

INSTITUT FÜR INFORMATIK
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



Masterarbeit

Simple 3D Low Poly Modeling Tool with Intuitive GUI

Yueyue Cheng



Masterarbeit

Simple 3D Low Poly Modeling Tool with Intuitive GUI

Yueyue Cheng

Aufgabensteller: Prof. Dr. Heinrich Hußmann

Betreuer: Dr.-Ing. Wolfgang Höhl
Henri Palleis

Abgabetermin: 15. Juni 2015

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. Juni 2015

.....
(Unterschrift des Kandidaten)

Abstract

This paper deals with the design and development of a small 3D modeling tool which is mainly applied on the PC. At first beyond a short overview of the application of 3d model and the history of 3d modeling software development, it explains the present situation of the modeling software on the market, as well as deficiencies that should be improved. Then the relevant theoretical knowledge is introduced and several modeling software on the market are used and discussed through the aspect of game design to guide the design and development of the modeling tool. This tool which is required to be with an user-friendly user interface and convenient operations is inspired by the famous sandbox game “Minecraft”, and developed by Qt Creator with OpenGL. This tool provides a homemade cube structure as the basic unit of modeling, and uses the Catmull-Clark Subdivision Surface and its variants to optimize the edge of the model.

After the tool is complete, it will be applied by actual example and its strengths and weaknesses will be evaluated. Considering all this, it will show some strategies to solve the weaknesses of this tool. Finally, the development direction of this tool in the future will be prospected and described.

Keyword: 3D Modeling, Catmull-Clark Subdivision Surface, Low Poly, Intuitive GUI, Qt Creator

Contents

1. Introduction	1
1.1. Background and Motivation	1
1.2. State of the Art	1
1.3. Overview of the Chapters	3
2. Theory and Technique	5
2.1. Low Poly	5
2.2. Game Design	6
2.3. Relevant File Format	7
3. Environment and Analysis	9
3.1. Development Environment	9
3.2. Existing Modeling Software Analysis	10
3.2.1. the Basic Features	10
3.2.2. Workflow of Modelling	13
3.2.3. Use Case for Game Design	14
4. Concept and Design	21
4.1. Requirements Analysis	21
4.2. Workflow	22
4.2.1. Ready-made Mesh	22
4.2.2. Catmull-Clark Subdivision Surface	22
4.2.3. the Pre-processing	25
4.2.4. the Post-processing	28
4.3. Modeling Functions	29
4.4. User Interface	35
5. Programming and Implementation	39
5.1. Cube Structure	39
5.2. Algorithm of Catmull-Clark Subdivision Surface	40
5.3. 3D Modeling File Format	43
6. Discussion and Conclusion	47
6.1. Practical Use Case	47
6.1.1. the Description of Modeling Process	47
6.1.2. Model Export	50
6.2. Lessons Learned	51
6.2.1. Performance Evaluation	51
6.2.2. Good Points	53
6.2.3. Problems and Solutions	54

Contents

6.3. Future Work	59
A. Version and Configuration of System and Related Software	61
B. Contents of CD	63
List of Figures	65
Bibliography	67

1. Introduction

1.1. Background and Motivation

3D modeling is the process to produce 3D model which represents objects by three-dimensional polygons and usually displayed with computer or other video devices. The displayed 3D model can be anything that can be existed physically in nature, either the entity in the real world or something in imaginary world. Today, 3D modeling is widely used in various fields. For instance, the medical industry uses 3D modeling to produce the accurate models of the organs; the engineering and manufacturing uses it to design new products; the the construction industry uses it to show simulated buildings and landscape; the film and video industry uses it to create animated characters and simulate their actions; the video games uses it to obtain creative resources. With the continuous development of 3D computer graphics technology, and popular of 3D graphics on the PC, more and more individuals are getting started with 3D modeling, because of the needs for study and job, or just as a hobby. In this way, appropriate software of 3D computer graphics is necessary, and very important.

