

A NEW GRAPHICAL USER INTERFACE FOR A
3D TOPOLOGICAL MESH MODELER

A Thesis

by

DAVID VICTOR MORRIS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2008

Major Subject: Visualization Sciences

A NEW GRAPHICAL USER INTERFACE FOR A
3D TOPOLOGICAL MESH MODELER

A Thesis

by

DAVID VICTOR MORRIS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Ergun Akleman
Committee Members,	Vinod Srinivasan
	Jianer Chen
Head of Department,	Mark Clayton

May 2008

Major Subject: Visualization Sciences

ABSTRACT

A New Graphical User Interface for a 3D Topological Mesh Modeler. (May 2008)

David Victor Morris, B.E.D, Texas A&M University

Chair of Advisory Committee: Dr. Ergun Akleman

In this thesis, I present a new platform-independent, open source, intuitive graphical user interface for TopMod, an application designed for interacting with 3-dimensional manifold meshes represented by a Doubly Linked Face List (DLFL). This new interface, created using the Trolltech Qt user interface library, enables users to construct and interact with complex manifold meshes much faster and more easily than was previously possible. I also present a method for the rapid creation of a successful online community of users and developers, by integrating a variety of open source web-based software packages. The new website, which includes a discussion forum, a news blog, a collaborative user and developer wiki, and a source code repository and release manager, received an average of 250 unique visits per day during the first two months of its existence, and it continues to be utilized by a variety of users and developers worldwide.

ACKNOWLEDGMENTS

I would like to thank my committee chair, Dr. Ergun Akleman, committee members Dr. Vinod Srinivasan and Dr. Jianer Chen, for all their help and encouragement throughout this project. A special thanks to Dr. Ergun Akleman and Dr. Jianer Chen for formulating the theory behind the double linked face list (DLFL) data structure, Dr. Vinod Srinivasan for creating the original implementation of TopMod, and all other TopMod developers past, present and future, including Esan Mandal, Eric Landreneau, Zeki Melek, Stuart Tett, Michael Stanley, Ozgur Gonen, Paul Edmundson, Fusun Eryoldas, Cansin Evrenosoglu, Xu Bei, and Brian Barran.

I would also like to thank my family for supporting me in all my academic and professional endeavors. Without their continuous encouragement and support I would not have been able to make it this far.

Finally, I would like to thank all the faculty, staff and students of the Visualization Laboratory who have helped create such an intense, creative, and fun learning environment. Another special thanks to Stuart Tett, without whom I would not have learned how to debug my code all by myself. He has been an integral part throughout the development of this thesis.

TABLE OF CONTENTS

CHAPTER		Page
I	INTRODUCTION AND MOTIVATION	1
II	BACKGROUND AND PREVIOUS WORK	7
	A. Topological Mesh Modeling	7
	B. Graphical User Interfaces	8
	C. Existing Interfaces	9
	1. Maya	9
	2. Blender	10
	3. Rhino	11
	4. Wings 3D	11
	5. Other Applications	13
	D. Pop-up Command Line Interfaces	13
	E. GUI Toolkits	14
	1. FLTK	14
	2. wxWidgets	15
	3. GTK	15
	4. Qt	16
III	METHODOLOGY AND IMPLEMENTATION	17
	A. Removing FLTK Dependency	17
	B. Qt Interface Design	18
	1. Customizability	19
	2. Popup Command Line Interface	23
	3. Context-Sensitive Right Click Menus	24
	4. Heads Up Display	24
	C. Selection Routines	26
	1. Edges	27
	2. Faces	28
	3. Vertices	30
	4. Growing and Shrinking Selection Sets	32
	D. Thesis Research Consolidation	34
	1. Planar Modeling	35
	2. Advanced Extrusions	35

CHAPTER	Page
3. Bezier Patch Rendering	36
4. Rectangular Wireframe Modeling	38
5. New Approach to Interactive Rind Modeling	39
6. New Approach to Extrusions	39
7. New Approach to Other Basic Operators	40
8. Scripting Interface	41
E. Portability and Platform Independence	42
F. Internationalization	43
G. Online Community	44
1. Blog	47
2. Discussion Forum	49
3. Image Gallery	50
4. User Manual	52
5. Source Code Documentation	54
6. Subversion Repository	56
7. Web Site Usage Statistics	58
IV RESULTS	60
V CONCLUSION AND FUTURE WORK	68
REFERENCES	71
VITA	76

LIST OF FIGURES

FIGURE	Page
1	Screenshot of the previous version of TopMod, which uses the FLTK user interface library. 8
2	Screenshot of the interface from Maya [®] 7 on OS/X. 10
3	Screenshot of Blender's interface. 11
4	Screenshot of Rhinoceros3D's interface. 12
5	Screenshot of the Wings3D interface. 12
6	Screenshot of the user interfaces of Silo, SLIDE, Google SketchUp, Cheetah 3D, k3dSurf, and Sculpture Generator. 13
7	Screenshots from Quicksilver pop-up CLI for OS/X and Launchy, a similar interface for Windows. 14
8	Screenshot of the combo box interface in the previous version of TopMod. 18
9	Screenshot of the color customization dialog in the TopMod preferences window. 21
10	Screenshot of different viewport styles that can be created using customization options in the TopMod preferences dialog. 21
11	Screenshot of the stylesheet customization dialog in the TopMod preferences window. 22
12	Screenshot of the shortcut customization dialog in the TopMod preferences window. 23
13	Screenshot of the TopMod pop-up command line interface. 24
14	Screenshot of the face selection mode right click menu. 25

FIGURE	Page
15	Screenshot of the TopMod heads up display, rendered in the OpenGL viewport. 26
16	Examples of edge loop selection, edge ring selection, and crossing window selection. 29
17	Examples of face loop selection, select similar faces, select faces by surface area, and checkerboard face selection, and crossing window selection. 31
18	Examples of converting vertex and face selections to edges. 32
19	Screenshots illustrating the selection expansion (growing) and contraction (shrinking) process for faces. 33
20	Screenshots illustrating the selection expansion (growing) and contraction (shrinking) process for vertices. 33
21	Screenshots illustrating the selection expansion (growing) and contraction (shrinking) process for edges. 34
22	Screenshots demonstrating the process of cutting multiple vertices and edges. 36
23	Two of the advanced extrusion operators from Eric Landreneau's thesis research, dodecahedral and octahedral, used in conjunction with various remeshing modes. 37
24	Screenshots of bezier patch models in the TopMod viewport. 37
25	A series of wireframe models with different cross-sectional dimensions. 38
26	Example of a face selection set that is rind modeled all at once using the new rind modeling feature in TopMod. 40
27	Example of a face selection set that is extruded all at once using the new extrusion feature in TopMod. 41
28	Screenshot from TopMod with the Python scripting interface by Stuart Tett. 42

FIGURE	Page
29	Screenshot from the Qt Linguist application. 44
30	Screenshot of the TopMod interface translated into French. 45
31	Screenshot of the Jotero Image Gallery, by Torolf Sauermann. 46
32	Rendering from the Jotero Image Gallery, by Torolf Sauermann. 46
33	Screenshots from the topmod3d.org blog administration interface, the plugin panel, and the main page. The main page style was adapted from the Big Blue theme created by Bob Jiwakacau. 48
34	Screenshots from the topmod3d.org forum administration panel, the main page, and an example forum post. 50
35	Screenshots from the TopMod Image Gallery main page, sample image gallery page, sample image page, and the administration panel. 51
36	Screenshots from the TopMod Wiki main page, user manual, tutorials index, model index, script index, and a sample operation page. 55
37	Screenshot from the TopMod source code documentation web page. . 56
38	Screenshot from the TopMod Google Code project page. 57
39	Screenshots from the Google Analytics page for topmod3d.org showing geographic usage data for Texas, France, and Germany over a two week period. 59
40	Screenshot of the old (FLTK) TopMod interface. 61
41	Screenshot of the new (Qt) TopMod interface. 62
42	A model created using the new version of TopMod by Torolf Sauermann. ©2007 (http://www.jotero.com). 64
43	A twisting torus model with stellated extrusions on each of the 1280 faces. 64
44	A model created by simultaneously extruding a series of face loops with over 1100 faces. 65

FIGURE	Page
45	A model created by selecting three face loops containing more than 1000 faces and using the new crust modeling tool. 65
46	A model created by Torolf Sauermann. ©2007 http://www.jotero.com . 66
47	A model created using the new version and the Python scripting engine by a user from the TopMod web community. 66
48	A model created using the new version by a user from the TopMod web community. 67
49	Voronoi subdivision wireframe models created with a series of Python scripts and imported into TopMod. 69

CHAPTER I

INTRODUCTION AND MOTIVATION

To interact with any piece of computer software, there must exist some form of user interface or front end to allow users to access important functions or algorithms in order to complete specific tasks. The most common user interface design, still used in the majority of desktop computer applications today involves a series of graphical (W)indows, (I)cons, (M)enus, and (P)ointing devices, collectively referred to as a WIMP interface.

WIMP interfaces are quite powerful for a variety of different software applications, but are often limiting or obtrusive for more complex programs such as 3D modeling packages and interactive games. For example, WIMP interfaces tend to take up a large portion of the screen space that could otherwise be used to display complex data sets. Compromises must be made in terms of how much screen space the interface uses and how much the data uses. One simple solution to the screen space problem is to use a system of graphical icons to represent various commands and operations in an application, and provide a string of text, or tooltip, to further describe the operation as the user hovers the mouse over that particular icon.

However, in many complex computer graphics applications with literally hundreds and often thousands of unique commands such as Autodesk[®] Maya[®], defining a clear and consistent system of graphical icons can be an overwhelming task and can often result in more confusion for the users. This problem can be lessened slightly by grouping similar icons into convenient toolbars that can be placed on or near other windows to provide faster access with the mouse.

This thesis follows the style of *IEEE Transactions on Automatic Control*.

Any remotely complex computer graphics package necessitates a clear hierarchy of menus and submenus, as well. As features are added to a particular application and the menus become buried deeper into sub-levels of the main menus as is the case with Maya and Blender, it is necessary to provide users with context-aware popup menus, typically accessible by right clicking the mouse. These popup menus allow users to access commonly used operations with a single click of the mouse without having to move it to the menu bar usually located at the top of the screen or application window. With a clear and easy to understand hierarchy of top-level and sub-level menu items, as well as an intuitive right click popup menu functionality, complex computer graphics applications become much more efficient and user friendly.

To make a piece of software available to the largest possible base of potential users, it should be designed to run on as many operating systems and platforms as possible. In the case of computer graphics applications, it is often desirable to export data generated by one program for use in another program. Many computer graphics packages are only available for one specific operating system, such as Autodesk 3D Studio Max[®] for Windows, while others such as Blender are only available for Linux platforms, OS/X, and Windows[1].

Many popular 3D modeling applications provide users the ability to customize the user interface in various ways to allow for a greater level of efficiency with particular operations or work flows. For example, the interface for Autodesk Maya is written entirely in the proprietary Maya Embedded Language (MEL), which allows users to move various elements of the interface around the screen and even write their own interface elements with simple MEL script commands. Autodesk 3D Studio Max offers a similar feature with MAXScript. Other software packages such as Rhinoceros[®] and Blender have similar features as well through the VBScript and Python programming languages.

In many cases, the application is designed to remember a particular user's preferences in terms of where various elements of the interface are placed for the next time the program is opened. This allows the user to pick up quickly where he/she left off in their previous work session without having to relocate windows and toolbars in the interface for the particular task at hand.

Keyboard shortcuts allow for quicker access to commands without having to use a mouse to access a particular graphical element on the screen. In complex 3D modeling applications such as Maya and Blender, keyboards shortcuts are essential to reaching a certain level of efficiency that is necessary in all steps of a production or content creation pipeline.

Moreover, as users become more efficient with a particular software or portion of an application, it is often desirable to create custom keyboard shortcuts that enhance the workflow for someone who uses a particular command very often that may not have a defined shortcut or may have a shortcut counter-intuitive shortcut key sequence. Allowing users to customize shortcuts for virtually any command or operation in an application gives an extreme amount of power to the users and it allows them to share their customized work flows or procedures with other users in the community.

Many 3D modeling applications also gives users the ability to customize the look of the user interface and 3D viewport displays in terms of colors and layout. Features of this type of customization include changing the color of 3D objects, the viewport background, lighting, sub-object elements (vertices, edges, faces) and features (normals, face centroids, selection sets).

Other pieces in the interface that affect the visual display such as the camera field of view and the thickness of edges, vertices, and normals are also customizable through a graphical interface. Adjustments to the visual display of 3D objects such

as these are important for users with different types of vision impairments such as low vision, colorblindness, or amblyopia. For example, a user may perceive a 3D model better with a lighter foreground and thick, dark edges and a darker background color, or vice versa, while others may prefer an orange to blue contrast between the object color and the viewport background based on the sensitivity of their eyes [2].

There exist a myriad file formats for storing 3-dimensional mesh data, and it is often desirable to work with the same mesh in multiple software packages. This requires each application to have the ability to import and export data in at least one of many 3D file formats. One of the most commonly used and widely support mesh formats is the .obj format developed by Wavefront Technologies during the 1980s. It continues to be widely supported across a large number of 3D modeling applications, but is limited in its use for certain purposes such as rapid prototyping and interactive display on the web. While there are a plethora of free 3D file format converters available on the web, providing support for other common 3D file formats within a specific application can be beneficial to speed up work flow and compatibility.

For example, one common format with an extremely simple implementation that interfaces with rapid prototyping equipment such as 3D printers and fused deposition modelers (FDMs) is the Stereolithography, or STL, format [3]. Another common format used for displaying 3D meshes in a web browser through a simple Java applet is the Mathematica LiveGraphics3D (*.m) file format [4]. Like OBJ files, STL and LiveGraphics3D files can be stored in a human readable ASCII format.

As computing hardware and software has become cheaper during the last twenty years, many more users around the world are able to afford complex systems that can perform previously unthinkable tasks such as 3D modeling and video editing. As a result, there is a huge desire for popular computer graphics applications to be offered in multiple languages to cater to the needs of other cultures where English is not

the main language or is simply not widely used. Many software companies have responded to this demand by providing their software in other languages such as Google SketchUp[®], which is currently offered in English, French, Japanese, Spanish, German, and Italian, and Blender, which is available in English, Japanese, Dutch, Italian, German, Finnish, Swedish, French, Spanish, Catalan, Czech, Brazilian Portuguese, Simplified Chinese, Russian, Croatian, and Polish.

Virtually all computer graphics applications provide some sort of user documentation that attempts to explain the various functions of the application in an easily understandable format. Maya comes with an entirely separate documentation server application that runs as a background process on a user's machine and provides enhanced searching capabilities for the entire contents of the user documentation. Other packages, such as Blender provide user and developer documentation through a web site, and provide access to various parts of the documentation through links in the program. In an open source collaborative application such as Blender it makes more sense to offer the documentation in a web-based environment where users around the world can work together to develop it and even provide help in multiple languages[5].

One example of a complex 3D modeling application that necessitates a friendly and customizable user interface is TopMod, the Topological Mesh Modeler, which is a topological mesh modeling system that allows users to create high genus 2-manifold (water tight) meshes. Prior to this thesis, the implementation of TopMod had a primitive user interface that was difficult for new users to learn and adopt quickly. It lacked a clear hierarchy of windows, menus, icons, and toolbars, and did not provide any level of customizability.

It also did not allow quick access to the important operations in the program nor does it promote discovery of new or previously unused operations. An early version of TopMod was available for Windows XP, but the newest version only ran on Linux-

based systems. There also existed no comprehensive set of documentation for users to learn how to use the program or for developers to promote future development. As a result many users and potential developers were turned away from the program without ever being introduced to some of its most powerful features and its true potential.

The goals of this thesis are to provide solutions to these problems. TopMod should provide a familiar and intuitive interface that resembles other popular modeling packages and functions in similar ways. TopMod's source code should be open to development outside the walls of the Visualization Laboratory at Texas A&M University, and there should exist an online user community and comprehensive set of documentation for users and developers alike from which to learn how TopMod works on a variety of levels.

CHAPTER II

BACKGROUND AND PREVIOUS WORK

A. Topological Mesh Modeling

Akleman, Chen and Srinivasan introduced a minimal set of operators for the development of robust manifold mesh modeler represented by the DLFL data structure [6], a theoretical framework which guarantees topologically correct 2-manifold, or water tight, meshes. The minimal set of operators are INSERT-EDGE, DELETE-EDGE, CREATE-VERTEX, and REMOVE-VERTEX. Using these operators it is possible to create all and only manifold meshes. The system is used primarily for architectural and sculptural design. Since TopMod was first introduced, a myriad of researchers and students have extended its functionality [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20] (See Fig. 1).

As a result of the work and research of various students in the Visualization Sciences program at Texas A&M, TopMod now provides a wide variety of operations built upon the minimal set of operators. These high level operations include extrusion, remeshing, planar modeling, and high-genus modeling. The High-genus operations result in meshes with a large number of holes or handles. For example, wireframe modeling is the process of converting the edges of a mesh into connected solid bars of a user-specified thickness, creating a lattice or truss-like structure [21]. For even the most basic operations on a mesh, such as inserting an edge between two vertices, the user must interact with a graphical user interface.

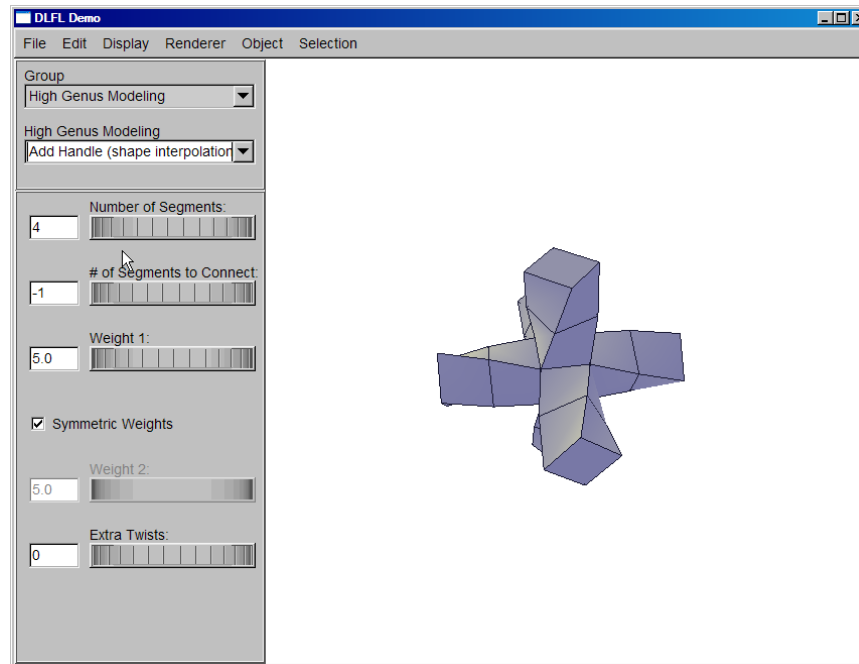


Fig. 1.: Screenshot of the previous version of TopMod, which uses the FLTK user interface library.

