Virtual Reality as a Gateway for Cultural Immersion

Erhardt Graeff

Arctic Region Supercomputing Center
University of Alaska Fairbanks

Abstract

This paper examines the process of creating a 3-d interactive world for educational purposes. The case study is modeling a Japanese home which students can interact with in order to learn the Japanese language and culture. 3-d model building and design are discussed in terms of the software package Maya. Models are exported using Maya's own exporting tools for VRML, as well as the conversion plug-in PolyTrans for OpenGL C code. The VRML files are associated with a configuration file and then can be viewed in 3-d using the VRScape virtual reality software engine. The C files, after code editing, are displayed via OpenGL through the VRJuggler virtual reality framework. Final use is intended for immersive environment use by a small class of students, specifically within a CAVE-based system such as the Discovery Lab at the University of Alaska Fairbanks. Methodology and troubleshooting involved with the overall project is discussed extensively throughout the paper.

1 Introduction

One advantage that virtual reality has over actual reality is the user's ability to participate in an activity, or interact with an environment, that would be impossible under normal circumstances. Capitalizing on this particular strength, there are many educational applications of virtual reality technology. One such implementation is creating a 3-d interactive world that espouses culture-specific elements to allow students to learn a foreign language and/or become accustom to the functional use of that language within an appropriate setting. Such a world could be conveniently, albeit anachronistically, viewable within an immersive environment like the Discovery Lab located at the University of Alaska Fairbanks (UAF).

The foundation of the interactive world and the project herein examined is that of a normal, present-day Japanese home. The idea was first put forth by UAF Professor of Japanese Yoko Collier upon learning about the Discovery Lab's capabilities. She discussed her ideas with friend and academic colleague Miho Aoki, Professor of Arts at UAF. They wanted to create an effective and fun learning tool for students enrolled in Japanese language and culture classes at the university. Being affiliated with the Arctic Region Supercomputing Center (ARSC), Professor Aoki offered the project up as a Research Experience for Undergraduate (REU) opportunity in coordination with ARSC's summer internship program. Thus, the building began over the course of 10 weeks during the summer of 2004—the chronological scope of this paper.

The goals of the summer were to become competent in using the 3-d modeling package Maya, use the software to design and build a model of a quintessential Japanese home, and finally export the model in a format capable of running in the Discovery Lab's immersive environment.

2 Methodology

Starting with how to manipulate a polygon and ending with how to apply texture materials to faces using OpenGL C++ syntax, the aforementioned goals were accomplished through the course of the summer by virtue of patience and pragmatism.

Learning Maya 5

In order to learn the intricacies of the Maya 3-d modeling software produced by Alias|Wavefront, the *Learning Maya 5|Foundations* tutorial workbook and the *Maya 5 Beginner's Guide|Modeling* DVD were used as instructional aids.

Necessary skills learned for the project include: polygon face extrusion and appending, vertices reduction, texture mapping, and mathematical/visual alignment of objects along multiple axis. The various skills were honed through interaction with the Maya software package guided by the workbook and DVD tutorials. Proficient ability to fulfill the subsequent project goals was measured by the completion of tutorial exercises building example models using the specified techniques.

Model Construction

The Japanese home began as a blueprint furnished by Professor Aoki during the second week of the project's life.

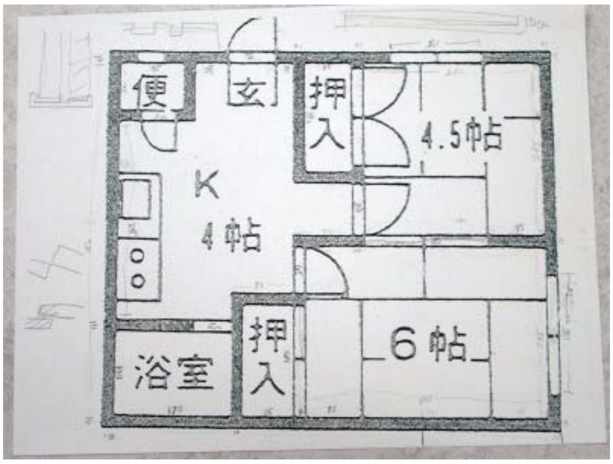

Figure 1: This top view perspective blueprint was used to map out the rooms of the Japanese home.

Dimensions from added to the blueprint in Figure 1 were based on realistic measurements in centimeters assigned to particular door/closet spaces. This provided a foundation for mathematically modeling the layout of all the walls.

Building the model in Maya commenced after mentally assigning the grid divisions of the object-editing window to represent 10 centimeters squared. Thus, beginning with a single polygon cube, faces were extracted to create a scale 3-dimensional skeleton of the house with each wall being 10 cm thick.