The 3D computer graphics software is the software which produces 3D models or CGI (Computer-Generated Imagery) through 3D modeling and rendering. These kinds of software are often used in the production of movies and video games, most of them also include high-end dedicated plug-ins for construction and other fields. On the other hand, this type of software often has a complex user interface and needs to add additional software kit while it is used. Sometimes it even requires the user to have some relevant skills and knowledge.

The objective of this project is to design and develop a small 3D modeling software which is mainly applied on the PC. This software is inspired by the famous sandbox game “Minecraft”, developed by Qt Creator with OpenGL, to design more user-friendly user interface and operations, so that users can make model quickly and easily.

1.2. State of the Art

The 3D computer graphics software allows users to create models and to change mesh of model by adding, subtracting, stretching or other ways to the shape which the users want. Meanwhile, the software like this can also display the model from a variety of angles, and usually its view can be zoomed, rotated or translated. Most modeling software support a number of different model formats, so that they can export their models which can be imported into other applications for further processing. In addition, some software also include a number of related features, such as texture mapping facilities, NURBS (Non-uniform rational B-spline) and even animation which can generate video with a series of rendered scenes.

1. Introduction

Currently the most popular 3D computer graphics software in the world include 3ds Max, cinema4D, Maya etc[1]. The 3ds Max which is formerly known as 3D Studio was originally released in 1990 for the DOS platform[2]. At that time it is still far away from 3D computer graphics software on the PC. However, Autodesk's 3D Studio showed its powerful features and stability. This is also one of the reasons that it can be widely popular. Before the emergence of Windows NT, the production of industrial-grade computer graphics is monopolized by SGI (Silicon Graphics, Inc.) graphics workstations. Later because of the 3D Studio MAX 1.0, Maya and other similar software for Windows NT platform have been launched. The combination of 3D Studio Max and Windows NT reduced the costs of CG production significantly. After that, the CG software production platforms have been moved from the Unix workstation to Web-based PC.

Most of the PC-based high-end 3D computer graphics software, including the three examples above are commercial software. They support for 3D rendering, are integrated 3D graphics software for making 3D models, images, animations and games, and request relatively low configuration. For example of 3ds Max, it began to use at first the animation in computer games, further became to product also special effects of movies, such as "X-Men 2", "The Last Samurai" and so on[3]. 3ds Max has strong modeling capabilities, not only include regular polygon modeling which is especially used on game design, but also support the NURBS to smooth surface and to eliminate the straight edges of polygon models. The auxiliary tools in this software are very rich[4]. At the same time 3ds Max can also provides additional functions and improves the ability of the original by installing plug-ins. The built-in scripting language MAXScript is be supported to allow users to develop new tools and create more plug-in modules themselves.

Of course, there are still many deficiencies of current modeling software. Taking 3ds Max as an example once again, due to the abundant auxiliary tools and plug-ins, users can deal the same question with several different options. This is its strength but also its weakness, especially for beginners. And because it is commercial software, a lot of plug-ins is not for free. Some high-end special plug-ins which have high price, only will be used by specialized production companies. In order to use these diverse functions and tools, the whole software including the user interface becomes complex. Although some modeling software like Autodesk's FormIt and Google's SketchUp offer a friendly user interface and simple operations already, they had to give up some features, and do not provide advanced modeling tools such as NURBS or subdivision surfaces. In addition, building a good 3D model often requires the user's own ability: a certain art foundation, a good sense of space, and sometimes it even requires the user to have special knowledge of certain aspects. In this case, even the top-end modeling software today wouldn't help very much.

This master project will provide a cube as the basic unit of modeling which can be transformed and smoothed instead of the modeling unit (point, edge and face) in traditional modeling software, and probe a solution to make the 3D models easier, faster and more human-friendly by combining these basic units.

1.3. Overview of the Chapters

In this project, several modeling software on the market will be compared and analyzed, and a new simple modeling software will be developed. Then we will build new models by this software and export the models to use further for game design and rapid prototyping. This entire process, including test results will be discussed and analyzed.