B. Graphical User Interfaces

Graphical user interfaces have existed in some form or fashion since the late 1960's, with the unveiling of Douglas Englebart's "oN-Line System" (NLS), which consisted of a keyboard, keypad, mouse, and windows[22, 23]. Since then major advancements in GUI techniques include Doug Englebart's on screen pointer controlled by a pointing device, Xerox-PARC's menu system, dimming of inactive buttons and context-aware cursors by David Tilbrook all introduced around 1975. Xerox-PARC demonstrated the concept of popup menus as early as 1976, and hierarchical menus, bitmapped displays, dialog boxes, and multiple fonts and styles showed up as early at 1978-1979[24]. Other key advancements during the late 1970s and early 1980s include pulldown menus, a main menu bar, disabling of menu items, move, copy and delete commands, tiled and overlapping windows, the first commercial mouse, scroll bars,

push buttons, radio buttons, check boxes, and the ability to cut, copy and paste with a mouse. Silicon Graphics began shipping the first 3D graphics workstations in 1983 and around the same time MIT began developing the X-Window system, still widely used[22, 25].

C. Existing Interfaces

In this section I will discuss briefly some of the many 3D modeling applications available today that offer powerful modeling tools and unique user interfaces.

1. Maya

One of the most popular 3D graphics applications in use today found its beginnings as early as 1983 through developments by Silicon Graphics, Alias, and Wavefront Technologies, who were all developing 3D graphics solutions for various industries such as high-end CAD and product design to television commercial and movies. By 1995 the companies would see their research in 3D computer graphics merge into a single company and a piece of software which eventually became known as Maya[25]. The creators of Maya have continually added new user interface elements to enhance productivity and ease of use of their software during the last twelve years (See Fig. 2).

One key feature which came about in 1999 with the release of Maya 4 that revolutionized the idea of the WIMP interface was the idea of the marking menu or hotbox[26]. Marking menus basically allow the user to access all the commands that are normally present in the top bar of the application window, directly under the mouse cursor simply by holding the spacebar key. A transparent window pops up displaying a grid which allows faster access to hundreds of commonly used Maya

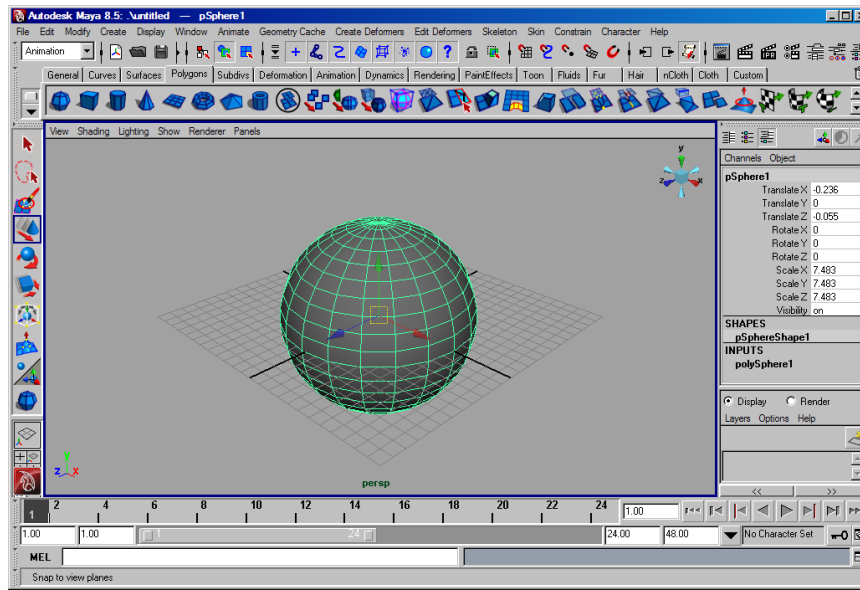


Fig. 2.: Screenshot of the interface from Maya[®] 7 on OS/X.

commands[27]. Maya also overrides the typical right click menu with a gestural marking menu that spreads the options out radially instead of in a linear list. These interface elements have become essential to being efficient when working with Maya, and other software companies have begun to implement similar interface paradigms in order to increase efficiency and ease of use[28, 25].

2. Blender

Blender is a free open source and cross-platform 3D modeling and animation system that provides a rich set of features for advanced modeling, animation, and compositing. The interface is written using its own GUI library, known as GHOST, which renders all interface elements in OpenGL. The GHOST user interface library is deeply embedded in the Blender source code, and is not available as a separate library for use in other OpenGL applications [29] (See Fig. 3).

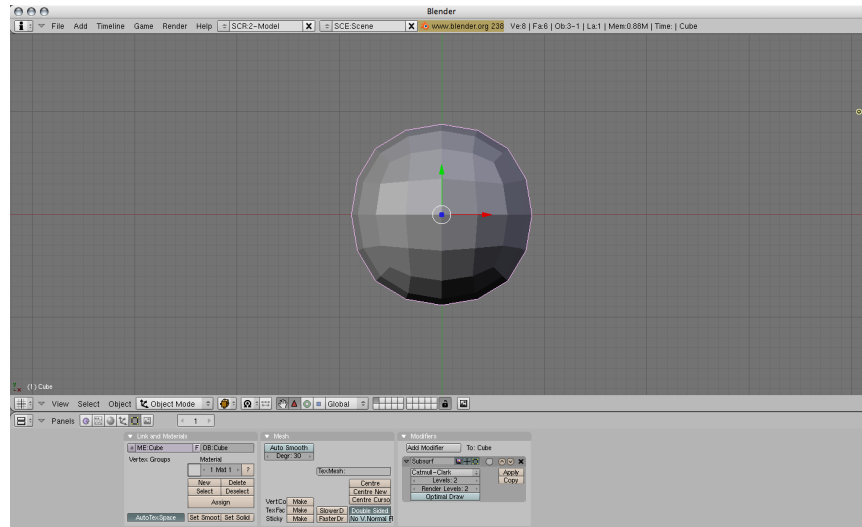


Fig. 3.: Screenshot of Blender’s interface.

3. Rhino

Rhinoceros 3D is a NURBS-based 3D modeling application that is used in a variety of industries such as jewelry design, automotive design, architecture, and multimedia design. It was originally developed as a plug-in for Autodesk AutoCAD[®], and the primary input mechanism for executing commands is through a command line interface, or CLI. The CLI features an auto-completion mechanism which will suggest possible commands based on what the user types as they are typing. This feature is similar to the way AutoCAD works, and allows for faster access to a large number of commands [30, 31] (See Fig. 4).

4. Wings 3D

Wings 3D is a free open source 3D polygonal mesh modeling application available for Windows, Linux, and OS/X. It features an easy to use user interface with a series of context-sensitive right click menus, which allows the interface to be much simpler and easier to learn initially (See Fig. 5).

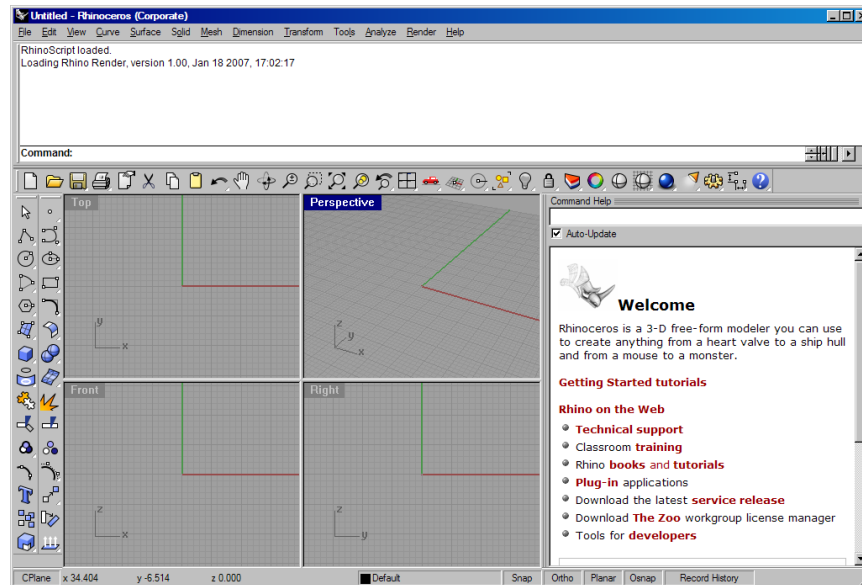


Fig. 4.: Screenshot of Rhinoceros3D's interface.

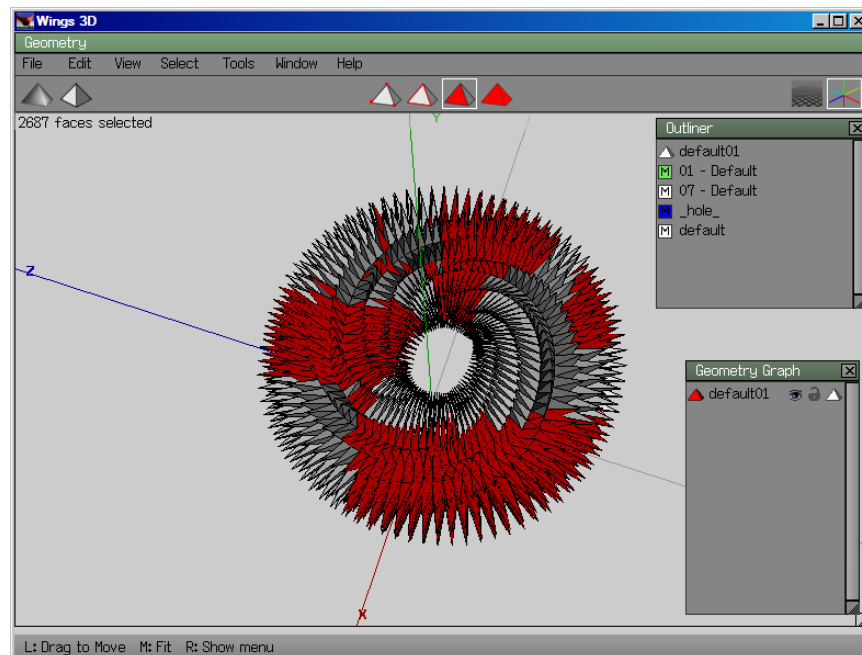


Fig. 5.: Screenshot of the Wings3D interface.

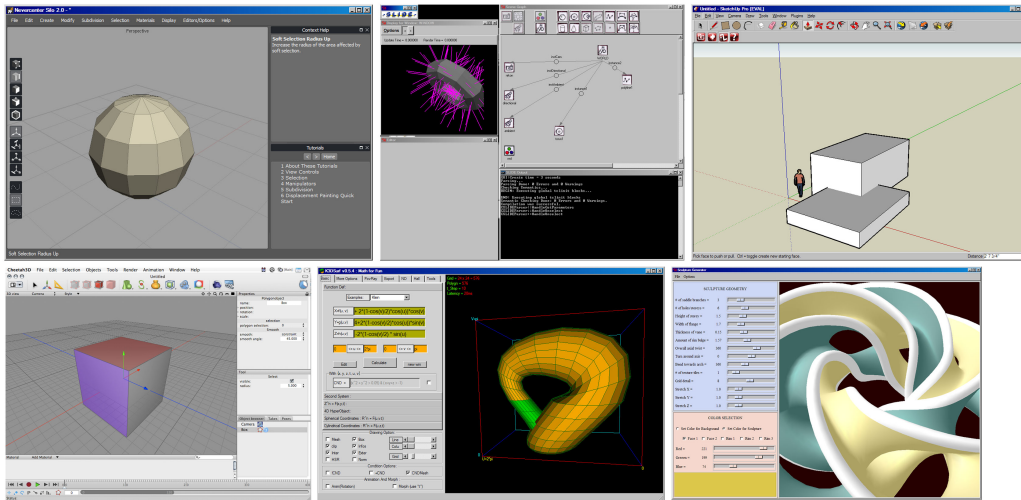


Fig. 6.: Screenshot of the user interfaces of Silo, SLIDE, Google SketchUp, Cheetah 3D, k3dSurf, and Sculpture Generator.

5. Other Applications

There are a wide variety of other 3D modeling applications that provide unique interfaces, such as Silo, SLIDE, Google Sketchup, Cheetah 3D, k3dSurf, and Sculpture Generator (See Fig. 6).

D. Pop-up Command Line Interfaces

Like the command line interface in Rhino 3D described above, other applications feature similar interfaces for keyboard input and allow access to a wide variety of commands. Quicksilver (<http://quicksilver.blacktree.com/>), a free application available for OS/X from Blacktree Software that runs continuously in the background of the operating system, provides a pop-up command line interface that allows users to rapidly execute commands such as opening a URL, launching an application, manipulating local files, and sending emails and instant messages.

The interface can be invoked by hitting a simple keystroke combination such as

CTRL + SPACEBAR, and as the user types, Quicksilver suggests actions or operations based on the types of plug-ins enabled. There is a similar application developed for the Windows operating system, known as Launchy (<http://launchy.sourceforge.net/>), but is used primarily to launch applications and lacks extra features such as email and instant messaging (See Fig. 7).

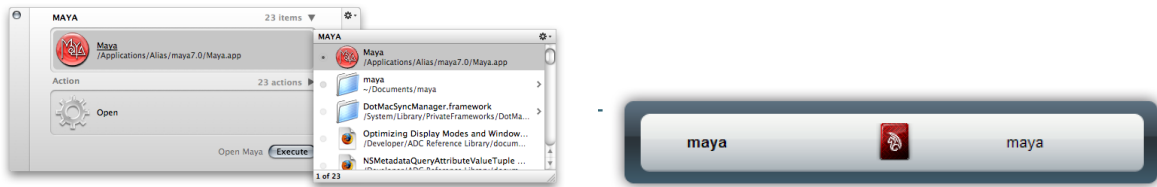


Fig. 7.: Screenshots from Quicksilver pop-up CLI for OS/X and Launchy, a similar interface for Windows.

E. GUI Toolkits

Over the last fifteen to twenty years many individuals and corporations have taken on the daunting task of creating a universal graphical user interface toolkit to allow software developers to create and maintain complex graphical applications with relative ease. Today many open source and commercial GUI toolkits are available for use in a variety of programming languages, such as FLTK, WxWidgets, GTK, originally developed for the GNU Image Manipulation Program, and Qt[®][32, 33, 34].

1. FLTK

FLTK (Fast Lightweight Toolkit) is a free cross-platform GUI library first developed in 1998 by Bill Spitzak and others under the LGPL license for the C++ programming language. It has a significantly lower learning curve for new users and features a

lightweight memory footprint. As a result of the smaller package size when used in executable applications, however, FLTK does not contain as many interface widgets and is not as customizable as some of its competitors. Like other libraries, it now contains bindings for Python and Ruby, other object oriented programming languages [35].

2. wxWidgets

wxWidgets, another cross-platform C++ GUI toolkit, was created in 1992 by Julian Smart, and available free of charge under the permissive, OSI-approved WxWidgets license. Unlike FLTK, wxWidgets is a native mode toolkit, which means it uses graphic primitives from each operating system to draw widgets, but allows developers to create custom widgets as well. It is also available in other programming languages such as Perl, Python, LUA, Ruby, Smalltalk, Java, and JavaScript [34].

3. GTK

GTK (GNU Image Manipulation Program Toolkit, or GIMP Toolkit) was originally created in 1997 by Spencer Kimball, Peter Mattis, and Josh MacDonald as the GUI library for the open source GNU Image Manipulation Program. It is available under the LGPL license like FLTK, and is written primarily for the C programming language. It also has bindings for Python, Java, Perl, Ruby, C++, Pascal, PHP, Haskell, Pike, Tcl, and various C# .NET. GTK is cross platform and open source, but it requires X11 libraries to run on OS/X which does not allow programmers to exploit the unique GUI features of OS/X as is possible with Qt[33].

4. Qt

Development of the Qt user interface library began in 1991 by Haavard Nord and Eirik Chambe-Eng, who would later incorporate their developments into what became known as Quasar Technologies in 1994, and eventually as Trolltech[®]. Key features of the Qt library includes platform independence and a meta object compiler, which allows for the use of programming concepts not present in C++ such as introspection (runtime determination of object types) and a signal/slot system to allow for asynchronous input and output and event driven programming constructs.

By preprocessing the source code and attaching meta information to the classes, programmers have the ability to safely connect multiple signals to multiple slots in different threads, a feature common to many languages other than C++[32, 36]. Other features of Qt that will be exploited in the new interface for TopMod include a robust main window architecture, a multithreaded programming support, Wacom Tablet support, an OpenGL module, an XML Module, and internationalization support[37, 32].

CHAPTER III

METHODOLOGY AND IMPLEMENTATION

My approach to designing and implementing a new user interface for TopMod involved removing TopMod's dependency on FLTK, duplicating the old interface using the equivalent Qt interface widgets, redesigning the entire interface structure to use a common tool options box and other more common WIMP interface elements instead of a series of combo boxes, writing new sub-object selection routines to speed up many of TopMod's unique modeling operations, and consolidating previous thesis research projects into the new version. This process also involved designing all displayable text in the source code to be localizable, making sure the new version was portable and platform-independent, adding a diagnostic heads up display, and creating a robust online community complete with user and developer documentation, a discussion forum, a news and updates blog, and a public image gallery.

A. Removing FLTK Dependency

The first step in redesigning the TopMod interface was to remove from the source code everything that involved FLTK interface elements. The FLTK-based user interface of TopMod provided access to all of the operations through a series of combo boxes in the left pane, the first which allows the user to choose a mode of operation including Basics (INSERT-EDGE, DELETE-EDGE, COLLAPSE-EDGE, SUBDIVIDE-EDGE, COLLAPSE-EDGE and SPLICE-CORNERS), Extrusions (Cubical, Icosahedral, Stellate, etc...), Remeshing (Doo-Sabin, Pentagonalization, etc...), High-genus (Wireframe modeling, Rind modeling, Multi-face handle), and Texturing. After choosing the mode of operation, another combo box would appear below with the list of operations for that mode (See Fig. 8).

