ndsu.nodak.edu/~juell

ABSTRACT

We present a tool to aid in quickly developing 3 dimensional worlds. The collection of objects are described in an English like script file. The 3D Wizard program then converts this into a 3D world in VRML. This VRML then provides a web accessible 3D environment. The system is designed to allow quickly creating lecture material, presentations and various types of diagrams and visualizations. The basic commands are simple and clear enough that faculty and students should be able to quickly create sophisticated worlds with almost no training. The system allows rapid prototyping of a world and the ability to quickly change and try ideas. The intent is to provide tools that makes students created visualizations a reasonable part of an assignment. It is also easy to create presentations using this system. The programmed navigation through the world then shows the steps in the presentation. This produces a truly spectacular effect.

Introduction

We present a tool to aid in quickly developing 3D worlds. A world is specified in a English like, text description. The Wizard program then translates this description into a 3D VRML world. The system is designed to allow quick creation of lecture material, presentations and various type of diagrams and visualizations. The basic commands are simple enough and clear enough that faculty and students should be able to quickly create sophisticated worlds. The description of a world is clear, simple and compact. These properties allow rapid prototyping of a world and minimal overhead in changing a world. Since the file is just text, the creation and changing is performed in any text editor.

This system was developed to address a problem with an effort to provide visualizations for classroom use [1]. We were developing visualizations for various Artificial Intelligence techniques in VRML [2]. We found that we needed to evaluate numerous versions of the visualizations before we addressed the issues we wanted. We would have liked a tool to aid this rapid prototyping process. In addition, our final products were based on the run of a student's program. The run needed to be converted into the visualization. All of our systems were taking a short symbolic description of the problem, and converting these into final visualization. A tool such as 3D Wizard could simply both the rapid prototyping and final delivery of visualization.

The Wizard program translates a very simple and short description of interrelated 3D objects into VRML notation. That is, a convenient notation for specifying 3D objects is converted into a more detailed specification for describing 3D worlds. VRML requires a large number of lines to specify objects and also requires exact locations to be specified. VRML is a powerful way to specify 3D worlds and to distribute them over the Web. The VRML language [3] is a clear way to specify collections of objects in space. However, VRML requires coding a significant amount of redundant information. The 3D Wizard program [4] can automatically provide much of the redundant information and can also automatically provide placement for the objects. This reduces development effort for a 3D world. In addition, the intelligence use of default values and locations reduces the number of decisions for the designer of the world. These defaults come from a model of how the user wants the objects positioned in space. The notation of Wizard is basically an extension into 3D of a 2D language called PIC [Kernighan 81]. The PIC preprocessor is used in conjunction with troff or TeX text processors supplied by UNIX systems. The PIC preprocessor scans the troff file looking for lines specifying PIC processing. These lines are then translated from the picture specification into troff or TeX commands. The troff or TeX processor would then convert that file into a formatted text document with diagrams embedded. An example in PIC are the following commands:

```
box;
arrow;
box;
```

In PIC these draw two boxes with an arrow from the first box to the second. 3D Wizard (or Wizard for short) will take the same commands and produce a 3D VRML world with two cubes with an arrow from the first cube to the second.