In chapter 2 the relevant background knowledge will be described, including the theory and technology related to this project, for example: polygonal modeling, game design and some common file formats of 3d model.

In chapter 3 we will focus on comparing and testing several existing modeling software (Blender, Wings3D and SketchUp) through the aspect game design, and then discuss and analyze the results.

In chapter 4 we will explain the design concept of the new modeling software in this project, the expected functions and the actual functions of success at the last. Especially the algorithm and principle of subdivision surface that be involved in the software will be introduced in detail.

In chapter 5 some implementations of the major and important functions in this modeling software will be described and the programs will be analyzed through the class diagram and the sequence diagram. According to some examples those functions will be used and tested.

Finally in chapter 6 we will make a use case for this software, record and analyze the results, determine strengths and weaknesses, find out the improvement for the problems. And the entire project will be reviewed and summarized; the experience will be learned.

2. Theory and Technique

The topic of this project thesis is “Simple 3D low poly modeling tool with intuitive GUI”. This chapter will introduce and describe the theories and techniques which are involved to 3D computer graphics technology in this project

2.1. Low Poly

3D model is broadly divided into two categories, one is a solid, and the other is shell. The solid model which is more realistic, but also more difficult to build, is mainly used for the CAD (Computer Aided Design) and non visual simulations such as engineering simulations and medical[5]. The model of shell is a polyhedral object like a thin eggshell which represents the object’s boundary but not volume. It is common in computer graphics, because the object’s appearance depends largely on its boundary. And the model’s shape is defined with a polygon mesh which is a collection of vertices, edges and faces[6]. Low poly means a polygon mesh with relatively few polygons. The alternative is high poly which has relatively more polygons in a polygon mesh.

In theory, the polygon mesh can be constituted by the polygons which have any number of edges, but in fact the mesh is commonly divided into triangles for display. Generally, the more triangles in polygon mesh make the shape of model more detailed, but also take more rendering time. The number of triangles that can be rendered to appear in a scene and the frame rate of rendering that can be accepted, are depend on the combination of rendering method and the performance of computer. In order to decrease rendering times or to increase frame rate, it is necessary to reduce the number of triangles in the scene. This is why low poly is needed for a polygon mesh[7].

Therefore the low-poly models are especially useful in computer games, films or other software in which the user have to manipulate 3D objects in real time. Rendering in real time requires higher frame rates. The processing power of personal computer or games console is very limited. A large network of computers also must take hours to render films or still images with a higher polygon, although the rendering does not need to be done in real time[8]. In a case such as this, the low poly modeling sacrifices some detail of model in favor of precious computational cycles. In this way, with only less memory, the game can be rendered quickly, even be playback and give best game effects to players.

The common models of low poly are often blocky and lacking in detail, for example rectangular fingers or no individual fingers of a hand. When the model represents the object which is supposed to be circular or spherical, it needs high number of triangles to make a curve to appear smooth. In addition, low poly is a relative term. There is no threshold to define a low poly mesh or a high poly mesh. The low poly depends on the shape of meshes, the required detail and the performance of hardware. With the improvement of computer

power, the number of polygons can also be increased.

2.2. Game Design

In this project, whether the analysis of the existing modeling software on the market in preparatory work, or the testing after completion of the new modeling software, will use a simple self-designed video game as a use case. So what is game design? Game design is a process of designing the content and rules of a game[9], involving every detail for how would the game be played. Here it mainly refers to video games: the rules and playing of the game, the effect of players' choice for the game process, the standard of winning or losing, the background story, the game programming, and the visual art and sound in order to make a better user experience. Finally, there may be also the problems of the transition.

Essentially, the game design is restrained by a series of constraints which guide the development of the game exactly. These constraints are designed depending on the different types of game. There are many kinds of constraints, such as: funds and technique for game development, product copyright and profit ability, the constraints from user groups and political constraints.

Early in the video game, the game designer is often programmers. In 1970s, with the expanding of the video game industry, game designers' task became independent of the main programmer. For some very complex games, such as MMORPG (Massive Multiplayer Online Role Playing Game) game, there are usually one or two major designers and many junior designers for subsets of the game.