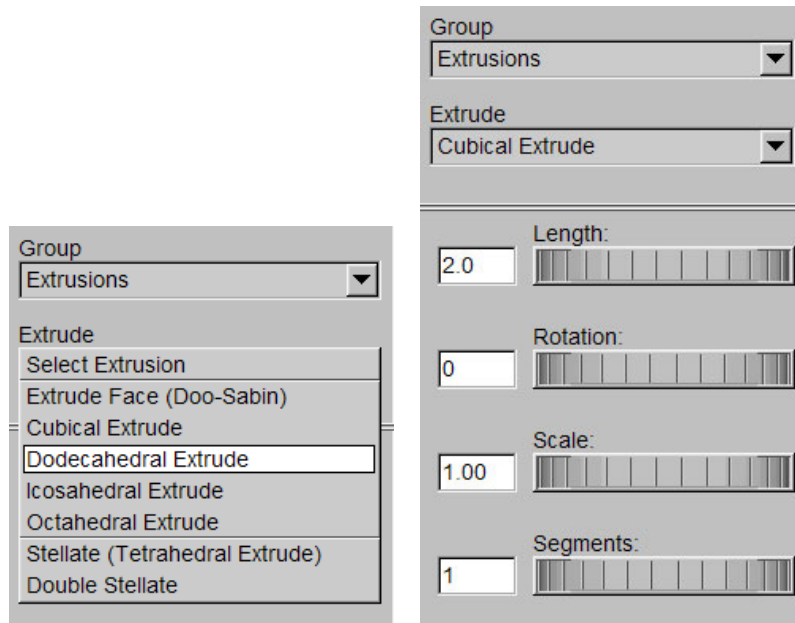


Fig. 8.: Screenshot of the combo box interface in the previous version of TopMod.

This type of user interface is quite tedious and restrictive, and does not allow users to discover and learn how to use different operations quickly or easily. All of the basic operations were buried below at least two combo boxes, and required too much mouse interaction to access them. Also, the custom slider widgets used to set floating point values throughout the TopMod interface did not allow users to manually type in values when precision was desired, and they made it quite difficult to slide to a specific value with a mouse.

B. Qt Interface Design

The new interface I have created for TopMod features an intuitive system of iconography, organized menu items, a fully customizable interface with configurable shortcuts and style sheets that will allow the user to tailor their copy of the application to his or her specific needs. This has been accomplished through the use of various features

of the Qt interface library. Every operation or function in the program is stored in the code as a Qt action, or QAction. The actions are represented graphically in a variety of ways including toolbar buttons and menu items. This action driven design allows other TopMod developers to easily expand the functionality of the interface and expose more of the internal structure of the doubly linked face list to the user.

In addition to the clear hierarchy of icons and menus, TopMod provides more advanced interface elements as users become more comfortable with the program's functionality. One such interface will be a popup command line interface for accessing all the program's operations (from INSERT-EDGE to SAVE FILE AS...) by typing the command. The command line input will automatically complete the commands as the user types and will display a list of possible choices below the cursor based on the current input. This will essentially allow every command to be accessed in less than five keystrokes without touching the mouse.

1. Customizability

Interface customizability is another common feature of 3D modeling applications, and was an important factor in the redesign of TopMod. As users become more comfortable with a particular 3D modeling application, or any application for that matter, they often use certain tools or toolsets more than others, and may desire to reposition buttons or menus accordingly. In the new version of TopMod the menus, submenus and toolbars can be detached from their default location and placed anywhere on the screen to allow for faster access to specific commands and tools.

Another enhancement to the overall usability of TopMod involves saving files. As TopMod contains many experimental features and algorithms that have not been fully tested, it is prone to crashing while a user is in the middle of making a complex model. While it is out of the scope of this thesis to investigate and fix all the internal

errors related to the DLFL data structure that cause TopMod to crash, I have taken these issues into consideration when implementing an auto-save and an incremental save feature. Auto-save allows the user to choose an interval of time (in minutes) after which to automatically save their model to disk overwriting the previously saved version of the file. This allows users to pick up where they left off if TopMod happens to crash during a complex operation without having to explicitly choose “File-¿Save” before each operation is executed.

Incremental save, when toggled, will append a 3-digit string of numbers to the end of the current file and save it under that new name with the same extension. Each time a save operation is executed, whether manually or automatically through the auto-save feature, TopMod will increment the 3-digit number up to the limit set by the user in the preferences window. The default limit is 20, so TopMod will save up to 20 versions of the model. If save is executed more than 20 times, TopMod will wrap the suffix number back around from 020 to 000 and begin to overwrite the existing files in that order. This is a feature common to many 3D modeling applications, as it is very difficult to manage revisions within a single file.

Other interface customizability enhancements include viewport coloring and general display options such as wireframe color and thickness, silhouette color and thickness, lighting color and intensity, and the viewport background color (See Fig. 9). Using these color and line thickness settings, users can create a variety of different viewport styles (See Fig. 10).

The style of the general user interface elements and the application window as a whole can be customized through the use of the TopMod stylesheet editor. The stylesheet editor allows users to select the native drawing mode of the widgets based on their preferred operating system style, such as Windows XP, Aqua, CDE, and Cleanlooks. These styles can be further customized by using the text editor

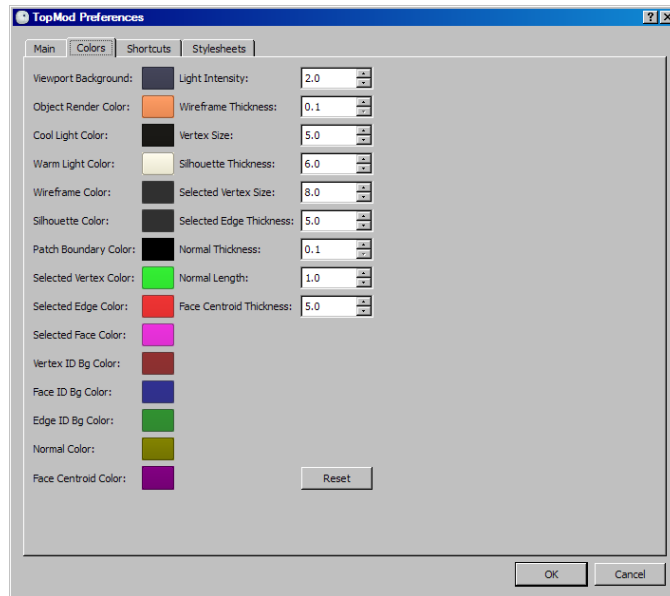


Fig. 9.: Screenshot of the color customization dialog in the TopMod preferences window.

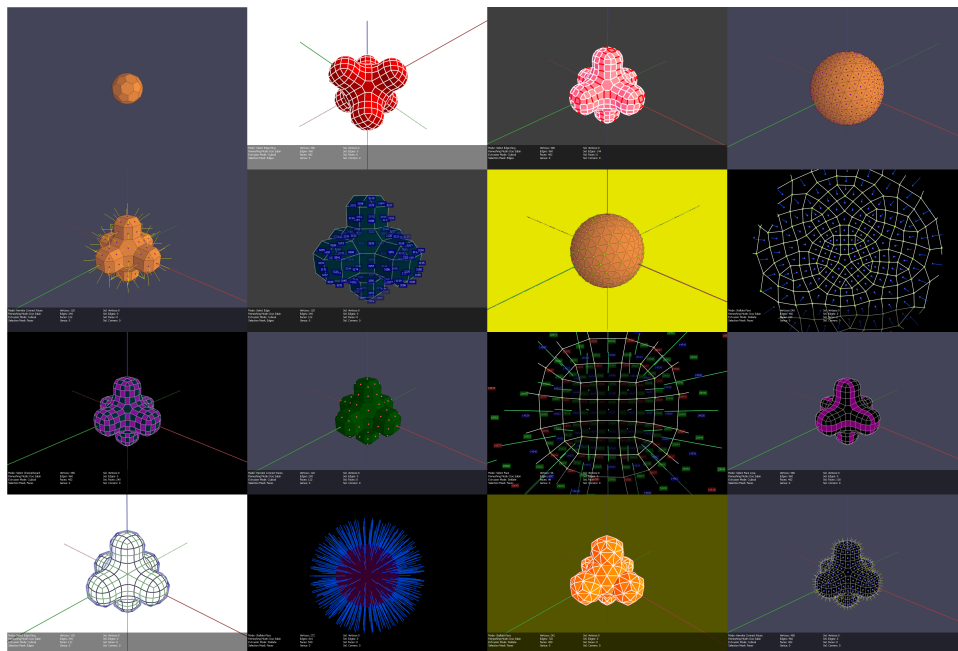


Fig. 10.: Screenshot of different viewport styles that can be created using customization options in the TopMod preferences dialog.

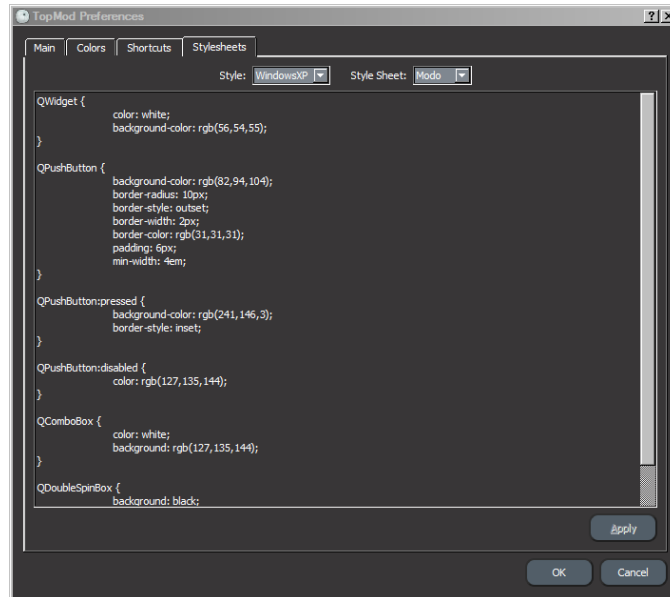


Fig. 11.: Screenshot of the stylesheet customization dialog in the TopMod preferences window.

window that allows users to type CSS-like commands to change various properties (borders, margins, padding, colors) of different widgets such as toolbars, menus, push buttons, and spin boxes (See Fig. 11). This allows users to give their version unique characteristics or settings with which they are more comfortable.

I have also implemented an interactive shortcut customization feature which allows users to set custom shortcuts for any command available in TopMod. The shortcut preferences dialog will accept keyboard input to set each shortcut (e.g. hit the CTRL key and then the X key) as opposed to having the user type in a custom string such as “CTRL+X”. The dialog will capture your keystrokes and attempt to set the custom shortcut for that command (See Fig. 12). These settings as well as others can be reached through the preferences dialog in the “Edit” menu.

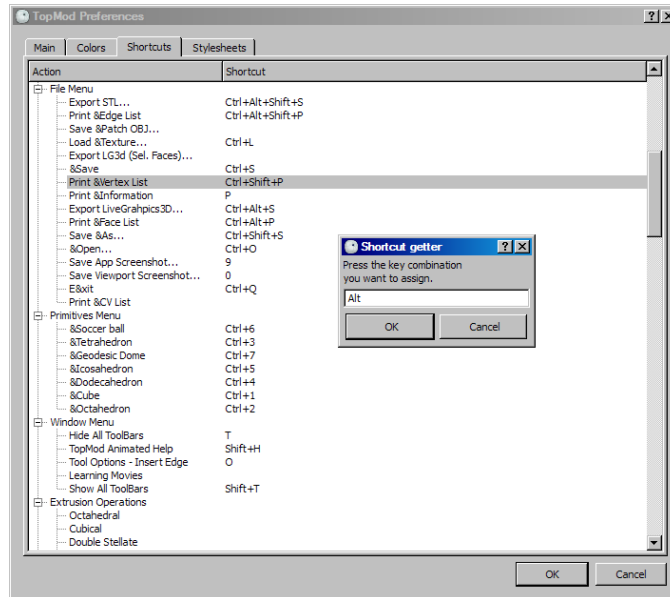


Fig. 12.: Screenshot of the shortcut customization dialog in the TopMod preferences window.

2. Popup Command Line Interface

This new version of TopMod also features a popup command line interface (CLI), similar to the command line and auto-completion functionality offered by Rhinoceros 3D. The popup CLI can be invoked by hitting the spacebar key on the keyboard. When the CLI is revealed in the center of the viewport, the user can begin typing a command, and a list will fill in a box below the text field with values that match what is currently being typed. The arrow keys or mouse can be used to choose a command from the auto-completion list, and the command can be executed by hitting the return key.

By hiding the CLI in the background, it does not take up valuable screen real estate while it is not being used, but it still provides extremely fast access through the use of the spacebar shortcut. The popup command line interface will also allow users to work in the full screen mode, which will temporarily eliminate almost all of



Fig. 13.: Screenshot of the TopMod pop-up command line interface.

the graphical input elements from the display and allow the majority of the screen to be occupied by the mesh data (See Fig. 13).

3. Context-Sensitive Right Click Menus

Users are also be able to access the majority of the commands from a series of context-sensitive right click menus. The menus will offer different lists of commands based on the users current mode of operation and their current selection in the viewport. For example, if a user is in a mode tool which requires a face or multiple faces to be selected, such as cubical extrusion mode, the right click menu will show a list of commands towards the top of the menu which have to do with selecting faces (See Fig. 14). The bottom portion of the right click menu will then be filled with commands that are common across the entire application, such as “tools,” “remeshing modes,” and “selection masks.”

4. Heads Up Display

The heads up display in the TopMod viewport window displays various information about the current state of the model and the application itself. The HUD currently displays the operating mode in which the user is working, the current remeshing mode,

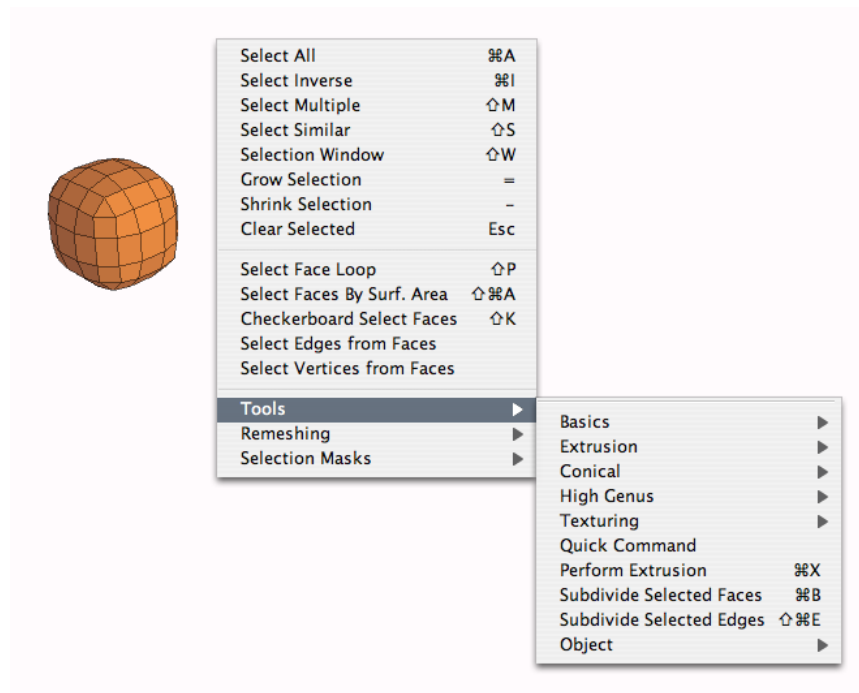


Fig. 14.: Screenshot of the face selection mode right click menu.

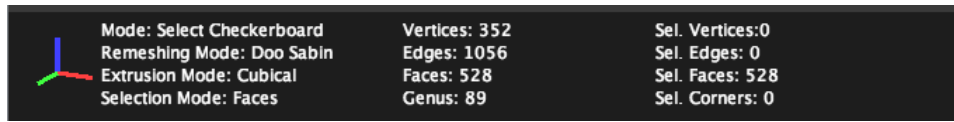


Fig. 15.: Screenshot of the TopMod heads up display, rendered in the OpenGL viewport.

which can be invoked at any time by hitting CTRL+R on the keyboard, the current extrusion mode which can be invoked by selecting faces and hitting CTRL+X, the current selection mask (faces, vertices, edges, corners, or none). The second column of the HUD displays how many faces, edges, and vertices the current model, has, and it also calculates the genus of the model (number of holes), which is a function of the number of faces, edges, and vertices. Lastly, the third column of the HUD shows how many faces, edges, vertices, and corners are currently selected in the model. This information can be beneficial in combination with the variety of new selection routines described in detail below (See Fig. 15).

C. Selection Routines

A large number of TopMod operations require some kind of sub-object selection set in order to execute such as a single edge, face, or vertex, or a multiple edges, faces, or vertices. For example, the DELETE EDGE command requires a single edge as an input, and the EXTRUDE FACE command requires a single face. In order to speed up the execution of various, often repetitive operations such as EXTRUDE FACE, I have provided a new set of selection mechanisms that allow users to select a wide variety of edges, faces, and vertices.

1. Edges

Some examples of TopMod operations that require edges as input include `DELETE EDGE`, `COLLAPSE EDGE`, and `SUBDIVIDE EDGE`. The previous version of TopMod required the user to delete and collapse edges one at a time, which required at least a mouse click for each edge that was to be deleted. It is often desirable to delete series of edges in a model that share a connection based on the structure of the mesh, such as a series of edges in a quadrilateral mesh that form a ring or a loop around part or all of the mesh. I facilitate the operation of delete an entire ring or loop of edges by providing the user with an edge ring and edge loop selection routine which can be used in conjunction with each other as well as with other selection routines such as crossing window selection, selection painting, and a `SELECT SIMILAR` command.

An edge loop is a line of edges in a quadrilateral mesh that are connected end to end. Edge loop selections are obtained by checking to see if the first side of the selected input edge is connected to exactly three other edges, and then selecting the edge which does not share a face with the currently selected edge. This process is repeated recursively for the current edge until it reaches the end of an edge which does not connect to only three other edges, or until it reaches the original selected edge (See Fig. 16).

An edge ring is a series of edges that are not connected, but sit on opposite sides of a series of quadrilaterals. Edge ring selection sets are obtained by first checking if the input selected edge is connected to exactly 3 edges on either end, and then selecting the edge opposite itself and checking again if it is connect to exactly 3 edges on either end. This process is repeated recursively until an end condition is met such as a non quadrilateral face (triangle, pentagon, etc...) or the selection reaches the original selected edge (See Fig. 16).