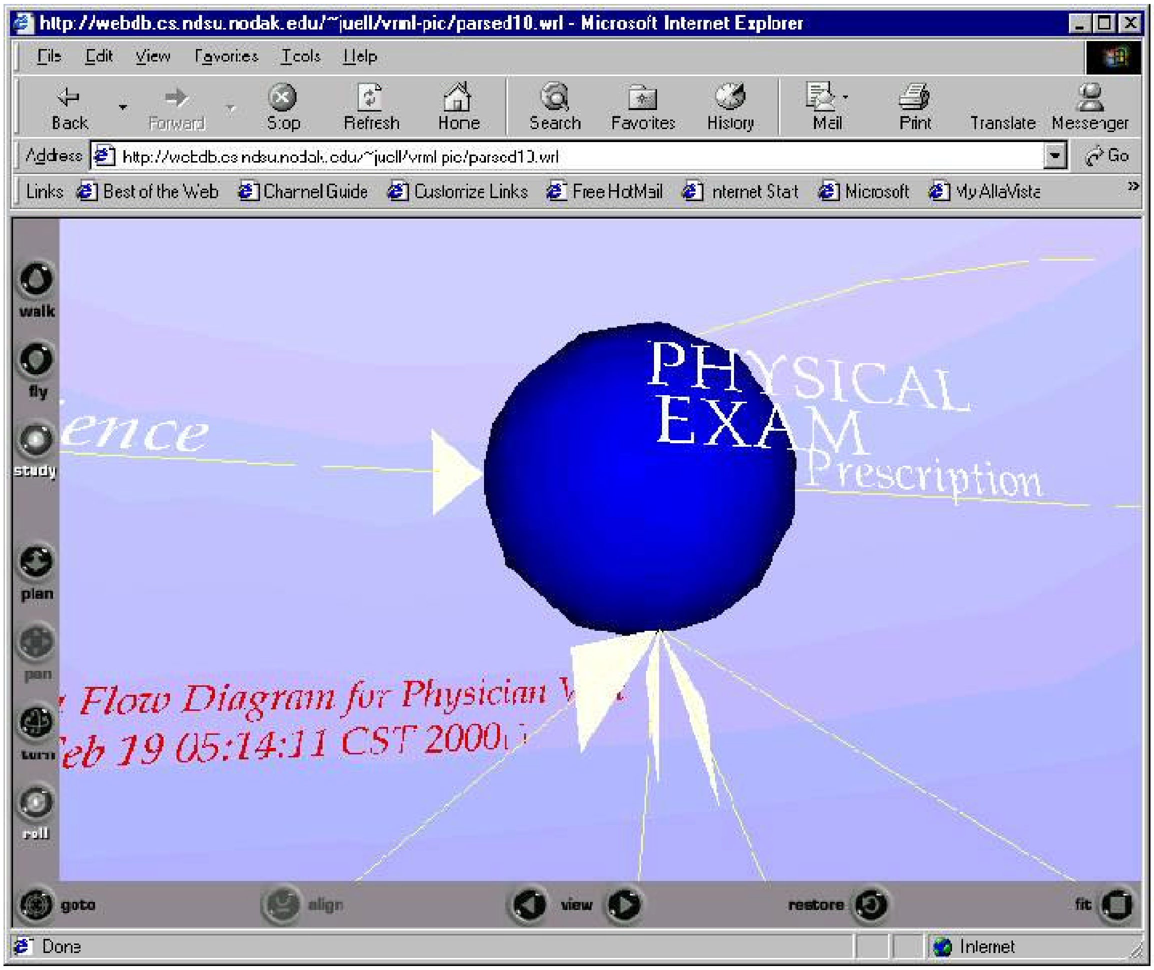
http://webdb.cs.ndsu.nodak.edu/~juell/vrml-pic/parsed10.wrl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Translate Messenger
Address http://webdb.cs.ndsu.nodak.edu/~juell/vrml-pic/parsed10.wrl Go
Links Best of the Web Channel Guide Customize Links Free HotMail Internet Start Microsoft My AltaVista
walk
fly
study
plan
pan
turn
roll
PHYSICAL
EXAM
Prescription
ence
Flow Diagram for Physician V
eb 19 05:14:11 CST 2000
goto
align
view
restore
fit
Done
Internet

Figure 1 Example of Arrow Heads and Text

The Language

The Wizard language is a simple description of the world to be created. The PIC language did may of the things we wanted, so we followed their model for much of the language. However, we needed to add to the language to address items not considered by the designers of PIC. The 2 dimensional specifications needed to be extended out to 3 dimensions. This also gave more possible default directions. The default direction in PIC is left to right. This is reasonable for 2 dimensional diagrams. In 3 dimensional space you may want the default directions, left to right, up to down and front to back. Also, PIC could refer to edges of box as top, bottom and other descriptive terms. These needed to be augmented to address the larger number of faces and points of interest, including front and back. PIC is designed to produce black and white print images. We needed to add colors and transparency. There also are additional feature in VRML, applying textures (images) and links (URL's) to other worlds or web pages. The Wizard notations allow specifying: objects, colors, textures, anchors, text, and positioning information. It also has a simple macro processor to simplify build complex groupings.

If a location is not specified, the next location is chosen a default distance in the default direction. The direction can be changed at any point in the input file. You can also position objects anywhere in the world. Subsequent objects will follow the object, in the default direction. There are variables to allow the user to specify the default values. For example you could change the default radius of a cone or the default length of an arrow. All of the default values can also be set on the command line for the individual object.

Making Diagrams in VRML

We want to produce explanations, diagrams, visualization and presentations. The resultant world/image should be clean and clear. It is particularly important that diagrams not be hidden by artifacts in the worlds. For this reason we needed to modify the normal colored background for VRML. The normal background is either all black or has a horizontal line in the center. This line made it hard to read text and drew the eye away from the diagram. We built a continuous gradated color background, without a horizon line. This highlights the 3D effect, but does not detract from the diagrams. Our default background is a graduated blue, ranging from dark blue at the bottom to a light blue at the top. For long arrows, we automatically scale the size of the arrow head based on line length. We chose a flat arrow head because it highlights the 3D character of the diagram. The combination of the flat arrow heads and the head scaled to line size produces a pleasant feel to diagrams. If an object is labeled, the text is displayed centered and just slightly in front of the object. The process of navigating the world will highlight the independent movement of the text and the object. This produces a striking effect.