Depending on the different game designers and companies, the process of game design will have different principles and procedures[10], and according to the operating post of game designer, the working content is also different. In general, the game designers will pre-create a game design document which determines the direction and related details of the designed game. This document will probably base on the existing games or concept and be continually improved with the process of game development[11]. With the help of pre-existing engines and tools like Game Maker or Unity3D, concept prototyping can be implemented more easily. Besides, other works of game designers include: writing game scripts and requirements for art resource, making game scenes, participating in the game testing and rewrite the game engine if it is necessary.

Before the start of a new project, which is in the discussion stage, the game designers may be required to submit a short concept at first, including the game's specific features, audience, platform and so on. According to this concept of short, designers will consult with other relevant departments, and give the available time, the size and the budget to determine its feasibility and then to decide whether the project be implemented. With the development of production, there may be more and more problem about elements missing from the design. How to deal with these issues needs the game designer to make decisions. The consequences of these decisions are difficult to predict, and the merits of some decisions often may appear only after complete implementing. These are known as design's unknowns, the sooner they are found the less risk will be brought to the entire process of game development and even the game itself.

2.3. Relevant File Format

Blender and other 3D modeling software often support several file formats. By using the software, we come inevitably into contact with these file formats.

- OBJ

OBJ(.obj) is a standard 3d model file format first developed by Wavefront Technologies and used for the 3d modeling and animation software “Advanced Visualizer”. The file format is a text based, open and universally accepted format by other 3D graphics applications. Therefore, in this project, we also choose this format, as one of the export format for the models of our new modeling software. Except ASCII format (.obj), Object files can be also binary format (.mod) which is unpublished as a patent[12].

OBJ3.0 file format supports lines, polygons, surfaces and free-form curves. Lines and polygons are described by their points. Curves and surfaces are according to their extra information which are controlled by points and are attached the type of curve. This information supports the regular and irregular curves, including Bezier, B-spline, Cardinal/Catmull-Rom and Taylor equations.

OBJ file consists of lines of text, does not require any kind of file header, although there is often a few lines of comment with the symbol “#” as the beginning of the file. The meaningful lines are marked by one or two letters (the keyword) at the beginning of the line. The keywords can explain what kind of data in this line. For instance vertex definition which defines the coordinates of each vertex, and face definition which defines the vertices in each face of the model, are respectively beginning with the keyword “v” and “f”. Other common keywords are as well as “l” for edge definition, “vt” for texture vertices, “vn” for vertex normals and so on[13]. Further discussion and examples of OBJ file’s format will continue in subsequent chapters.

OBJ file does not contain information about the properties of material. The file format Material Template Library(.mtl) is a companion file format for OBJ that describes material properties of objects. MTL is defined by Wavefront Technologies for ASCII files. A OBJ file can also reference plural MTL files with different material descriptions by name[14].

- FBX

FBX(.fbx) was file format of 3D character animation software FilmBoX which is renamed Motionbuilder later. This file format is developed by Kaydara, and now owned by Autodesk as a proprietary file format. The FBX format specifies both ASCII and binary representations. The Blender Foundation publishes an unofficial spec for the FBX binary file format[15].

The biggest use of FBX is interoperability of model, texture and camera information between digital content creation applications such as software Max, Maya and Soft-image. So the user can take the advantages of different software. It can be said, the performance of FBX in this part is quite ideal for transferring models. The transferred model is basically perfectly.

A C++ FBX SDK provided by Autodesk can read, write, and convert to or from FBX

2. Theory and Technique

files. Although the FBX file format is proprietary, its description can be founded in the FBX Extensions SDK that provides header files for the FBX readers and writers[16].

- 3DS

3DS(.3ds) is one of the 3D file formats used by the Autodesk 3ds Max for modeling, animation and so on. Other 3D file formats is MAX(.max).