Edge selection sets can also be obtained through a crossing window selection mechanism as well as through a selection painting function. To obtain a crossing window selection, the user enters the selection window operating mode, and clicks the mouse in one area of the viewport, drags it across the model in the areas where selection is desired. Once the mouse is released, through the use of the OpenGL viewport picking functionality, the selection set will include every edge that fell inside the user's selection window (See Fig. 16).

Selection painting is simply a mode where the user can click and drag the mouse over various edges in the model and have each one added to the current selection set. Edges can be removed from the selection in selection painting mode by holding the CTRL key. There is no simple way to prevent the mouse from selection edges which are not actually visible in the viewport, so as the user drags the mouse around the screen, edges toward the back of the model may also be selected.

Using the SELECT SIMILAR command, edges whose vertices have the same valence can be selected simultaneously. For example, if a user selects an edge with a 3-valence vertex and a 4-valence vertex, then the rest of the edges matching that pattern will immediately be selected. Edge selection sets can also be obtained by converting a selection set of faces or vertices to the corresponding edges.

2. Faces

There are also many operations in TopMod which require faces as input, such as EXTRUDE FACE, RIND MODELING, SUBDIVIDE FACE, MULTI-FACE HANDLE, CONNECT FACES, and ADD HANDLE. The previous version of TopMod allowed the user to select multiple faces for the SUBDIVIDE FACE and MULTI-FACE HANDLE operations, but these modes still required each face to be selected individually. RIND MODELING also required the user to punch out each face individually, which was

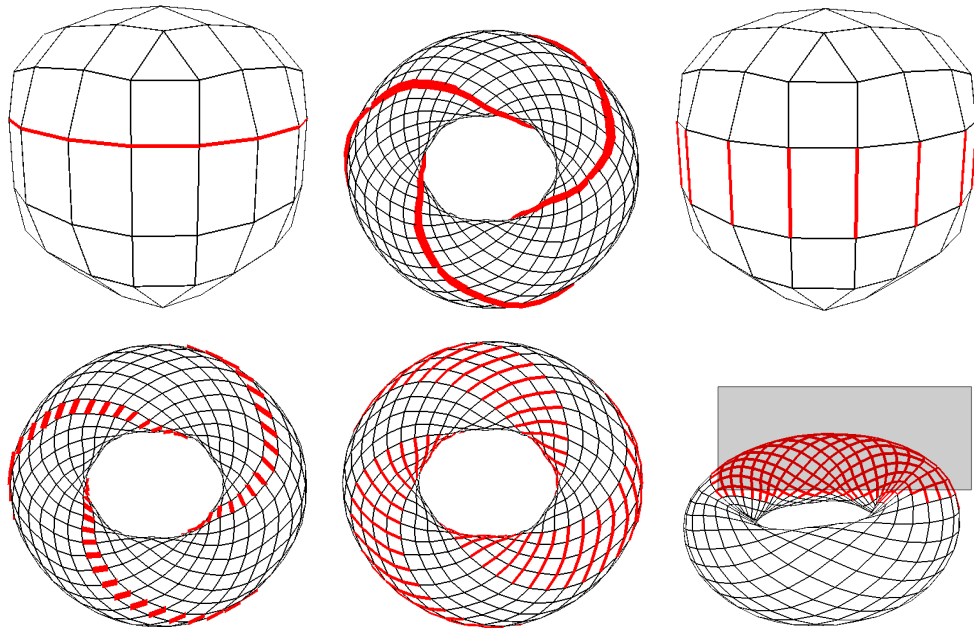


Fig. 16.: Examples of edge loop selection, edge ring selection, and crossing window selection.

quite time consuming for complex models.

In order to speed up the face selection process, I have implemented a series of face selection routines which can be used in individually or in combination such as face loop selection, select similar faces, select faces by surface area, checkerboard face selection, and crossing window selection.

A face loop selection set is obtained exactly the same way as an edge ring, except that the corresponding faces are selected instead of edges. A face loop can also be described as being enclosed by two edge loops. It is important to note that face loop selection primarily works for quadrilateral meshes, but can yield interesting results with meshes that contain other types of faces (See Fig. 17).

I have also implemented a routine which allows the user to select all faces in a model that have the same number of edges (size) as the selected face. For example, if a user selects a triangle while in select similar faces mode, all triangles will be selected

immediately (See Fig. 17). Along the same lines, there is a new routine that will allow users to select all faces in a model that have a similar surface area as a particular face. If the surface area of a face is equal to or close to the are of the selected face within a user-selected tolerance value, then it is also selected. This allows a variety of new patterns to be selected in one click as opposed to clicking once for each face (See Fig. 17).

A Checkerboard selection can be obtained by recursively selecting all faces that share exactly one corner with the currently selected face. This is a phenomenon that works quite well in conjunction with the Doo Sabin remeshing algorithm (See Fig. 17).

Crossing window selection and selection painting is also implemented for faces, but OpenGL allows faces which are not visible in the viewport to be excluded from the paint selection set, unlike edges. These routines are helpful for operations such as EXTRUDE FACE and RIND MODELING.

3. Vertices

Using the same process described above, I have implemented a crossing window selection functionality for selecting multiple vertices. This allows users to remove unwanted vertices from the model, which effectively performs the DELETE EDGE operation on all edges connected to the selected vertices. Users can also use the SELECT SIMILAR routine to select all vertices in a model with the same valence, or number of edges connected to them.

I have also added routines that allow the user to swap the current sub-object selection set for the corresponding set of a different sub-object type. For example, a user may select a series of edge loops, edge rings, or face loops, and convert that selection to vertices, edges, or faces, depending on what is currently selected (See

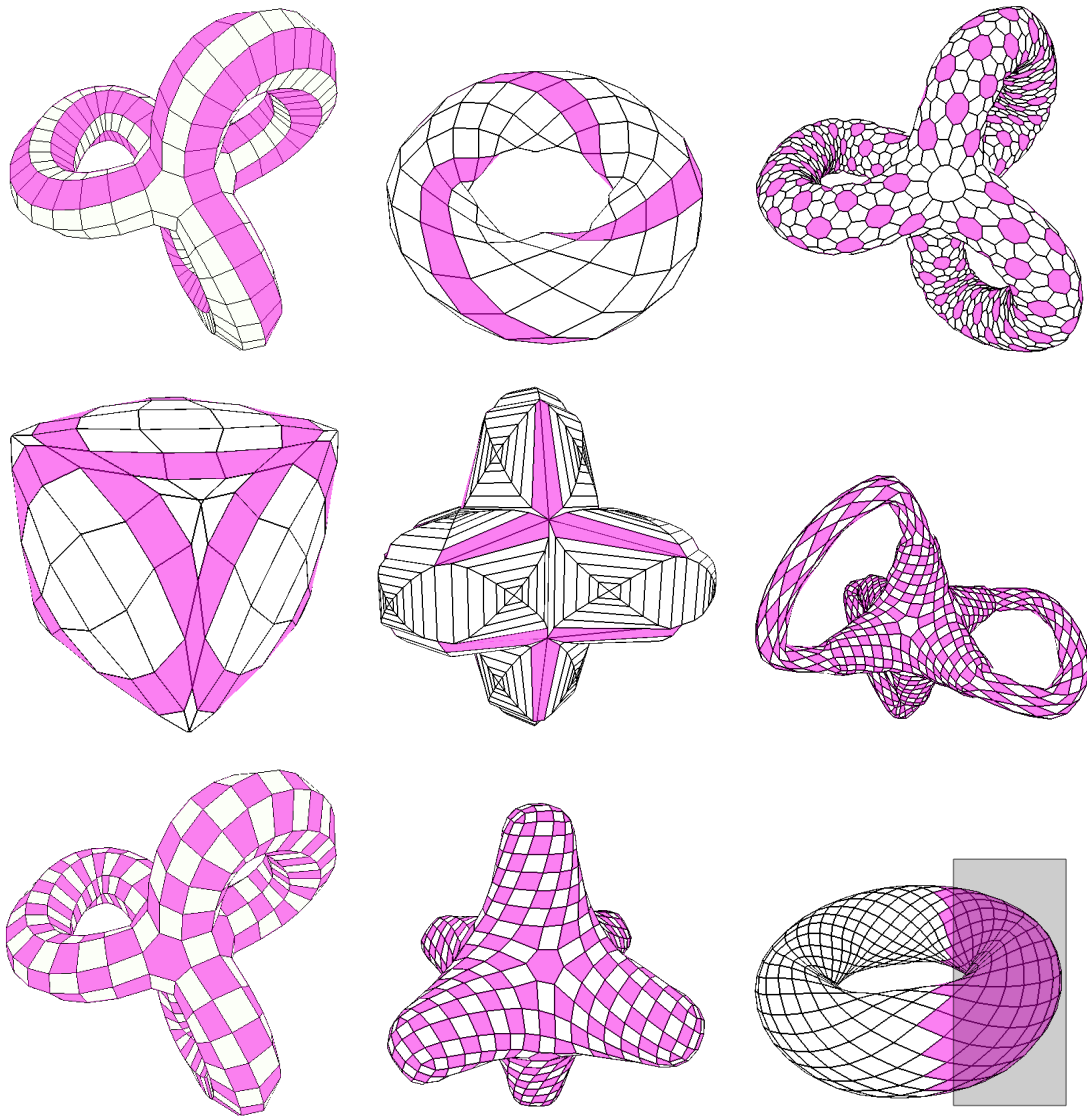


Fig. 17.: Examples of face loop selection, select similar faces, select faces by surface area, and checkerboard face selection, and crossing window selection.

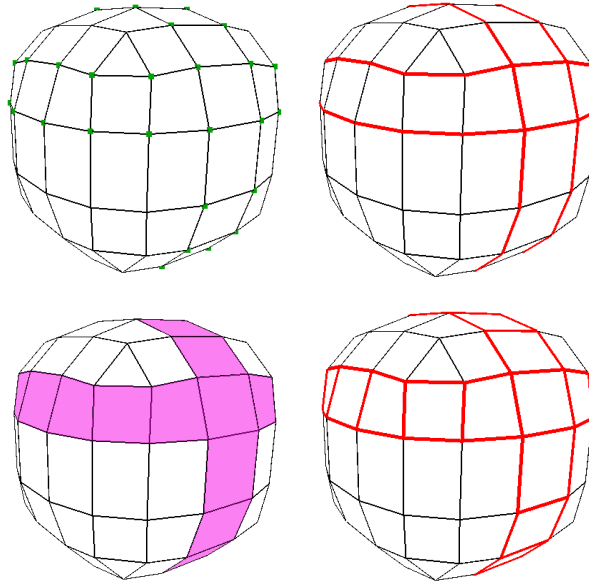


Fig. 18.: Examples of converting vertex and face selections to edges.

Fig. 18). This functionality provides even more methods for obtaining interesting sub-object selection sets and ultimately makes modeling in TopMod faster and easier.

A previous version of TopMod provided an “edit vertex” functionality added by Brian Barran which allowed users to select individual vertices and move them along all three axes. I have recovered this code and integrated it into the new version as part of the sub-object selection menus.

4. Growing and Shrinking Selection Sets

Another interesting and powerful selection mechanism I have implemented in the new version of TopMod gives users the ability to expand (grow) and contract (shrink) the currently selected sub-objects (faces, edges, or vertices). The growing and shrinking methods work slightly differently for faces, edges, and vertices, but they accomplish the same basic task.

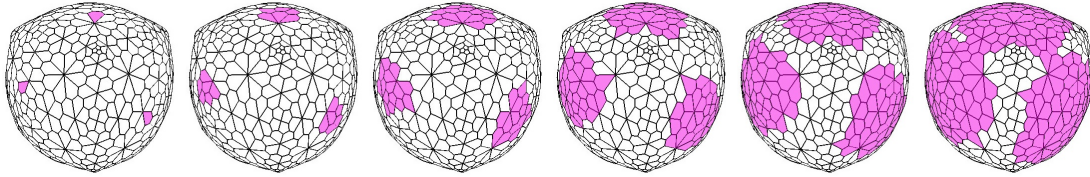


Fig. 19.: Screenshots illustrating the selection expansion (growing) and contraction (shrinking) process for faces.

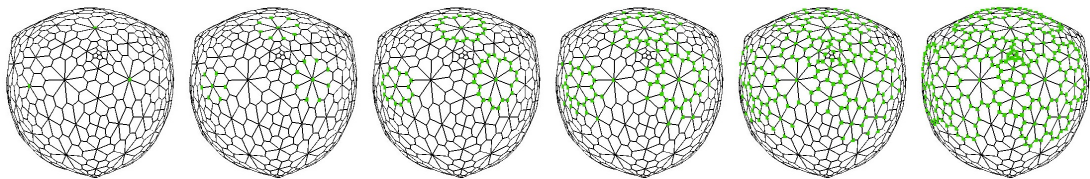


Fig. 20.: Screenshots illustrating the selection expansion (growing) and contraction (shrinking) process for vertices.

To expand the selection for a single face on a surface, all faces which share an edge with the selected face or faces will be included in the next selection set. This process can be reversed by deselecting all the currently selected faces which have a neighboring face that is not also selected. This will shrinking or contracting operation remove all boundary faces from the selection set (See Fig. 19).

To expand the selection for vertices on a surface, all vertices which share an edge with the selected vertex or vertices will be included in the next selection set. This process can also be reversed by deselecting all the currently selected vertices which have a neighboring vertex (through a shared edge) that is not also selected. This operation will remove all boundary vertices from the selection set (See Fig. 20).

Lastly, to expand the selection for edges on a surface, all edges which share a vertex with the selected edge or edges will be included in the next selection set. This

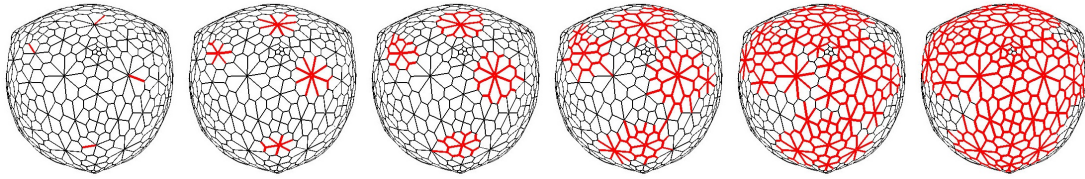


Fig. 21.: Screenshots illustrating the selection expansion (growing) and contraction (shrinking) process for edges.

process can be reversed by deselecting all the currently selected edges which have a neighboring edge that is also not selected. This shrinking or contracting operation will remove all boundary edges from the selection set (See Fig. 21).

D. Thesis Research Consolidation

Over the last six years a number of students in the Visualization Sciences program have contributed to TopMod through personal interest and thesis research projects. A student working on the TopMod source code would typically download the code to their own directory, learn how it worked, add their own code to extend functionality of existing features or to create entirely new features, and test the results on the lab computers. It was not often that the code written into their own versions was integrated back into the main version on the server. For that matter, there was never really a main centralized version of TopMod to which code could be added. As a result, there were many different versions of TopMod floating around the lab, and no one had the opportunity to experience all the new features in one single version. I decided to locate some of the projects that were never introduced into the public version of the software and add them into a final centralized code repository. In searching through the various versions of the TopMod source code, I also exposed

some interesting hidden features from the source code to the new user interface.

1. Planar Modeling

The thesis research of Ozgur Gonen involved implementing a series of planar modeling or conical sculpting operators in TopMod. The main operators implemented included CUT BY EDGE, CUT BY VERTEX, CUT BY FACE, and CREATE CONVEX HULL. All of these operations guaranteed the results to be planar, which is quite helpful when unfolding or developing the surfaces onto 2-dimensional planes to be cut out of paper and folded back into physical 3-dimensional models. The implementation of the planar modeling operations was incomplete and buggy, and only allowed users to cut one edge, vertex or face at a time. In the new version, I rewrote the user interface for planar modeling to allow any edge, vertex, or face selection set to be cut all at once by hitting the “Cut Selected” button or by invoking the shortcut key CTRL+T. This allows users to work much more quickly and easily with these tools, and helps ensure that the new version of TopMod includes as much of the prior research as possible (See Fig. 22).

2. Advanced Extrusions

Eric Landreneau added a series of extrusion operators to a previous version of TopMod as part of his thesis research, that for whatever reason did not quite make it into the main version. These included a dodecahedral and an icosahedral extrusion mode that gave users the ability to produce some interesting organic forms. After realizing that these features were not present in the most recent version of the software, I contacted Eric, located the code that did contain these extrusion modes and integrated it into the new repository and added the necessary interface widgets. These extrusion modes are extremely versatile, and can produce very interesting results when used in

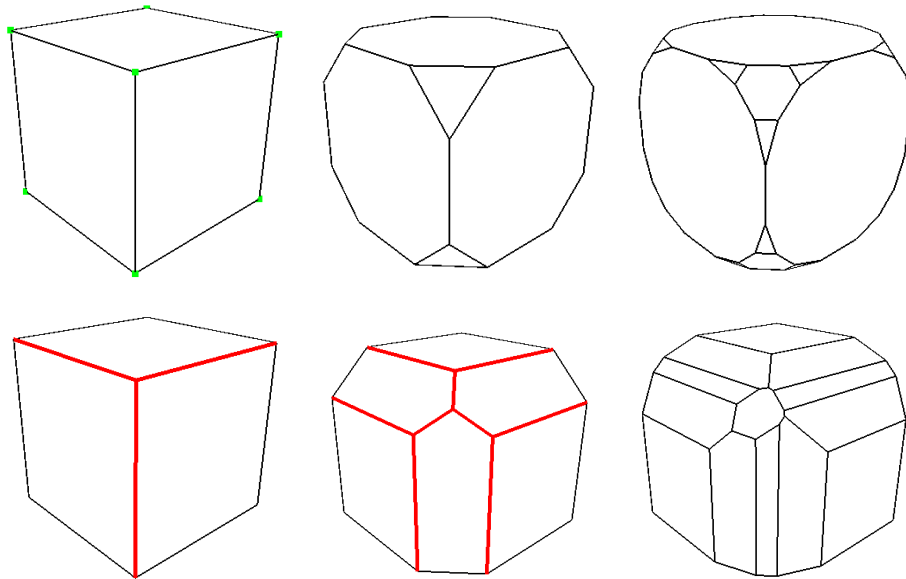


Fig. 22.: Screenshots demonstrating the process of cutting multiple vertices and edges.

conjunction with various remeshing modes (See Fig. 23).