The walls were given a simple lambert material with an eggshell color. Tatami mats were the next objects built, being placed on the floor level in the areas labeled on Figure 1 as 6 and 4.5 (the number of mats per room). Professor Aoki created a texture resembling the straw surface of a traditional tatami mat, which was then mapped to the newly created rectangular prismatic polygons.

Over most of the rest of the project's duration, windows, doors, and furniture were added to the model. These were either simply constructed from polygons and then textured, or were imported from Maya binary source files created by Professor Aoki. These source files contained furniture already translated in accordance with the home's dimensional layout. Some of these objects were created using Nonuniform Rational B-Splines (NURBs) and then converted to polygons for uniformity across the Maya model. The objects created using the alternate NURB approach needed to have some or all of their faces' normals reversed after the conversion process in order to preserve the outward mapping of their textures.

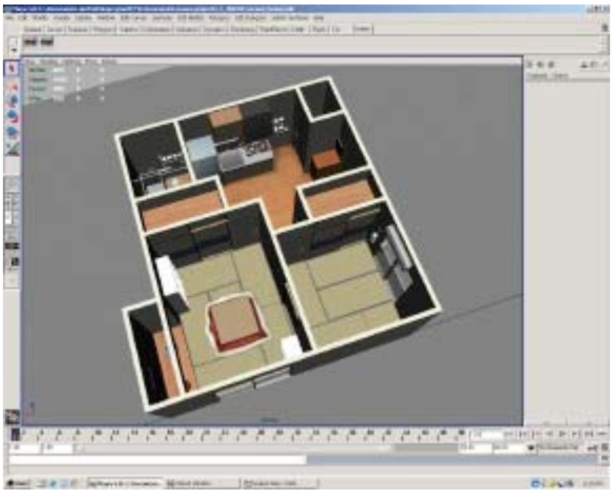

Figure 2: Finished model of Japaneese Home displayed within Maya's perspective editing view.

Exporting the Model

With a majority of the model complete, it was possible to being testing it in a virtual reality environment. To do so meant exporting the objects from Maya into usable formats.

Initial testing began using the VRScape virtual reality engine, which is capable of reading OpenInventor (.iv) and VRML (.wrl) file formats. Both of these formats are natively supported by the export plug-ins included with Maya 5. Thus, the exporting process from Maya was a simple "Export All..." or "Export Selection..." command, and then choosing one of the aforementioned formats—making sure, first, that the plug-in preferences were set so as to enable those file types for export.

To display the OpenInventor/VRML files with VRScape, a configuration file (.vrd) needed to be written and saved in the folder with the rest of the project's object and texture files exported as data from Maya. The configuration file uses a bracketed language in the same strain as the source files (VRML being based on the OpenInventor modeling language). The majority of the programming required to do this is in declaring the objects as Model nodes of the root display with a path to the file and a name declared specific to that object. The object must be then set as either "pickable" (able to be manipulated by the user) or not.

```
Mopel ./modelName.iv THIS_MODEL
Pickable false
```

The other virtual reality engine used in testing was VRJuggler, which reads OpenGL C++ files. Exporting the model for use on this platform is considerably more time consuming and complicated.

First, all polygons in Maya are triangulated in order to better preserve their shape and texture mapping once exported. Then, individual objects in the model that use the same texture or material color are selected and exported into the Maya Object (.obj) format. Each of these Object files must then be separately opened and converted to OpenGL C code files using the PolyTrans 3-d format conversion program.

Each C file needs to be edited by replacing the include call to the automatically generated, and functionally unnecessary, header file with an include-by-decision statement referencing the OpenGL library headers required for drawing the objects.

```
#ifndef OS_X
#include <GL/glut.h>
#else
#include <GLUT/glut.h>
#endif
```

The vertex drawing function in each C file, which by default is named "mesh," needs to be changed to a unique function name which can be called independently by the scene drawing class. The schema used for this project entailed naming the function after the texture commonly shared by the items grouped in that file. The C files are then saved with a CPP extension for C++, which is more readily accepted by the VRJuggler application.

The textures associated with the models then need to be converted to the SGI RGB format which can be read by VRJuggler. This task is accomplished using Apple Shake due to its ability to natively read and write the RGB image format.

To send the graphical data to the VR engine, an application called panorama written by ARSC staff programmers (Don Bahls, et al) was adapted. All code adjustments deal specifically with preparing the models for integration with this software—specifically, the drawscene class which initializes all textures and maps them to polygonal objects available by external function call in their respective C++ files.

The drawscene.cpp class file has to be updated to include internal functions in order to set textures or colors and then draw each individual model to the screen using an external function call. The edit of the drawscene.cpp file for this specific project references 59 textures and has external calls to the 100 model files currently used in the Japanese house.