3DS was the native file format of the old 3D Studio in DOS, which is the predecessor of 3D Studio Max. Since 1990 3DS has been in existence, and today it has become a standard for transferring models between 3D programs[17]. This format can save the information of mesh model and UV stably, is also compatible with basic bones and basic animation for all of 3D software. 3DS is a binary file format.

- DAE

DAS also known as COLLADA is an interchange file format(.dae) for interactive 3D software, and is managed by the Khronos Group which is a nonprofit technology consortium. As a publicly available specification COLLADA has been adopted by ISO with number ISO/PAS 17506[18].

The digital assets that are described by COLLADA documents are XML files. COLLADA defines an open standard XML-based schema to exchange freely digital assets among various graphics software without loss of information, and combine multiple software packages into extremely powerful tool chains.

Besides, COLLADA can be imported by Adobe Flash. The model of Google Earth is COLLADA.

- STL

STL(STereoLithography) is an interface protocol developed by 3D SYSTEMS company in 1987[19]. It is 3D graphics file format(.stl) and widely used for rapid prototyping.

STL files describe only the surface geometry of a 3D object which consists of a plurality of triangular facets without color, texture or other attributes. The definition of each triangular is including the 3D coordinates of each point in triangle and the normal vector of the triangle surface. The order of the triangle vertices follows the right-hand rule. The STL can be represented as either binary or ASCII data, and more compact binary files are more common than ASCII[20].

In order to properly print 3D model by STL, surface model must be closed and connected, which means that each edge must belong to the two triangles at the same time, and the both triangles do not intersect each other. But the syntax of STL does not enforce this property, so that depending on the application and software, sometimes the closure property can be ignored.

3. Environment and Analysis

In this chapter, we will complete the preparatory work before to design and to program the target modeling tools, including the confirmation of the work environment and the analysis of several existing modeling software through a case study of game design. The results of analysis will guide us better, more specifically to design and to program the target modeling tools.

3.1. Development Environment

All works in this project, including the using of existing modeling software, the design of the sample game and the ultimate goal—the programming of modeling tools, will carry out on a PC (Lenovo ThinkPad E520). The selected existing modeling software are Blender, SketchUp and Wings3D and the selected game engine which is used for case study is Unity3D (see the Appendix A).

In addition, the target modeling tool will developed by Qt Creator. Qt Creator is a cross-platform integrated development environment for Microsoft Windows, GNU/Linux and Mac OS X, which supports C++, JavaScript and QML[21]. It includes an editor of source code with syntax highlighting and auto-completion, a visual debugger, GUI-Designer, project manager and rich sample programs. On Linux Qt Creator uses the C++ compiler of the GNU Compiler Collection, while on Windows it can use the default install MinGW or MSVC for compiling. Here MSVC has been used in this project.

After years of development, Qt not only has the improved graphics library of C++, but also integrates gradually the database, multimedia library, OpenGL library, network, script library and so on in recent years. Its core library also joined the inter-process communication, multithreading and other modules, which all of these has greatly enriched development ability of Qt for large complex cross-platform applications.

The user interface of Qt graphical is based QWidget, which includes all the types of Qt GUI components such as buttons, labels, toolbars and so on. Widget is responsible for receiving the events from mouse, keyboard or window system, and depicts itself on the screen. Each GUI component is a widget, and widget can also be used as a container which contains other widgets. The positions of widgets in the user interface of an application are described by the class of layout manager. When the window is scaling, the layout manager will automatically adjust the size, position or word size of widgets, to ensure their relative arrangement and whole user interface still usability[22].

In this project, in order to complete various functions of the 3D modeling tools, the technology about 3D computer graphics is needed. The QtOpenGL module makes it easy to use OpenGL which is a standard API for rendering 3D graphics in Qt applications. The

3. Environment and Analysis

OpenGL widget class that was provided can be used just like any other Qt widget. Moreover, it will open an OpenGL display buffer where the users can use the OpenGL API to render the contents.