3. Bezier Patch Rendering

Dr. Akleman and Dr. Srinivasan implemented a Bezier patch rendering functionality in one branch of TopMod that allowed users to view models as bezier patch surfaces, and Stuart Tett implemented an export feature that allows the bezier patch model to be exported as an .OBJ file, which can be imported into any 3D software that supports Bezier Patch OBJ files, such as Maya. These features were first implemented in different versions of the source code, and I made sure the necessary code from each version was integrated into the final code repository and functioned properly with the new Qt-based interface (See Fig. 24).

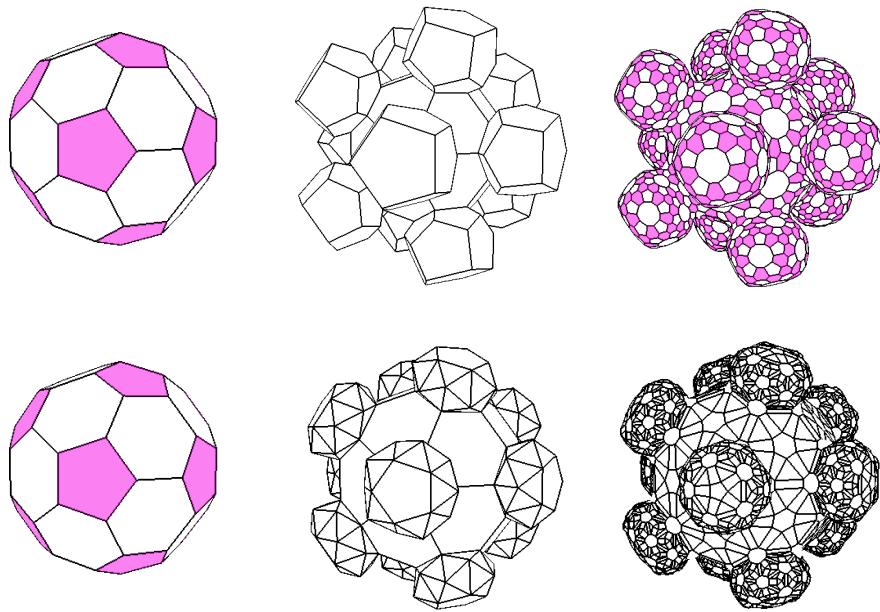


Fig. 23.: Two of the advanced extrusion operators from Eric Landreneau's thesis research, dodecahedral and octahedral, used in conjunction with various remeshing modes.

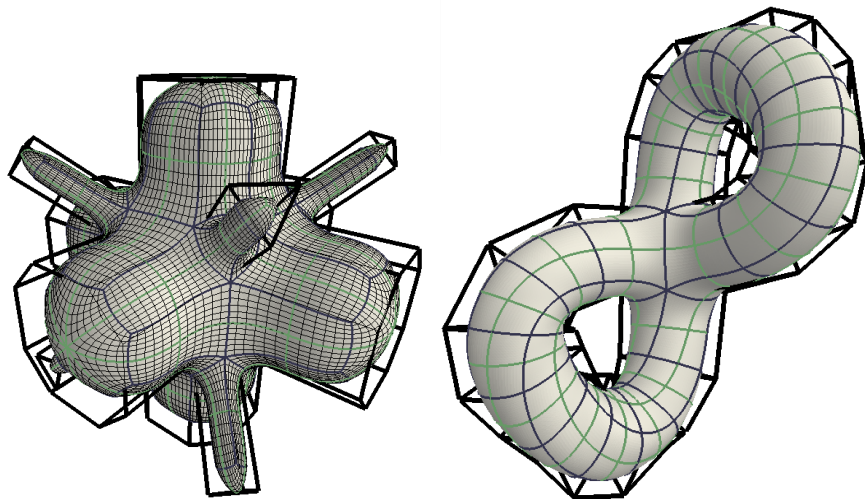


Fig. 24.: Screenshots of bezier patch models in the TopMod viewport.

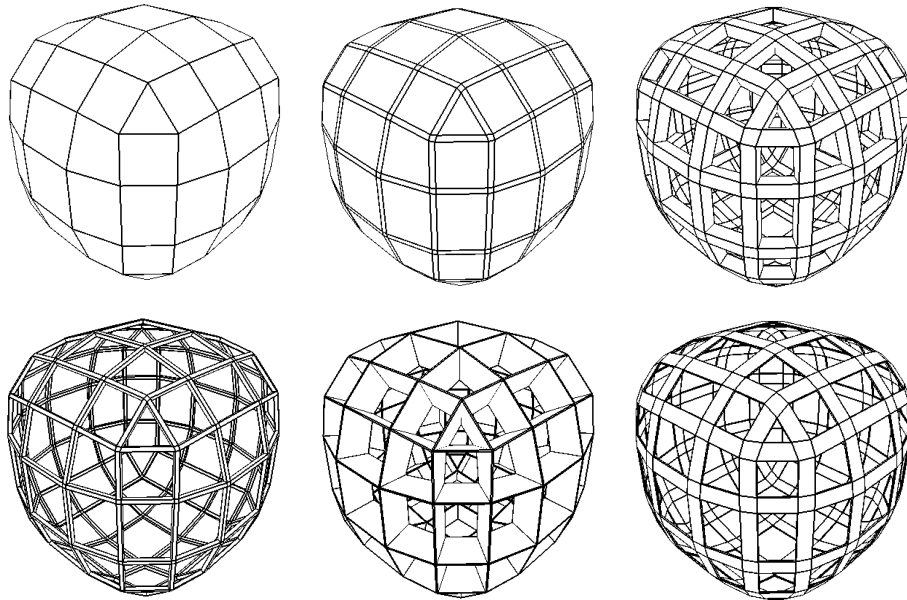


Fig. 25.: A series of wireframe models with different cross-sectional dimensions.

4. Rectangular Wireframe Modeling

The thesis work of Esan Mandal involved implementing a wireframe modeling feature in TopMod, that converts all of the edges in a model to square cross-sectional tubes that connect to form a solid mesh. After looking through the source code at the functions that make this feature possible, I realized that the square cross section was being determined by two separate values that the user interface only allowed to be set to the same value. By exposing two floating point spin boxes to the user interface, users can now create wireframe models with rectangular cross sections (See Fig. 25). This new mode can be especially helpful in designing architectural structures with TopMod.

5. New Approach to Interactive Rind Modeling

Another specialized feature in TopMod that allows users to create high genus models interactively is the rind modeling tool. In order to create a rind, or crust, in TopMod, users typically select a crust scale or thickness value, click the “Create Crust” button, and begin punching out faces one by one by clicking on them in the viewport. Each face the user selects pokes a hole in the model. This method of creating crusts proves to be quite time consuming for models with a large number of polygons, as each face requires a mouse click. Undoing a mistake required the user to restart the entire process from the beginning, which happened quite often, and ultimately discouraged users from the rind modeling tool as it took too much time and precision. In the new version of TopMod, I have redefined the process for creating rind models by allowing the user to select any number of faces on the model first. After a face selection set is obtained, the “Create Crust” button or shortcut key combination can be invoked, which punches out all of the selected faces at once and reveals the crust. This operation can be undone by simply hitting “Undo”, and the user can start over (See Fig. 26).

6. New Approach to Extrusions

In the previous versions of TopMod, users had the ability to extrude one face in the model at a time by selecting an extrusion mode and clicking on a single face. After letting go of the mouse button, the face would be extruded. TopMod required users to click once for every single face they wanted to extrude. For complex models of more than 1,000 faces, it would be extremely tedious and time-consuming to extrude every face by clicking on each one. I implemented a new feature in TopMod that allows users to select any number of faces using the face selection routines mentioned

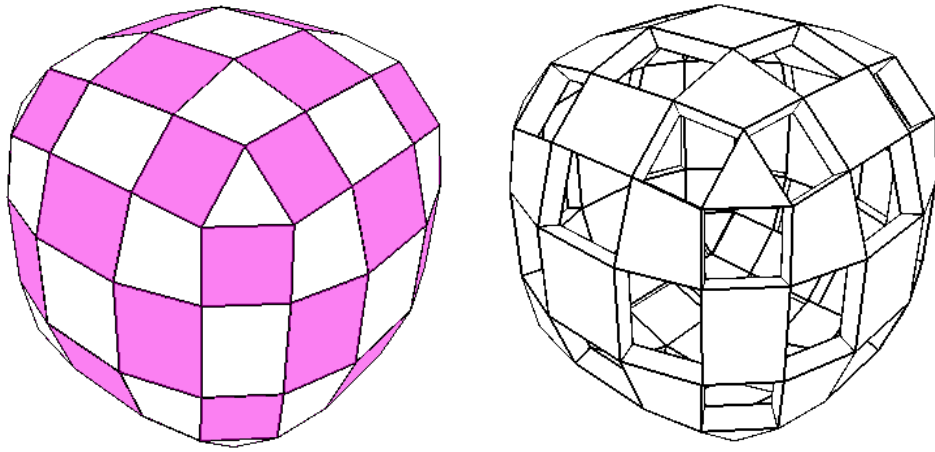


Fig. 26.: Example of a face selection set that is rind modeled all at once using the new rind modeling feature in TopMod.

above, and then extrude all of the selected faces at once by hitting CTRL+X or by pressing the “Extrude Selected Faces” button in the tool options window for each extrusion mode. Now users can select as many faces as they want in a variety of different ways, and extrude them all at once. This drastically reduces the amount of time it takes to create extrusions on models in TopMod, and makes the overall user experience much more interesting (See Fig. 27).

7. New Approach to Other Basic Operators

I also implemented similar features to the extrusion mode described above with other operating modes in TopMod. For example, users now have the ability to selected any number of edges and delete, subdivide, collapse them. Face selection sets can be subdivided, vertex selection sets can be deleted, which will simply delete all connected edges. These simple interface adjustments are extremely powerful when working on complex models, and they allow for a much faster workflow.

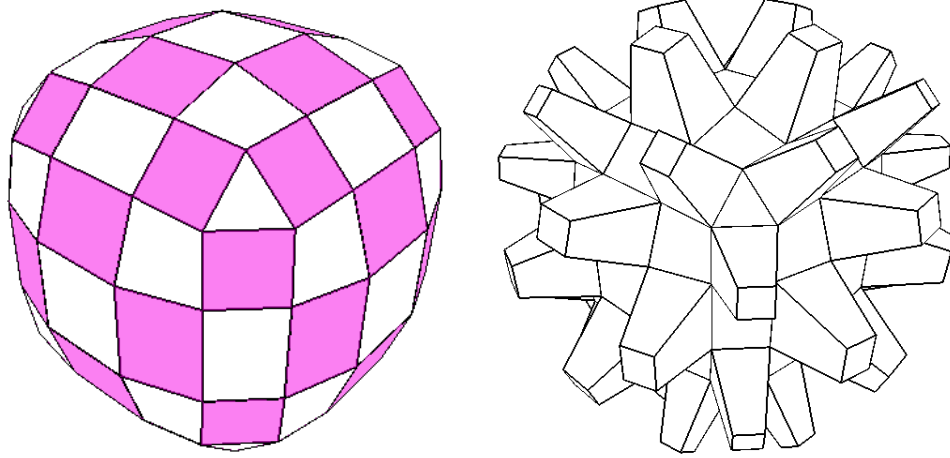


Fig. 27.: Example of a face selection set that is extruded all at once using the new extrusion feature in TopMod.

8. Scripting Interface

Stuart Tett's thesis research involved writing a scripting engine to expose TopMod's core and auxiliary operations to allow users to quickly and easily perform repetitive tasks as well as extend the functionality of TopMod without editing the source code. In order to allow users to write scripts interactively within the TopMod interface, a custom script editor interface needed to be implemented in the new version of TopMod. I helped Stuart through the process of creating the scripting interface using Qt's interface widgets, and made sure the code was consolidated into the final version (See Fig. 28). I also manage the various releases of TopMod that contain the scripting interface, as they need to be separated from the normal TopMod application that does not require Python to be installed.

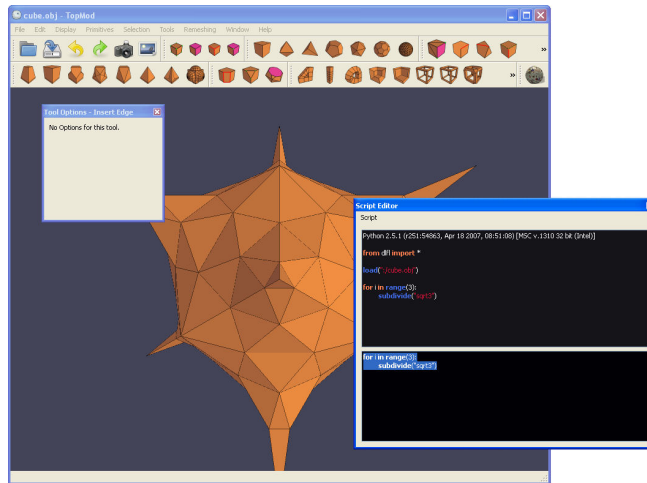


Fig. 28.: Screenshot from TopMod with the Python scripting interface by Stuart Tett.

E. Portability and Platform Independence

It is often desirable to run an application on an operating system or user account which does not necessarily have permission or sufficient privileges to install software that changes contents of system-wide files or directories. This is often the case with large companies as well as university computer labs. In order to allow TopMod to run in these situations as well as to make updating the program easier, it has been designed to run portably from any location on a user's hard drive. The program can be downloaded from the web site as a zipped archive file and copied into and run from any directory to which the user has access.

Qt provides a platform independent build system called qmake which allows programmers to store all the information necessary to build a particular application in one configuration file. This file contains a minimal amount of platform-specific build instructions, and will generate the necessary Makefiles for Windows, OS/X, as well as a variety of Linux distributions. The configuration file for TopMod was written

to support Windows XP and Vista, OS/X for PPC and Intel-based macs (creates a universal binary), Ubuntu, and Gentoo Linux. It can be extended to work on virtually any other Linux distribution with minor modifications as well. By supporting a wide variety of operating systems, TopMod is accessible to a much larger number of users and has the potential to expand its user base much more easily.

F. Internationalization

Internationalization is another highly requested feature of TopMod. Through the use of the Google Analytics web site statistics tools running on <http://www.topmod3d.org/>, I have noticed a large portion of the pageviews coming from the European Union, specifically Germany, Italy, France, and Spain. In addition to these countries, many of the previous TopMod developers come from other countries such as Turkey and India.

As with many other popular 3D modeling applications such as Blender, Maya, and Google SketchUp, it would be beneficial to the 3D modeling community at large to offer internationalization (i18n) support within TopMod. The Qt open source widget toolkit provides a set of classes that allow for real time translation of applications independent of the development platform. The TopMod source code is set up in such a way as to allow for real time translation into Spanish, Italian, German, French, Turkish, and Catalan. Translation data for each language is stored in an xml file that can be edited by volunteer users who speak any of these languages. Qt provides a platform-independent translation application that loads the xml file into an easy to use GUI that allowing editors to translate each string of text from the TopMod interface. The translator application also provides predefined language dictionaries that will suggest translations and translate certain keywords automatically (See Fig.

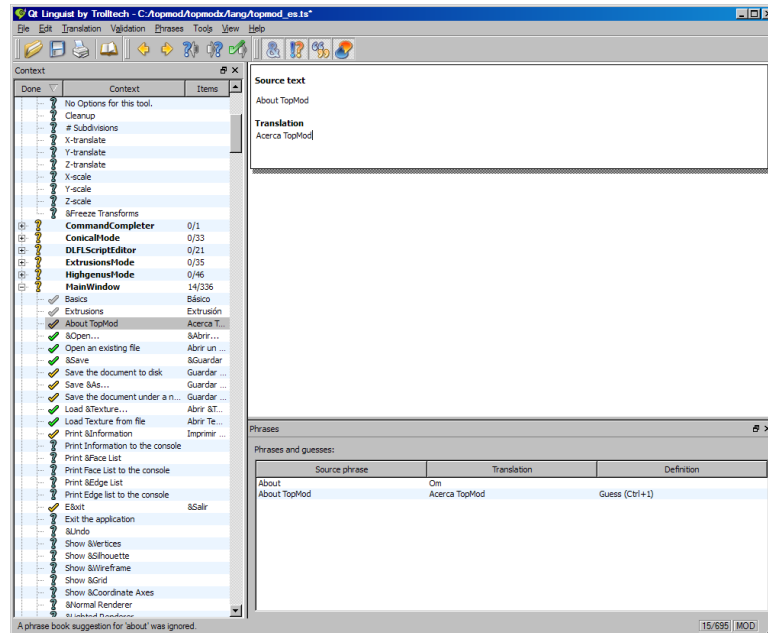


Fig. 29.: Screenshot from the Qt Linguist application.

29).

Once a translation file is complete, it can be resubmitted to the code repository and compiled into TopMod releases. TopMod can be translated by choosing a language from the “Help-¿Language” menu. At this time, TopMod has been translated into French by a member of the discussion forum, Frençy Pilou, and is currently being translated into German, Spanish, Italian, and Turkish by other members of the TopMod community (See Fig. 30).

G. Online Community

An old version of TopMod has been available since 2005 through Dr. Ergun Akleman’s web page at <http://www-viz.tamu.edu/faculty/ergun/research/topology/>. Through various google search terms such as “topmod,” “topological”, and “mesh modeling”,

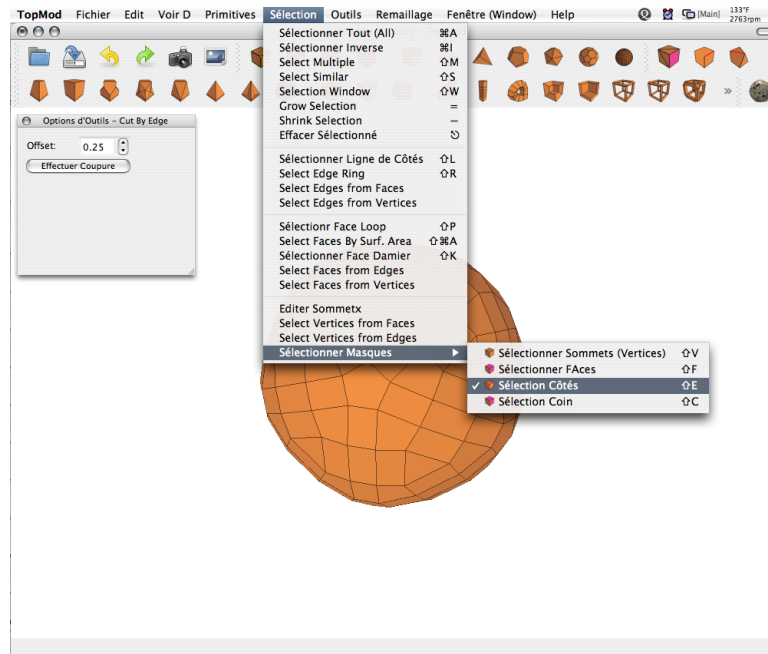


Fig. 30.: Screenshot of the TopMod interface translated into French.

many people have discovered the software and have been using it and publishing renderings and images all over the internet. As I began to work on the new version, I gradually discovered more and more of a community of TopMod users from all over the world who were eager to see new features added and updates made to the interface, especially through the Jotero Discussion Forum at <http://forum.jotero.com/>. Torolf Sauermann, who maintains the Jotero Discussion Forum as well as his own personal image gallery website at <http://www.evolution-of-genius.de/> (See Fig. 31 and 32).