Once all of the functions and declarations were coded and compiled, testing using a simulator version of the VRJuggler engine was possible. However, testing of this specific model had to be done on a computer with enough graphical texture memory to support all the RGB files. With the number of textures used throughout the Japanese home, testing was only possibly on the SGI Onyx2 (video2) and SGI Onyx5 (igie) systems available at ARSC.

3 Results

The result of the modeling and exporting methodology is the completed education model shown in Figure 2 viewed as a virtual 3-d world in the interactive, immersive environment.

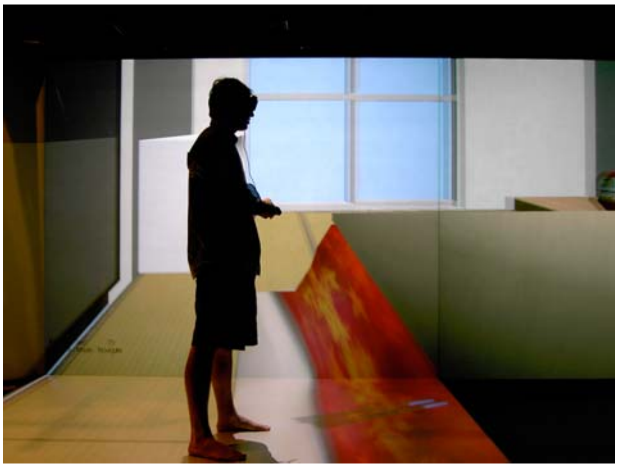

Figure 3: The author is shown in the Discovery Lab at UAF navigating through the model of the Japanese home rendered using the VRJuggler engine.

4 Discussion

The significance of this modeling project is an exercise in teaching as well as virtual reality technologies. These dually important realms each posed challenges that had to be considered in exploring the methodological approach to this project. The key to project's success was technologically realizing the envisioned use of the model.

Envisioned Use of the Model

Professor Collier's hope is that students will get a very real sense of how the Japanese language is used in a Japanese culture-specific setting. When she takes a class into the Discovery Lab, they will have to take their shoes off to enter the house. The taller students may have to duck to get through the virtual doorway, and the class as a whole may have to huddle together due to the cramped space within the home. She wants them to understand the size of a typical home and the common furniture elements in them ranging from the complicated toilet fixture to the simple elegance of a traditional tokonoma complete with tatami mats.

To help the students learn the language, Japanese characters cover the controls on the kitchen implements and bathroom fixtures. In addition, cubes like 3-d flashcards will be present in many of the rooms. These cubes feature a single Japanese Kanji (or Kana phonetic character) character, as well as a pictogram suggesting the derived meaning of the character and at least two readings containing the character for practice. Students will be able to "grab" the cubes using an interactive wand device. They will be able to see the information on the different faces of the cube, and then arrange multiple cubes to spell and identify objects in the home.

Making the Envisioned Model a Reality

As mentioned at the beginning of the Methodology section, the goals of this project were accomplished through patience and pragmatism. Many problems had to be overcome to realize a working model. Similarly, there are many decisions left to be made influencing the future of the project.

Exporting the model into the OpenGL/C++ format for use with the VRJuggler platform caused the majority of the problems encountered during the summer. This stands opposed to the VRScape platform, which allowed all basic functionality to work in a simple export procedure—a final version having been produced by Professor Aoki at the end of the summer.

Most of these said problems with VRJuggler were related to mapping textures and applying colors to the object surfaces using the panorama application. It was initially designed only to set a texture, then draw an object using an external functional call, and then disable that texture to prepare for the next object. However, many of the textures did not map to the faces in VRJuggler in the same fashion as they did in the original Maya design. This problem occurred with objects using the wood texture, as well as the scroll hanging in the tokonoma alcove, and the shoes and slippers at the entrance to the home. Each of these situations was due to a different variable spawning from the design process which was unable to be properly translated during conversion using PolyTrans.

The problem with the wood textures was that when mapped in VRJuggler, they faced in all one direction, meaning that it looked fine to the objects oriented lengthwise in that direction, but appeared stretched for objects oriented perpendicularly. To fix this, new planar mappings were assigned to the objects which created a simpler set of texturey normals to be processed by the conversion software. Many of the shelves had been originally mapped in 3-dimensions, which looked more realistic but caused the aforementioned problem.

The issue with the decorative scroll was far simpler. During testing, the scroll appeared shadowed to the point of visual distortion. This was caused by the default directional lighting in the VRJuggler display settings. By using a GLDisable function call to disable the directional lighting when drawing the simple planar scroll to the screen, the texture was made visible by way of the resulting ambient lighting scheme.

As for the shoes and slippers, they appeared to have a shadowing problem similar to the scroll. However, the ambient fix did not work—leaving nothing but grey soles visible in the tested model. Professor Aoki suggested that the duplicate-and-flip design process was the root of the problem. The slippers and shoes were then rebuilt using the geometry mirroring tool for polygons. This solved the problem such that both sets of footwear displayed properly under normal lighting conditions.