In the aspects of communication between objects, Qt uses the mechanism of signal/slot to replace the traditional callback[23]. When the operation event of an object happened, the object trigger will send a signal to the corresponding slot. The slot is a function which accepts the signal and then runs its own actions. Between signals and slots, the static method “connect” of QObject is used for link. The signal can be transmitted on any operating point, and even can transmit new signals in the corresponding slot. The links between signals and slots are not limited as one-on-one. A signal can link to multiple slots, meanwhile multiple signals can link the same slot, and the signals can also link other signals. The previous callback lacks safety of type, because it cannot determine the correct type of transmitted parameter. But the type of transmitted parameter between signal and slot must be consistent, otherwise the compiler will warn.

3.2. Existing Modeling Software Analysis

The Analysis of existing modeling software will be accomplished through the introduction of basic features and practical modeling. The finished models will be actually applied in a homemade video game to test the performance of the models. Moreover a typical modeling process of a simple model will be also recorded and described.

3.2.1. the Basic Features

Blender

Blender is a widely known open source, 3D computer graphics software which has BSD (Berkeley Software Distribution license) license. The functions that Blender provides including animation, modeling, rendering and materials can compare with the high-end 3D modeling software such as Maya and 3ds Max. Blender can run on different platforms (Windows, Mac OS X, and Linux)[24], and take little space after installation. Its main features are:

- Support for polygon meshes, subdivision surfaces, NURBS surfaces, Bezier curves and other a variety of geometric primitives
- Internal render engine and integrated external render engines through plugins, such as YafRay (Yet Another Free Raytracer)
- Animation tools including keyframe, timeline, (skeletal) armature, inverse kinematics, etc.
- Soft body dynamics including mesh collision detection and a particle system with collision detection
- Python scripting for game creation, task automation, custom tools, etc[25].
- Sub-project the Blender Game Engine for the production of real-time computer games

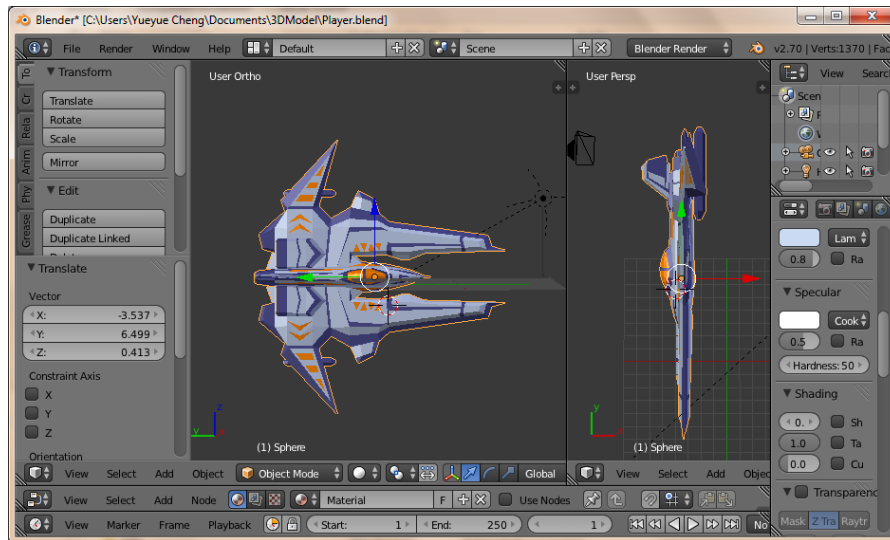


Figure 3.1.: the Screenshot of Blender with two Modeling Windows

Blender provides a lot of tools, so that the user interface is relatively complex for beginners, but it will be able to quickly get started with the help of documents. In general, it consists mainly of two basic modes: object mode which can be used to move, rotate and scale entire polygon meshes, and edit mode which can be used to manipulate the individual vertices, edges or faces of a single mesh. A large number of hot keys are used, and cooperated with menu to increase efficiency. Meanwhile the workspace of Blender can adjust itself, such as change the size of each window, increase or decrease different windows to display all kinds of information at the same time(see the Figure3.1).