After seeing just a glimpse of the amount of 3D content on the web that had been created or edited using TopMod, it became apparent that TopMod needed to establish a better presence on the web. TopMod needed a new web page that provided many of the services other popular 3D modeling applications provide, in order to serve current and future users of the software and encourage online collaboration and future

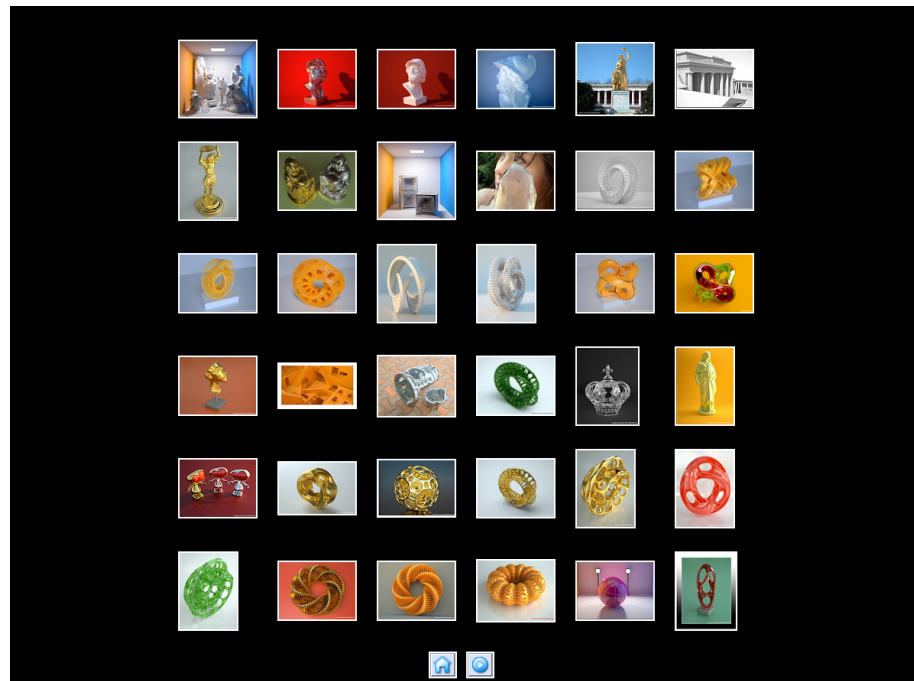


Fig. 31.: Screenshot of the Jotero Image Gallery, by Torolf Sauermann.

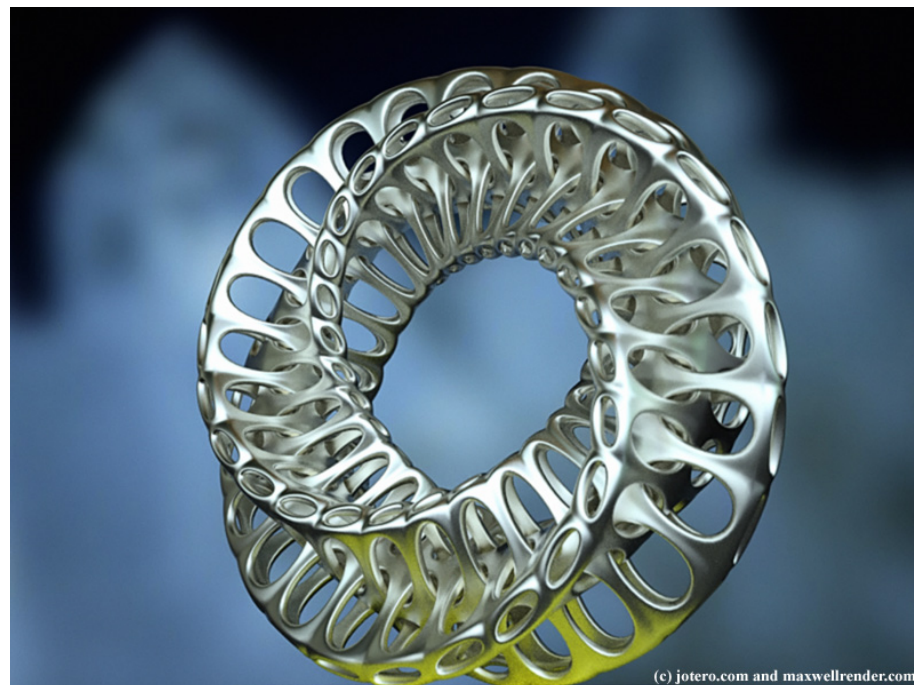


Fig. 32.: Rendering from the Jotero Image Gallery, by Torolf Sauermann.

development. The new TopMod web page would need to offer a blog or news feed to allow users and developers to keep track of updates to the software, a discussion forum to encourage communication about how to use the software and to report errors and give feedback, an image gallery to display models made using the software, and a user manual to teach new and existing users how to use all of TopMod's features.

Additional features that would be needed for the TopMod web site included developer's documentation, a subversion repository to host the code, and a place to store tutorial videos as a supplement to the user manual. In this section I describe the various technologies used to create the new TopMod user community web site.

1. Blog

Many popular 3D modeling application web sites feature blogs that discuss new developments and other news regarding the particular piece of software, such as <http://www.blendernation.com/> for Blender and the Google Sketchup blog at <http://sketchupdate.blogspot.com/>. The BlenderNation blog uses the WordPress Weblog platform software, while the Sketchup blog uses a Google blogging service known as Blogger (<http://www.blogger.com/>). I needed the TopMod blog to be integrated with the rest of the site and not sit on an external server such as <http://www.blogger.com/>, so I decided to try out a few different open source blogging platforms. The server where <http://www.topmod3d.org/> pointed to at the time of this writing provided support for PHP/MySQL web applications, so I was limited to trying out b2evolution, WordPress, ExpressionEngine, and Nucleus. After evaluating these web publishing platforms, I decided to use WordPress for a variety of reasons.

First of all, WordPress takes about five minutes to install on a PHP/MySQL-capable server, and provides an extremely simple and easy to use administration interface (See Fig. 33). It is quite easy to create and edit categories for blog posts,

The figure consists of three side-by-side screenshots of the WordPress administration interface for the topmod3d.org blog. The leftmost screenshot shows the 'Last 15 Posts' section, featuring a search bar and a table of recent posts. The table has columns for ID, When, Title, Categories, Comments, and Author. The middle screenshot shows the 'Plugin Management' page, which lists installed plugins such as Akismet, Atomic Media Player, Feed List, Google Analytics, Google XML Sitemaps, Hello Dolly, My Tube, PayPal Donate, and WPML. The rightmost screenshot shows the main blog page, which includes a sidebar with a 'blip.tv' widget and other content.

Fig. 33.: Screenshots from the topmod3d.org blog administration interface, the plugin panel, and the main page. The main page style was adapted from the Big Blue theme created by Bob Jiwakacau.

and also allows for the creation of static pages to store constant information such as an “About” page. WordPress is highly extensible, as well, with a variety of plugins for displaying videos inside blog posts, displaying custom RSS feeds in sidebar templates (See Fig. 33), and generating sitemap files to speed up search engine indexing of new content.

In addition, WordPress supports custom stylesheets and templating to allow designers to quickly and easily create unique presentations (themes) of their blog content. With a basic knowledge of CSS and PHP and by starting with an existing WordPress theme, a WordPress blog site can be customized in under an hour. The TopMod blog features a dynamically generated horizontal menu across the top with access to the main areas of the site, and two right sidebars which display various lists, such as links to external web content about TopMod, random images from the TopMod image gallery, recent posts on the TopMod forum, recently uploaded TopMod tutorial videos, as well as recently tagged links from <http://del.icio.us/> with

the keyword “topmod” (See Fig. 33).

2. Discussion Forum

Web applications designed for holding discussions and posting user generated content have been in existence in some form or fashion since the 1980s, and have been integral in developing and maintaining virtual communities centered around a specific subject matter. Discussion forums dedicated to particular software applications have become somewhat of a necessity for even the simplest of applications. Previously, the only outlet for discussing issues and ideas related to TopMod publicly on the internet was through the Jotero Discussion Forum (<http://forum.jotero.com/>), which is run on the free open source phpBB forum software.

After evaluating a variety of discussion forum software solutions such as vBulletin (<http://www.vbulletin.com/>), phpBB (<http://www.phpbb.com/>), SimpleMachinesForum (<http://www.simplemachines.org/>), and many others available at <http://www.forummatrix.org/>, I decided to use phpBB3, the current version at the time of this writing. phpBB is free and open source, and uses PHP/MySQL server technologies. Like WordPress, it is extremely easy to install and configure, and can be stored in the same database as WordPress by adding a table prefix such as “bb”. phpBB also contains a robust templating engine that allowed me to copy the style I had previously created for the TopMod blog in a short amount of time. The phpBB administration panel provides a number of advanced features such as forum pruning, spam filtering, user management, and site statistics, and the forum itself allows users to upload images, 3D model files, and scripts, post quoting and styling, etc (See Fig. 34).

It also has a robust plugin architecture that allows developers to extend the functionality even further. For example, I have the ability to periodically generate

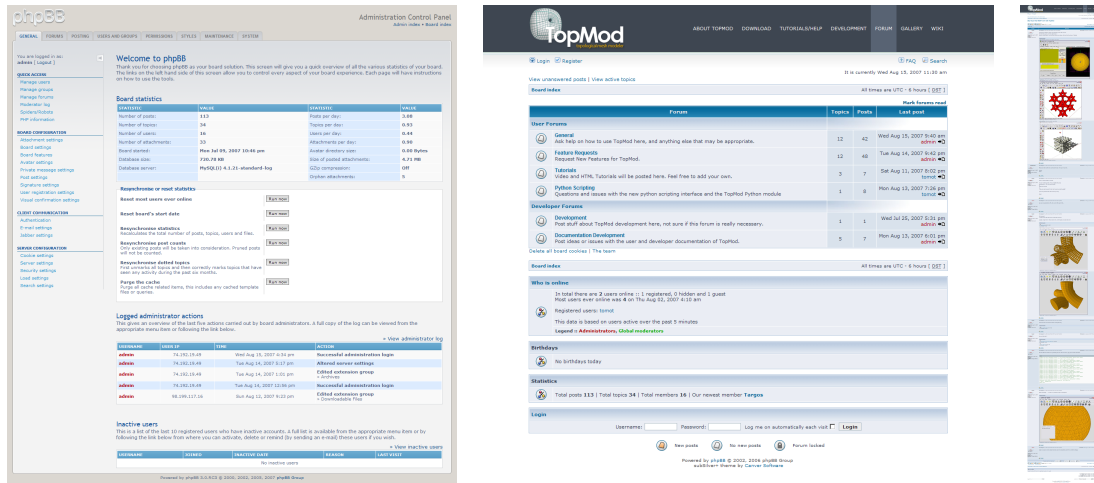


Fig. 34.: Screenshots from the topmod3d.org forum administration panel, the main page, and an example forum post.

sitemap XML files for submission to Google and other search engines, and I can eventually provide a unified secure login system for the TopMod blog, user manual, image gallery, and discussion forum in the future.

3. Image Gallery

Like many other popular 3D modeling applications such as Blender, ZBrush, and Silo3D, the TopMod web site needed a system that would allow users to upload images of models created or edited with TopMod. Many such web applications existed at the time of this writing including 4images, Gallery by Menalto, and Coppermine Photo Gallery. After evaluating these and other image gallery web applications, I decided to use Menalto's Gallery application, because it was designed to run on a PHP/MySQL enabled server, was free and open source like the other technologies I have used so far, and was highly customizable with a robust template and plugin architecture.

Gallery2, the current version at this time, allows individual users to create their

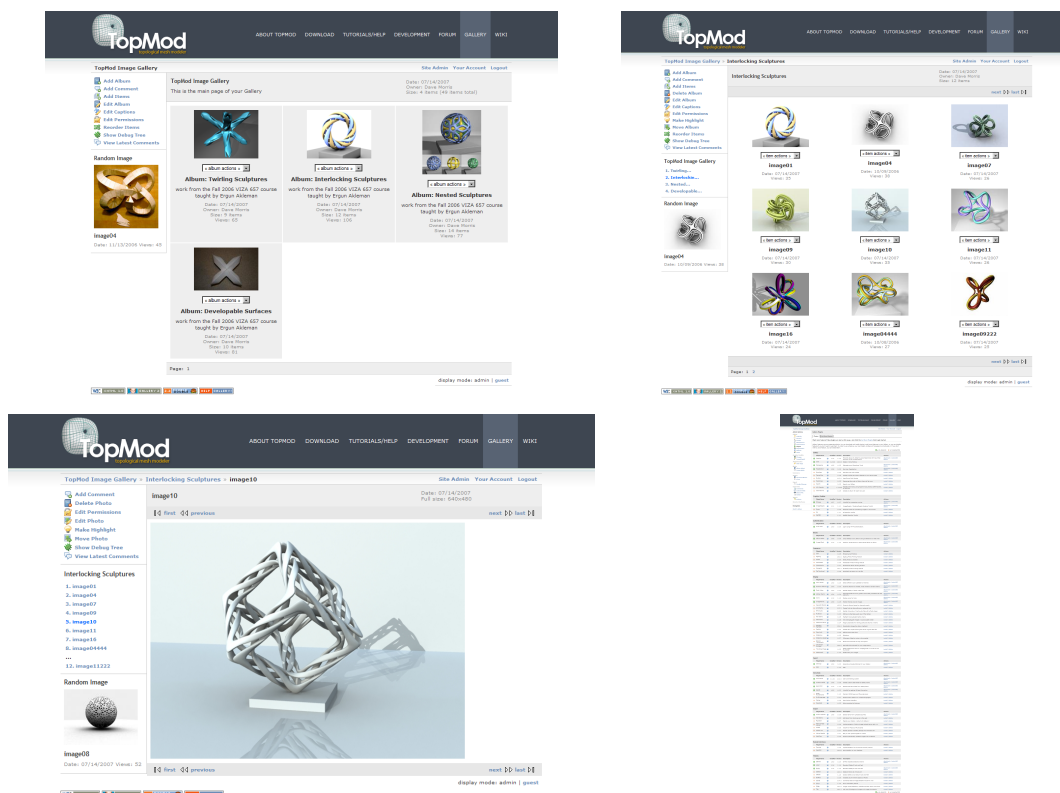


Fig. 35.: Screenshots from the TopMod Image Gallery main page, sample image gallery page, sample image page, and the administration panel.

own albums, make them public or private, as well as upload images to other public albums. It also accepts the widest variety of image file formats, and allows users to upload in virtually any manner including from the web browser, a mass upload java applet, a Windows Explorer plugin, a Picasa uploader plugin, and a commandline upload script as well. Developers can write custom stylesheets and templates for individual Gallery installs as well, which allowed me to match the same style of the TopMod blog and discussion forum quickly and easily (See Fig. 35).

4. User Manual

3D modeling applications do not often come without some sort of user manual, and TopMod should be no exception. The delivery format for 3D software user manuals differs greatly between applications, however, and it is important to be aware of the various possibilities. Maya, for example, comes with a separate documentation application that actually runs as a local web server application on the user's machine. This requires the user to install another application and requires a large amount of memory, but provides amazing search capabilities and fast lookup.

Blender, on the other hand, offers its user documentation through a collaborative wiki software online. This allows any Blender user to add to the documentation as they see fit and collaborate with others on how to approach documenting each piece of the program. However, this approach requires the user to be connected to the internet whenever they wish to view the documentation. As high speed internet connections are quite prevalent all over the United States and the European Union, this has not been a deterrent in Blender's decision to move all their documentation to an online wiki format [5]. Other applications such as 3D Studio Max offer user documentation in a local help file (*.chm and *.hlp) which does not require internet access, but limits the ability to collaborate with users on the process of documentation.

Since TopMod had little to no documentation before I began work on this thesis, I had the opportunity to start from scratch with the entire process. For an application as small and with as much room to expand as TopMod, it is more important to allow users in the TopMod community to collaborate on the documentation than to allow users without internet to access the documentation, so I decided to model TopMod's documentation after the Blender Documentation Project [38]. I needed a free open source collaborative wiki web application that would be PHP/MySQL

compatible, and Mediawiki, the software that drives <http://wikipedia.org/>, proved to be the perfect solution.

Mediawiki, like WordPress, phpBB, and Gallery, can be installed and configured in a matter of minutes, and also provides a robust templating system and plugin architecture. I was able to match the style of the TopMod main web site in a short amount of time and install and modify existing plugins to allow users to embed quicktime videos into wiki pages, generate Google sitemap files, and display custom RSS feeds.

The mediawiki templating engine allows users and developers to create customized display templates for different types of data, and all the templates remain open source to allow others to borrow templates from existing mediawiki sites. The BlenderWiki (<http://wiki.blender.org/>) uses Mediawiki as well, and I was able to borrow a large number of page templates from them, including templates for displaying keyboard shortcuts, tables, and tooltips. By starting from the BlenderWiki templates, I was able to easily modify them to fit the needs of the TopMod Wiki and extend them even further to fit into other parts of the wiki.

The TopMod Wiki contains a wealth of information about TopMod and is divided up into three main sections: the user documentation, the development documentation, and the wiki writing documentation (See Fig. 36). The user manual features pages that describe the new interface in detail as well as each operation. There is also a model index, a Python script index, a tutorial index, and custom templates for displaying models and scripts on individual pages (See Fig. 36).

In order to show models on individual wiki pages without having to take screenshots or create rendered images of each one, I found a Java applet known as LiveGraphics3D, created by Martin Kraus, that is used heavily on <http://www.mathworld.com/>, that can display complex 3D models within the browser window, and allows

the user to interactively rotate, zoom and pan the model. I have implemented a Mediawiki extension loosely based on the Quicktime Embed extension described above that allows this Java applet to be easily embedded into pages on the TopMod Wiki.