Charts and Diagrams

Wizard makes it is easy to specify diagrams such as: system charts, data flow diagrams (DFD), flow charts and various process diagrams. You might use Wizard to build these images just because the 3D worlds are attractive and seem to attract the viewer more than the corresponding 2D images. However, the 3D worlds also give more opportunities in displaying complex information. You can provide information in layers. The bottom layer the diagram the next layer with markers for a particular process, the next layer with alternative labeling, the next layer with an alternate interconnection scheme. Complex constructs such as this could allow examining two options while being able to see both options together. We think people will find diagrams presented in 3D useful for showing complex options and interactions.

Use of Worlds for Presentations

This system produces visually stunning presentations. This is particularly true with the Cosmos plug-in/player which smoothly flies from viewpoint to viewpoint. Each object, other than arrows, are entered into a viewpoint list. The world can then be used as a presentation by stepping trough the viewpoints. Each viewpoint can be treated as a page in a presentation. This is particularly useful for a presentation about a step by step process or where a diagram is the main driving force of the presentation.

The message of the presentation can be supplied in a range of ways. The object can have a multi-line label. This label works well for a few lines of short text messages. The object can have an image added as a "texture". This is an easy way to produce complex looking objects from simple items such as boxes and cylinders. This image or texture then is used to help present the content. If more detail is required, the object can have a "hot link" or anchor. That is the object can click on, taking you to another URL. The URL can be a web page (http) or another world (wrl). The object in the world then presents the concept, the attached web page presents details.

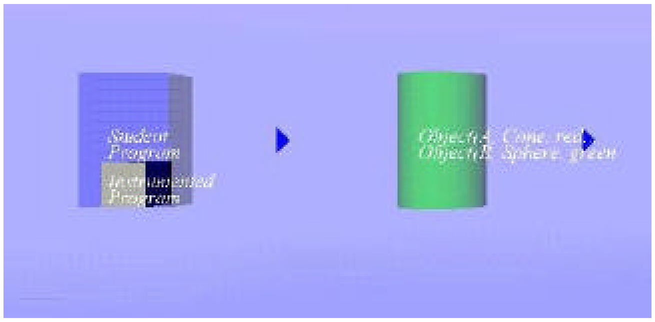

Figure 2 Part of a presentation built using 3D Wizard

Advanced Programming Features

The notation provides a set of defaults to make world building easy. For individual items, the defaults can be overridden by specifying parameters and values on the command line. The default values can be changed by setting the values of special variables on the command lines. This changes the default behavior of the system. The user can also specify his/her own variables, and set the values in command lines. The values can be specified by numbers, arithmetic with other variable values and from the default values. There is also a macro facility. The use of macros, variables and arithmetic calculations can make it simple to build substructures in the world. For example, clusters of groups and complex structures such as tree diagrams can be built.

Conclusions

We have presented a simple way to create 3D worlds. This is a powerful notation that allows compact and quick specification of worlds that are then presented in VRML. The VRML browser/plug-ins allows flying through the world under user control. This notation is convent for creating diagrams, presentations and prototype of systems. If you are interested in this system you can find the code and examples at: http://www.ndsu.nodak.edu/instruct/juell/vp/wizard

Bibliography

[1] Juell, Paul, Addressing Education with Rich Symbolic Visualizations, Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, SSGRR2001, L'Aquila, Italy, August 06 – August 12, 2001.

[2] Hoque, Mohammad M. & Juell, Paul Interactive Visualization of Genetic Algorithm, WebNet World Conference on the WWW and Internet, WebNet 2001, Orlando, Florida, USA, October 23-27, 2001 Association for the Advancement of Computing in Education, 2001.

[3] Hartman, Jed and Josie Wernecke, The VRML 2.0 Handbook: Building Moving Worlds on the Web. Reading, MA: Addison-Wesley Publishing Co. Silicon Graphics, Inc., 1996.

[4] Bhatia, J, 3D Wizard: A Language for Typesetting VRML Worlds, North Dakota State University, Computer Science Department thesis, 2000.

[5] Kernighan, Brian W., PIC - A Language for Typesetting Graphics, SIGOA Newsletter, Vol. 2, No. 1&2, Spring/Summer 1981 and Proceeding of the ACM SIGPLAN SIGOA Symposium on text manipulation, Portland, Oregon, June 8-10, p92-98, 1981.

Acknowledgments

These development efforts are partially funded by the National Science Foundation under grant DUE-9752548.