In Blender, the object and the data are separated, so Blender can makes modeling quickly. All the scenes, objects, materials, sound, pictures, special effects in post-production can be integrated to the animation file “.blend” which Blender provides.

Wings3D

Wings3D is an open source 3D computer graphics software, especially good at to create subdivision surface models. Wings3D can create low to mid-range polygon models and textures, but does not support animation. Although Wings3D can export to external rendering like POV-Ray and YafRay, it has only basic OpenGL rendering itself. However, Wings3D is often used in combination with other software, for example to export the model which is created in Wings3D to Blender for more specialized rendering and animation.

Wings3D uses the context-sensitive menu rather than a highly graphic, icon-oriented interface(see the Figure3.2). In the Wings3D in modeling there are four different selection modes: vertex, edge, face and body and each mode has own modeling tools. Users will use the mouse and keyboard to affect their model by selecting and applying modeling tools of different selection modes. Other features of Wings3D are[26]:

- Support for multilateral graphics (ngon, pentagonal or more)

3. Environment and Analysis

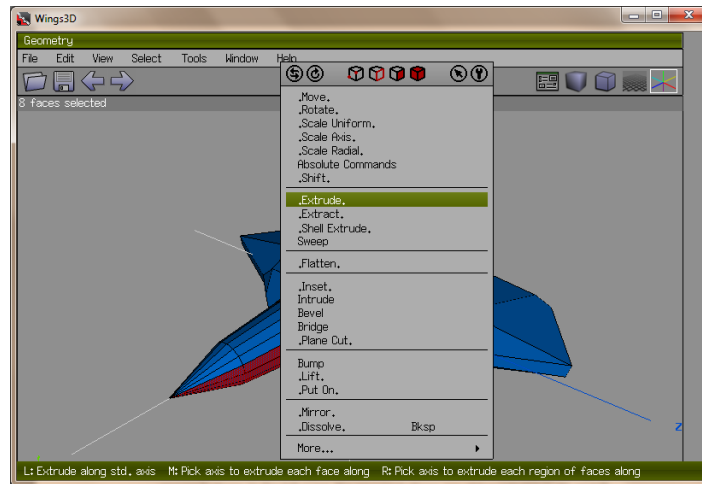


Figure 3.2.: the Screenshot of Wings3D with Context-Sensitive Menu

- Customizable interface and shortcuts
- Support for Magnets and Vector Operations about modeling tool
- Quick adjustments of a mesh by Tweak Mode
- Add and edit lighting, materials, textures and vertex colors, and automatic UV Mapping

SketchUp

SketchUp is a professional 3D modeling computer program for architects, urbanists, filmmakers, video game designers and other relevant specialized staffs. There are two versions of SketchUp, one is freeware version SketchUp Make and the other is paid version SketchUp Pro. SketchUp has a user-friendly interfaces so that it is flexible and easy to use and more intuitive than other 3D CAD program[27]. Some of the key features of SketchUp are[28]:

- Intelligent cursor system allows users to depict 3D parts by mouse in 2D screen.
- “Push/Pull” technology: grab a 2D surface and then push or pull to make it into a 3D object. Google stressed in tutorial that this technology is the most important feature in Google SketchUp.
- Simulation for the movement of the camera and the sun
- Coordination functions with Google Earth: modeling in Google Earth is also very convenient.
- Rotate: use protractor to rotate specific geometric components and to determine the Angle of rotation
- Move: move (or copy) selected object along the axes or the composite direction

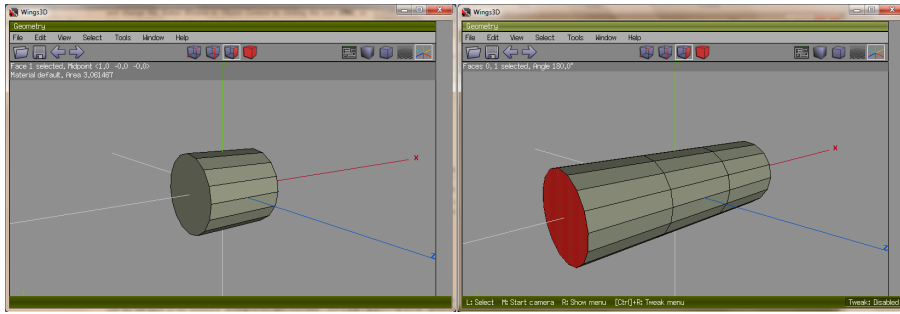


Figure 3.3.: the Cylinder (left) and the Extruded Cylinder (right)