I recently contacted Martin Kraus, developer of the LiveGraphics3D Java applet, with a proposal to extend the functionality of LiveGraphics3D to allow for synchronization of the camera position and orientation for multiple applets embedded in the same web page. We briefly discussed via email how the implementation would work, and a few days later he Martin sent me the new version with the changes we had discussed. I was not able to contribute to the code for the project as it is closed source, but the new features I suggested have allowed me to create more understandable documentation pages on the wiki which show a before and after model that illustrates in 3D what a particular operation such as Catmull-Clark subdivision does.

5. Source Code Documentation

TopMod was also in need of a method for storing source code documentation and a way for current and potential developers to access the same version of that documentation. To accomplish this task, I chose a popular C++ source code documentation utility known as Doxygen (<http://www.stack.nl/~dimitri/doxygen/>). Doxygen allows C++ developers to write comments in different places throughout the source code for a particular project using certain keywords and character combinations, that when run through the Doxygen processor, can generate hyperlinked HTML files that thoroughly describe the object oriented class structure, class and file dependencies, namespaces, and variables. I have written a BASH script that will run the source code through the Doxygen processor using a custom HTML header and footer and CSS file, zip up the generated HTML files, and upload them to the server all at once. This relieves developers of the task of doing each of these steps individually which can

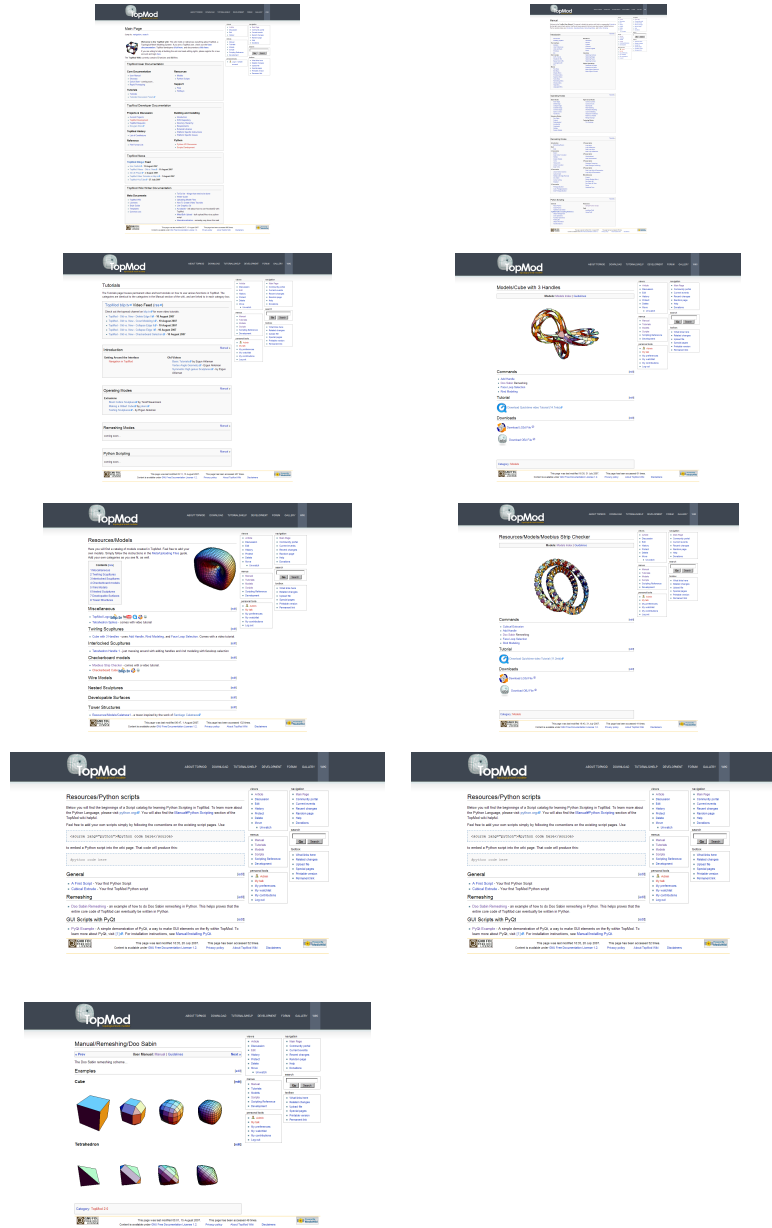


Fig. 36.: Screenshots from the TopMod Wiki main page, user manual, tutorials index, model index, script index, and a sample operation page.

Google Code **topmod** Topological Mesh Modeler [What's new?](#) [Help](#) [Sign in](#)

[Project Home](#) [Downloads](#) [Wiki](#) [Issues](#) [Source](#)

TopMod

TopMod is based on a new paradigm that allows dynamically changing the topology of 2-manifold polygonal meshes. The new paradigm always guarantees topological consistency of polygonal meshes.

Based on our paradigm, by simply adding and deleting edges, handles can be created and deleted, holes can be opened or closed, polygonal meshes can be connected or disconnected. These edge insertion and edge deletion operations are highly consistent with subdivision algorithms. In particular, these operations can easily be included into a subdivision modeling system such that the topological changes and subdivision operations can be performed alternatively during model construction.

TopMod demonstrates practical examples of topology changes based on this new paradigm and show that the new paradigm is convenient, effective, efficient, and friendly to subdivision surfaces.

Please visit the new TopMod homepage at <http://www.topmod3d.org/> to find out more and to download executable version for Windows and OS/X.

License: [GNU General Public License v2](#)

Labels: [3d](#) [mesh](#) [modeling](#) [topology](#) [opengl](#) [qt](#)

Featured Downloads: [Show all](#)

- [TopMod2 212.dmg](#)
- [TopMod2 213.zip](#)

Links: [TopMod Home](#) [TopMod Research](#) [TopMod Wiki](#) [TopMod Gallery](#) [TopMod Discussion Forum](#)

Blogs: [TopMod Blog](#)

Groups: [TopMod Developer Google Group](#)

Project owners: [Join project](#)

- [dumorris](#) [stuart.tett](#)
- [vinod.srinivasan](#)
- [ergun.akleman](#)

Project members:

- [ozgur.gonen](#) [onedanshow](#)
- [check227](#) [ddrell](#) [alex.fleming](#)
- [jacemiller4](#)

©2007 Google - [Google Home](#) - [We're Hiring](#) - [Terms of Service](#) - [Privacy Policy](#) - [Discussion Group](#)

Fig. 38.: Screenshot from the TopMod Google Code project page.

all the disparate features. The repository needed to be hosted on a server outside the Visualization Laboratory network to encourage developers outside of the lab to join the project. After evaluating a number of different solutions, including SourceForge, OpenSVN, codehaus, and Google Code, I decided to use Google Code as it provided a fully functional Subversion repository as well as a bug tracking system, a wiki, and a package release manager that tracks the number of downloads for each release. Another advantage to Google Code at this time is that it does not have distracting advertisements dispersed throughout the pages like SourceForge. The managers of Google Code also provide a few handy Python scripts for bulk uploading release files, which helps speed up the process of releasing new versions of TopMod. The TopMod Google Code repository can be viewed by visiting <http://code.google.com/p/topmod> (See Fig. 38). At the time of this writing, the various releases of TopMod have been downloaded over 11,000 times worldwide.

7. Web Site Usage Statistics

Once the new topmod3d.org user community was set up, I needed a way to figure out if anyone was actually using it, and possibly find out where they were coming from and what pages they were looking at most often on the site. To accomplish this task, I turned to yet another free and easy to use software package known as Google Analytics, formerly Urchin Software [39]. Google Analytics allows web developers to track usage statistics for web sites simply by adding a few lines of javascript code to the end of each page within the site. By using the various popular open source software packages described above, I was able to embed the necessary lines of javascript code into each page of the <http://topmod3d.org> website in about ten minutes. It required me to paste the lines of code into one template file for each application, the wiki, the forum, the blog, and the image gallery. Google Analytics also allows their service to be embedded into a Google Code project, where the TopMod subversion repository is currently hosted.

Once the proper Javascript code is embedded into each page of the site, Google Analytics begins to collect usage data as visitors come to and leave the site. It collects a wide variety of data, such as referring pages, geographic data (See Fig. 39), web browser and operating system data, as well as the duration of each visit and the page last viewed before exiting the site. At the time of this writing, the web site was only two months old and had received 90,000 pageviews and 15,000 unique visits.

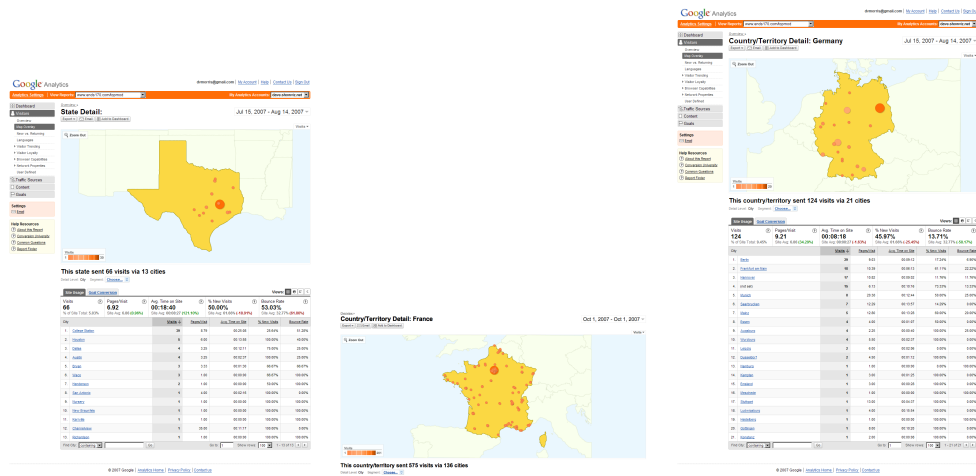


Fig. 39.: Screenshots from the Google Analytics page for topmod3d.org showing geographic usage data for Texas, France, and Germany over a two week period.

CHAPTER IV

RESULTS

As a proof of concept, all final images and animations have been created by a variety of users in different 3D rendering packages such as Blender, Maya, 3D Studio Max, and Maxwell Render, as shown in Figures 42, 45. The usability of the system was tested in a graduate level computer graphics course as well as by visitors to <http://topmod3d.org/>. Students with diverse backgrounds including art, architecture and computer science took the course, and other users who found TopMod through the TopMod web site come from a variety of industries and backgrounds including plastics engineering, architectural research, 3D modeling, computer science, and digital art. All the students and general users, regardless of their background, were able to successfully create very high genus models using a variety of input meshes. Following are some significant results that have been achieved, by using the new user interface presented in this thesis work:

- Very high genus models are created in significantly less time with the new interface (See Fig. 41) than was previously possible with the original interface (See Fig. 40).
- The enhanced sub-object selection routines allow for massive selection sets to be obtained in less time than previous methods.
- Application portability, platform independence, and internationalization allow TopMod to be used by a larger number of people worldwide.
- Keyboard shortcut customization and general interface customization and preferences options allow TopMod to be tailored to the needs and desires of the individual user and saved for future use.

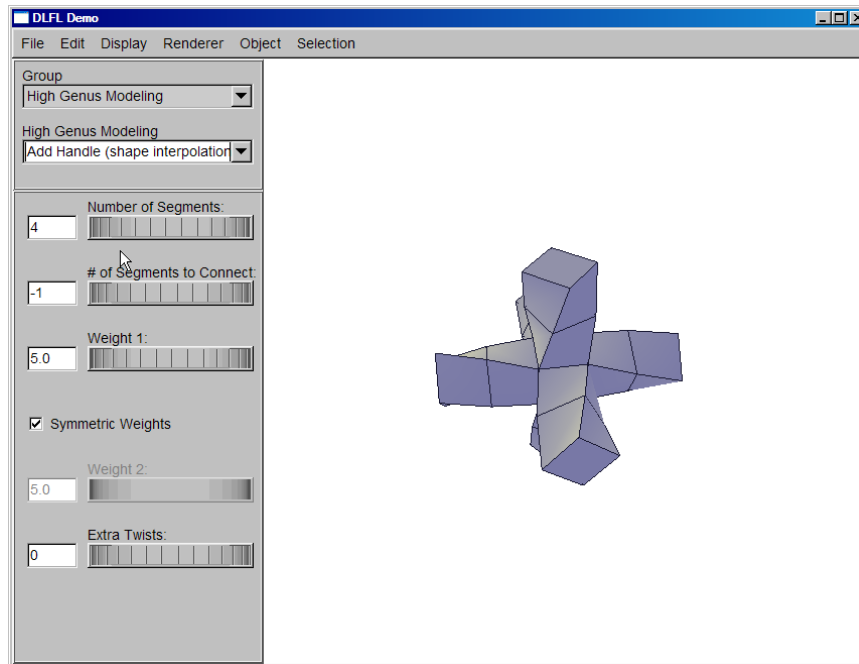


Fig. 40.: Screenshot of the old (FLTK) TopMod interface.

- The new web community provides TopMod users and developers access to the latest versions of the application and source code, and allows them to learn how to use the software and discuss problems and future developments.

It is expected upon completion that users will be able to interact with TopMod much more easily and intuitively to create complex manifold meshes in shorter amounts of time. All of the operations will be more easily accessible, and it will be much more obvious what each operation does through the in-context documentation. In particular, the Rind Modeling operation in the new version of TopMod will be much more robust. It will allow a user to select all the faces to punch out prior to executing the rind modeling operation. Previously a user would execute the command and then begin to select faces one by one, and if a mistake was made, they would have to start over without the ability to undo. With the newer face selection routines, a user has the ability to undo mistakes and select complex patterns of faces on a mesh

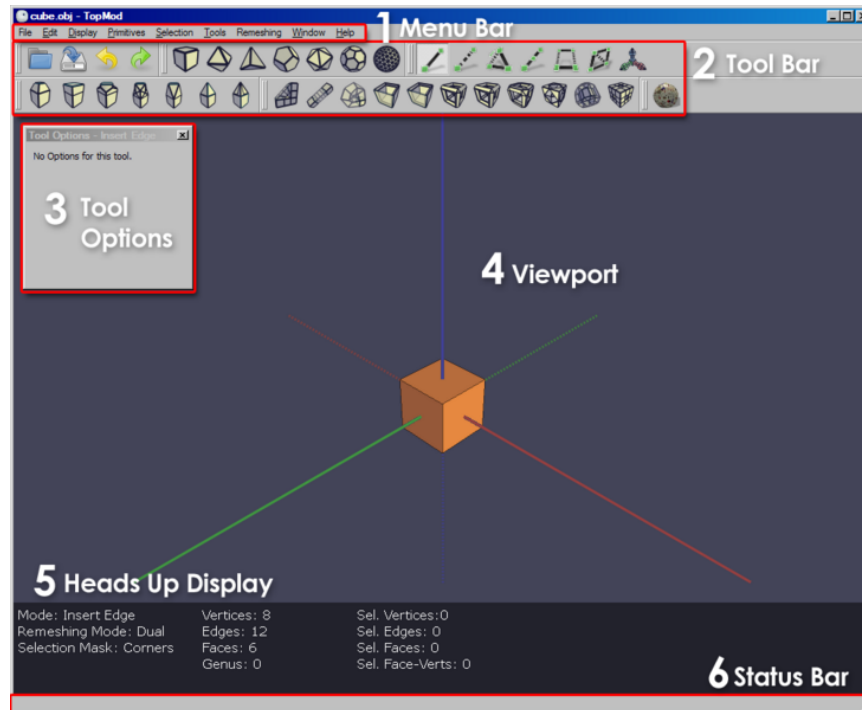


Fig. 41.: Screenshot of the new (Qt) TopMod interface.

much more easily.

One of the interesting changes to the viewport interaction techniques with regard to sub-object selection is facial extrusions. Figure 43 shows a twisting torus with 1280 faces that were selected simultaneously using the `SELECT ALL` command and extruded using the multiple stellate extrusion mode. Previously this model would have only been possible by clicking each of the 1280 faces individually, which would have taken a considerable amount of time thus discouraging the average user from even trying without knowing what the result would look like. Figure 44 was created using a combination of the new `SELECT FACE LOOP` routine along with the multiple stellate extrusion mode. Two face loops were selected as opposed to selecting over 1100 faces individually to achieve the same result.

Another interesting change to the TopMod interface is the technique or process for creating crust, or rind, models. Figure 45 was created by selecting three face loops containing more than 1000 faces and using the new crust modeling tool which punches the selected faces out of the model simultaneously. With the old version of the crust modeling tool this operation would have taken over 1000 individual clicks to remove the same faces.

With regard to the new TopMod web site, I have received a significant amount of feedback from the user community through the discussion forum and the blog, and have discovered people who have been using TopMod in a variety of environments from academic to professional. For example, Matias del Campo, a professor at the Southern California Institute of Architecture (SCI-Arc) and the University of Applied Arts (die Angewandte) in Vienna, Austria, uses TopMod with his students in his “Emergent Density” course which covers various aspects of urban design and digital fabrication. Santiago Perez, a professor at the University of Houston, also uses TopMod with his students in a digital fabrication and materials investigation course. I have included a series of images below with permission from the members of the TopMod discussion forum who created them, along with some of my own models (See Figs. 42, 46, 47, 48).

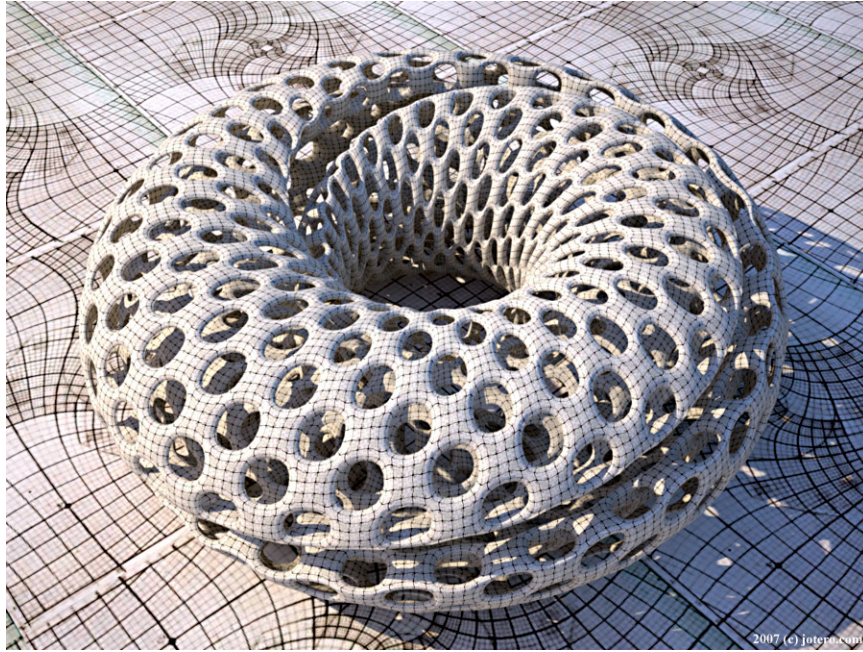


Fig. 42.: A model created using the new version of TopMod by Torolf Sauermann. ©2007 (<http://www.jotero.com>).