Display problems were not limited solely textures but colors in the model as well. To reduce the overall size of the project in megabytes, many surfaces were given simple colors in RGB format instead of applied texture files. However, the panorama application was originally setup just for applying textures to displayed objects, leaving 40% of the exporting surfaces with only a default pale cream color.

To solve this problem, the OpenGL commands for applying a colored material had to be borrowed from another code source. Logically following the concept that if the VRJuggler model had a default sky blue color applied to the background, it made sense that there was code written to accomplish such a look. Found in the panoApp.cpp class file, a set of gl3Materialfv functions and glFloat variables to hold color values were used to apply colors to surfaces. This code was copied to the drawscene.cpp class and pasted into a callable function for setting material values in the same way texture files could be set when drawing objects.

This worked for all colored objects except for the windows, which featured two transparent faces of varying opacity.Although, the glMaterialfv functions had a fourth (alpha) parameter for transparency, its tolerance when applied created complete transparency for values between 0.0 and 5.0, and complete opacity for values between 5.0 and 10.0. This is problem that will be overcome by using a glBlend function call in later iterations of the VRJuggler model.

VRScape versus VRJuggler

The scope of this paper and "current" state of the project realizes an interactive version of the model working in VRScape and a viewable version working in VRJuggler, each model with its individual strengths and weaknesses.

The VRScape model is easily exportable and already supports elements of automatic transparency, and interactive positioning of objects is available via simple configuration. However, the colors are washed out when displayed and any complicated interactive properties such as translating doors and windows along axis using a wand button cannot be programmed into the rigid system.

The VRJuggler model is very cumbersome to export and even simple changes require recompiling the whole application. Other issues are in scaling and adjusting floor height of the model when displayed. This is accomplished by adjusting undocumented scale constants used in the panoApp.cpp and panoApp.h files. Interactive elements and transparency are not preprogrammed into the modeling language, either. However, colors resemble the original design and any animation or complicated interactivity can be added with the proper C++/OpenGL syntax, allowing for unlimited potential.

Professor Aoki sees this potential in the VRJuggler model and hopes to eventually get all facets programmed and working in a final version. The VRScape model stands as an intermediate and temporary version ready for more immediate use.

Continuing development of a VRML version of the model has potential as well though. This is due primarily to the portability of VRML code. With Internet browsers such as Internet Explorer now capable of natively displaying VRML, students, faculty, and researchers alike can have convenient interactive access to the latest model right on their personal computers--another educational possibility for the model.

5 Conclusion

As stated in the introduction, the scope of this paper covers merely the foundation of an expansive virtual world to eventually be built to help teach students of Japanese language and culture at UAF. Many elements will be added in the future to expand the potential for learning through interactivity as well as navigable magnitude.

Interactive elements still needing to be programmed include doors and windows able to be opened using a button on the immersive environment's navigation wand. This is probably most easily attained by having exported object files for both the doors in an open and a closed position, and then using the button sensor to NOT a Boolean constant which gets tested when redrawing the model in exitFrame. Another element is the manipulation of certain items such as the Kanji/Kana cubes and various pieces of furniture, all needing to be programmed in OpenGL and C++ for the VRJuggler version of the model.

The model will eventually be expanded to include other Japanese locations such as a subway station and a recycling center to create a larger virtual Japanese world, offering a more diverse set of culture experiences for students to immerse in. This is the ultimate goal of the educational implementation of the virtual reality technologies.

The immediate goal is to have a functional model using either VRScape or VRJuggler ready for academic use by

Professor Collier during the Spring 2005 semester at UAF.

In many respects, this project was not itself a complete exercise in research but merely the building of an experiment in educational strategies for teaching. The Japanese home produced is the independent variable to test a hypothesis concerning whether students can better comprehend a foreign language and culture when they are forced to garner knowledge of it through virtual reality immersion in a simulated cultural context.

References

Alias|Wavefront. *The Art of Maya.* Alias|Wavefront, 2003.

Alias|Wavefront. *Learning Maya 5|Foundation.* Toronto: Alias|Wavefront, 2003.

Demers, Owen. *Digital Texturing and Painting*, Pearson Education, 2001.

Maya 5 Beginner's Guide|Modeling. Alias. DVD. Alias|Wavefront, 2003.

Nadeau, David R., John L. Moreland, Michael M. Heck. "Introduction to VRML 97." *SIGGRAPH 98.* 1997. 14 June 2004 <http://www.siggraph.org/education/material/siggraph_courses/S98/18/vrml97/vrml97.htm>.

Sumai no tebiki 2003. 20 May 2002: 10+.