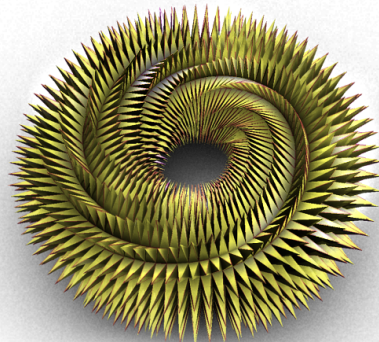


Fig. 43.: A twisting torus model with stellated extrusions on each of the 1280 faces.

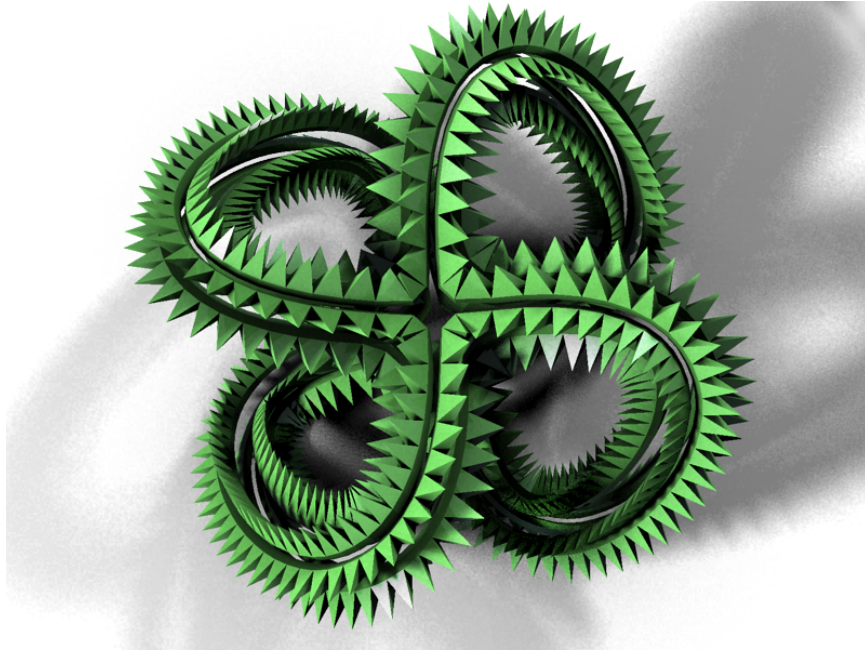


Fig. 44.: A model created by simultaneously extruding a series of face loops with over 1100 faces.

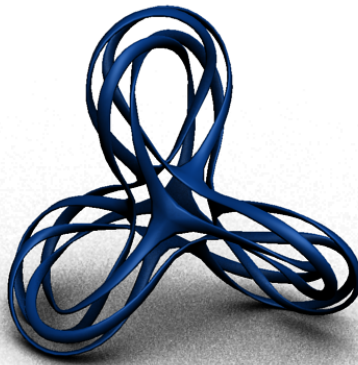


Fig. 45.: A model created by selecting three face loops containing more than 1000 faces and using the new crust modeling tool.

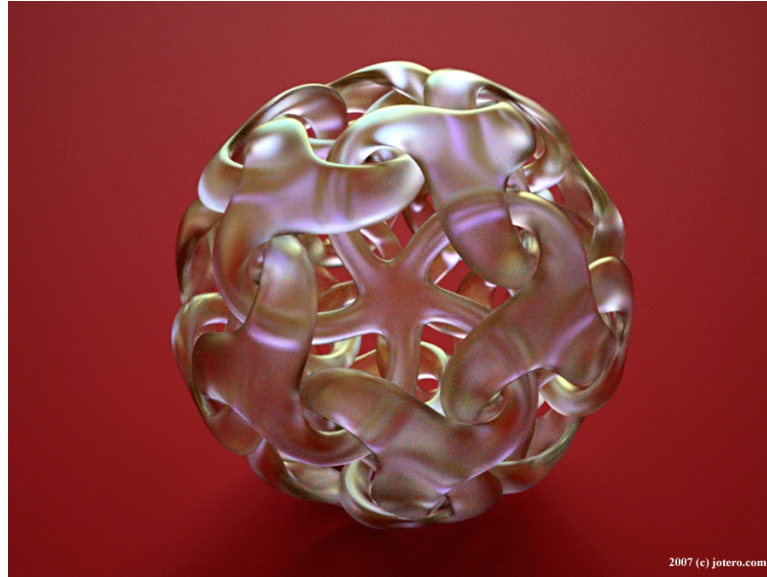


Fig. 46.: A model created by Torolf Sauermann. ©2007 <http://www.jotero.com>.

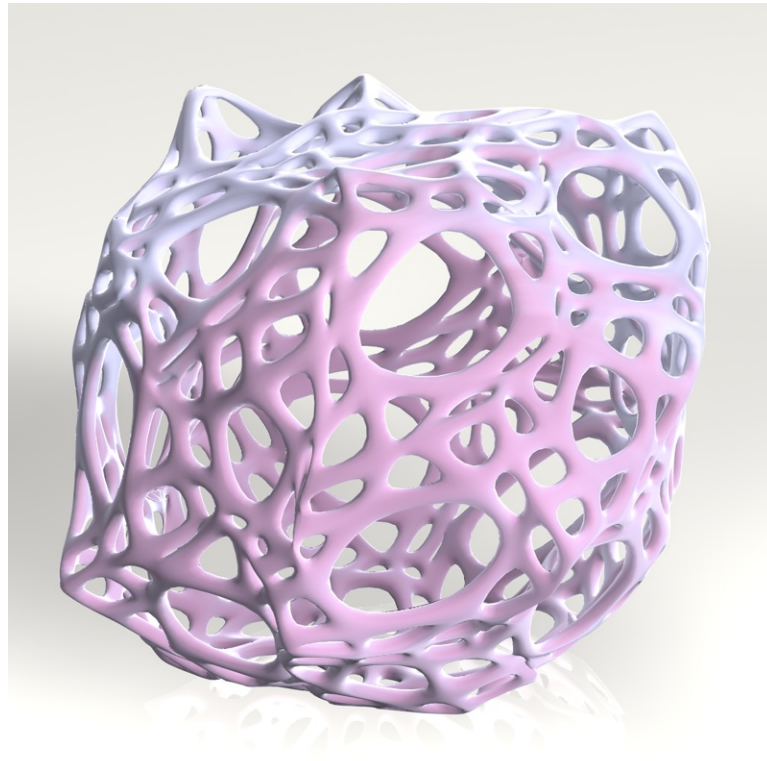


Fig. 47.: A model created using the new version and the Python scripting engine by a user from the TopMod web community.



Fig. 48.: A model created using the new version by a user from the TopMod web community.

CHAPTER V

CONCLUSION AND FUTURE WORK

With this new user interface, TopMod users can create high genus 2-manifold meshes much faster and more easily than ever before as well as customize the user interface to their personal preferences to allow for an even faster workflow.

TopMod now has an official web site (<http://topmod3d.org>) that offers a wide variety of content and collaborative systems including a discussion forum, a user and developer manual (the TopMod Wiki), a public image gallery, a source code repository, source code documentation, a news blog, and tutorial videos (<http://topmod.blip.tv/>). Users and developers alike can learn how TopMod functions on a variety of levels, share content created and modified using TopMod, and discuss issues, bugs, feature requests, and even contribute to the future development of the application.

The future work of TopMod will be in the hands of the user and developer communities created through <http://topmod3d.org/>. The discussion forum will allow for free exchange of ideas about how to make TopMod better, and future developers will be able to build of the ideas of the community at large through the web site.

Qt[®] allows developers to extend functionality in many ways, including support for Wacom Tablet device input, GPU shading with the NVIDIA Cg Toolkit, multi-threaded support, and using hash maps as opposed to linked lists for the internal data structure. This would significantly change the core of the TopMod code and the theory behind DLFL, but it would ultimately increase the speed of the internal operations by a significant amount, and would allow TopMod to handle even more complex meshes with more polygons.

Other future additions to the TopMod code that would increase extensibility and

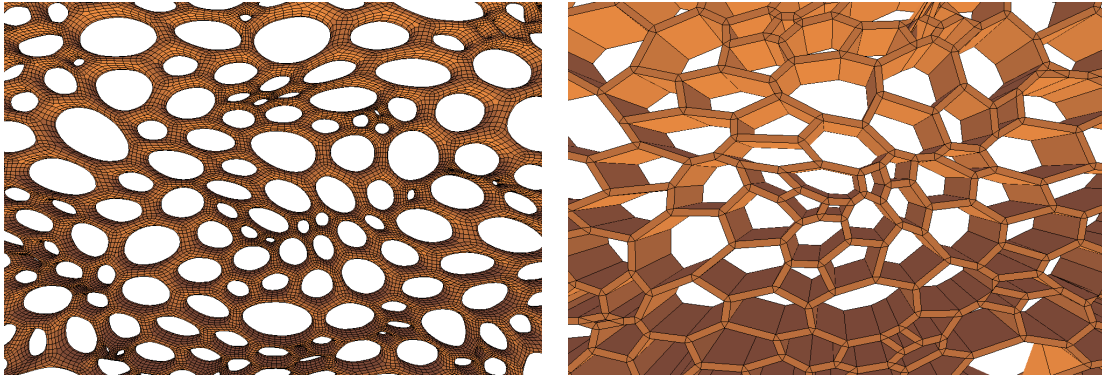


Fig. 49.: Voronoi subdivision wireframe models created with a series of Python scripts and imported into TopMod.

usability include poly-editing features such as translate, rotate and scale for multiple vertices, edges, and faces, support for multiple objects in a scene, support for a scene graph such as OpenSG or a Directed Acyclic Graph data structure, and a modifier stack. A modifier stack would make TopMod more parametric by allowing users to go back to previously performed operations and change parameters that would change the result of all operations above it in the stack. These changes will also require many changes be made to the core and auxiliary DLFL libraries.

Other additions and feature requests that have come in through the TopMod Discussion Forum include a Voronoi subdivision scheme, an option to extrude faces together, a progress bar, and a new set of menger sponge and sierpinski tetrahedra primitives (See Fig. 49).

Thanks to the thesis work of Stuart Tett, users will now be able to write Python scripts to further extend the functionality of TopMod, and even generate custom interfaces and controls with PyQt[®] for interacting with and editing 2-manifold meshes.

TopMod also needs to be modified to handle 1-valence vertices (dangling edges) better than it currently does. When a 1-valence vertex is encountered in a model and

certain remeshing operations are performed, the application will not respond and will ultimately need to be restarted. To remove this error, each function in the TopMod auxiliary library will need to be significantly modified to support 1-valence vertices.

REFERENCES

- [1] B. Saint-Moulin, “3d applications 2007 comparisons table,” http://www.tdt3d.be/articles_viewer.php?art_id=99, 2007, [Online; accessed 2-Oct-2007].
- [2] C. Ware, *Information Visualization: Perception for Design*, San Fransisco, CA, Elsevier Inc., 2004.
- [3] M. Burns, *Automated Fabrication: Improving Productivity in Manufacturing*, Englewood Cliffs, NJ, Prentice Hall, 1993.
- [4] M. Kraus, “Livegraphics3d homepage,” <http://www.vis.uni-stuttgart.de/~kraus/LiveGraphics3D/>, [Online; accessed 19-June-2007].
- [5] The Blender Foundation, “Blender wiki,” <http://wiki.blender.org/>, [Online; accessed 20-July-2007].
- [6] E. Akleman and J. Chen, “Guaranteeing 2-manifold property for meshes by using doubly linked face list,” *International Journal of Shape Modeling*, vol. 5, no. 2, pp. 149–177, December 1999.
- [7] E. Akleman, J. Chen, and V. Srinivasan, “A minimal and complete set of operators for the development of robust manifold mesh modelers,” *Graphical Models*, vol. 65, no. 5, pp. 286–304, February 2003.
- [8] E. Akleman, J. Chen, V. Srinivasan, and F. Eryoldas, “A new corner cutting scheme with tension and handle-face reconstruction,” *International Journal of Shape Modeling*, vol. 7, no. 2, pp. 111–121, December 2001.
- [9] E. Akleman and J. Chen, “Regular meshes,” *ACM Symposium on Solid Modeling and Applications, SM*, pp. 213–219, 2005.

- [10] E. Akleman and J. Chen, “Regular mesh construction algorithms using regular handles,” *Proceedings - IEEE International Conference on Shape Modeling and Applications 2006, SMI 2006*, vol. 2006, 2006.
- [11] E. Akleman and J. Chen, “Insight for practical subdivision modeling with discrete gauss-bonnet theorem,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4077 LNCS, pp. 287–298, 2006.
- [12] E. Akleman, O. Ozener, and C. Yuksel, “Designing symmetric high-genus sculptures,” *Proceedings - Bridges 2006*, 2006.
- [13] E. Akleman, J. Chen, and V. Srinivasan, “Interactive rind modeling,” *Proceedings - IEEE International Conference on Shape Modeling and Applications 2003, SMI 2003*, vol. 2003, 2003.
- [14] V. Srinivasan, E. Akleman, and J. Chen, “Interactive construction of multi-segment curved handles,” *Proceedings - Pacific Graphics 2002*, 2002.
- [15] E. Akleman, E. Mandal, and V. Srinivasan, “Soldifying frames,” *Proceedings - Bridges: Mathematical Connections in Art, Music, and Science 2004*, 2004.
- [16] E. Akleman, V. Srinivasan, Z. Melek, and P. Edmundson, “Semiregular pentagonal subdivisions,” *Proceedings - Shape Modeling International SMI 2004*, pp. 110–118, 2004.
- [17] E. Landreneau, E. Akleman, and J. Keyser, “Interactive face-replacements for modeling detailed shapes,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4077 LNCS, pp. 602–608, 2006.

- [18] E. Landreneau, E. Akleman, and V. Srinivasan, “Local mesh operators: Extrusions revisited,” *Proceedings - International Conference on Shape Modeling and Applications, SMI'05*, vol. 2005, pp. 351–356, 2005.
- [19] V. Srinivasan and E. Akleman, “Connected manifold sierpinsky polyhedra,” *ACM Symposium on Solid Modeling and Applications, SM*, pp. 261–266, 2004.
- [20] E. Akleman, V. Srinivasan, and E. Mandal, “Remeshing schemes for semi-regular tilings,” *Proceedings - International Conference on Shape Modeling and Applications, SMI'05*, vol. 2005, pp. 44–50, 2005.
- [21] E. Mandal, E. Akleman, and V. Srinivasan, “Wire modeling,” *Visual Proceedings of ACM SIGGRAPH'2003 (Siggraph Sketch)*, 2003.
- [22] M. Muller-Prove, “The core of information technology,” <http://www.cs.gmu.edu/cne/itcore/userinterface/timeline.html>, [Online; accessed 10-June-2007].
- [23] A. Fischer and K. Lee, “Vision and reality of hypertext and graphical user interfaces,” http://www.mprove.de/diplom/text/3_guis.html, [Online; accessed 10-June-2007].
- [24] J. Reimer, “A history of the gui,” <http://arstechnica.com/articles/paedia/gui.ars/>, 2005, [Online; accessed 11-June-2007].
- [25] C. Alt, “The materialities of maya: Making sense of object-orientation,” *Configurations*, vol. 10, pp. 387–422, 2002.
- [26] M. Chignell and J. Waterworth, “Wimps and nerds: an extended view of the user interface,” *SIGCHI Bull.*, vol. 23, no. 2, pp. 15–21, 1991.
- [27] R. Rowe, “Gfx: Alias wavefront maya 4,” *Linux J.*, vol. 2001, no. 90, 2001.

- [28] G. Fitzmaurice and G. Kurtenbach, “System for accessing a large number of menu items using a zoned menu bar,” <http://www.google.com/patents?id=WeEVAAAAEBAJ&dq=6915492>, 2003, [Online; accessed 10-May-2007].
- [29] T. Roosendaal and S. Selleri, *Blender 2.3 Guide: Free 3D Creation Suite for Modeling, Animation, and Rendering*, San Francisco, CA, No Starch Press, 2005.
- [30] R. Cheng, *Inside Rhinoceros 4*, Clifton Park, NY, OnWord Press, 2007.
- [31] Larry Gould, “3d modeling and rendering for the rest of us,” <http://www.autofieldguide.com/articles/090409.html>, [Online; accessed 30-August-2007].
- [32] J. Blanchette and M. Summerfield, *C++ GUI Programming with Qt 4*, Upper Saddle River, NJ, Prentice Hall PTR, 2006.
- [33] N. Bornstein and E. Dumbill, *Mono: A Developer’s Handbook*, Sebastopol, CA, O’Reilly, 2004.
- [34] J. Smart, K. Hock, and S. Csomor, *Cross-Platform GUI Programming with wxWidgets*, Upper Saddle River, NJ, Prentice Hall, 2005.
- [35] B. Hook, *Write Portable Code: An Introduction to Developing Software for Multiple Platforms*, San Francisco, CA, No Starch Press, 2005.
- [36] Trolltech, “About trolltech,” <http://trolltech.com/company/about>, [Online; accessed 8-June-2007].
- [37] Trolltech, “Qt reference documentation (open source edition),” <http://doc.trolltech.com/4.3>, [Online; accessed 8-June-2007].

- [38] The Blender Foundation, “The blender documentation project,” <http://www.blender.org/download/documentation/>, [Online; accessed 20-July-2007].
- [39] Google, “Urchin 5 web analytics software,” http://www.google.com/analytics/urchin_software.html, [Online; accessed 27-July-2007].

VITA

David Victor Morris

Dept. of Architecture

c/o Dr. Ergun Akleman

3137 TAMU

College Station, Texas 77843-3137

dvmorris@gmail.com

Education

M.S. in Visualization Sciences, Texas A&M University, May 2008

B.E.D., Texas A&M University, May 2005

Interests

Web Design/Programming

Social Media

Semantics

3D Computer Graphics/Parametric Modeling

The typist for this thesis was David Morris.