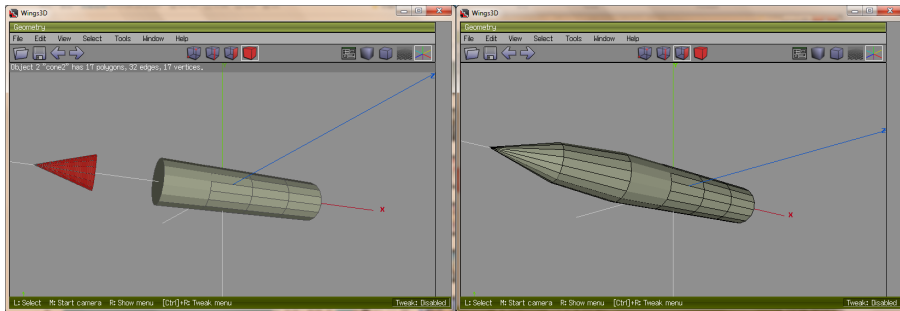


Figure 3.4.: the new Cone (left) and the Connected Mesh (right)

3.2.2. Workflow of Modelling

Although each modeling software has its own features, the basic concept and workflow of modeling are similar. Here a simple model of the spaceship was built in Wings3D as an example and the process will be briefly described. First all the three software provide a certain amount of ready-made mesh which are some simple and basic geometry. It is very helpful for modeling, when these meshes are better used. The user can select a ready-made mesh which is close to the desired shape to use directly, or to modify on its basis, in order to improve the efficiency of modeling, for example, here a simple cylinder is selected as the basic of spaceship hull. Both Blender and Wings3D provide cylinder directly. Although SkechUp provides not cylinder directly, it can change circle which it provides to cylinder easily “Push/Pull” technology.

Then the cylinder will be modified to achieve to desired shape, that means to operate the point, edge or face in the mesh, such as moving, scaling or rotation. Next, taking the cylinder as the starting, use can increase its volume and modify its shape at the same time to continue modeling. For increasing the volume of mesh, it can use “Extrude” or other similar functions to extend mesh toward the desired direction (see the Figure3.3), or can also add a new ready-made mesh and connect the both, for example to add a cone as the head of spaceship (see the Figure3.4). Sometimes to modify the details of the model can split an existing line or face to increase the operated object in order to achieve more detailed results (see the Figure3.5).

Modeling software provide the color and material function, and some can also create their own new materials. The color and material can be defined in the end after the model

3. Environment and Analysis

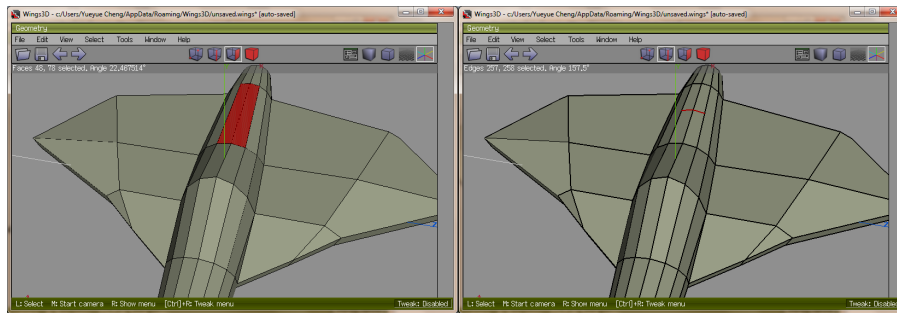


Figure 3.5.: Split two Faces (left) into four (right)

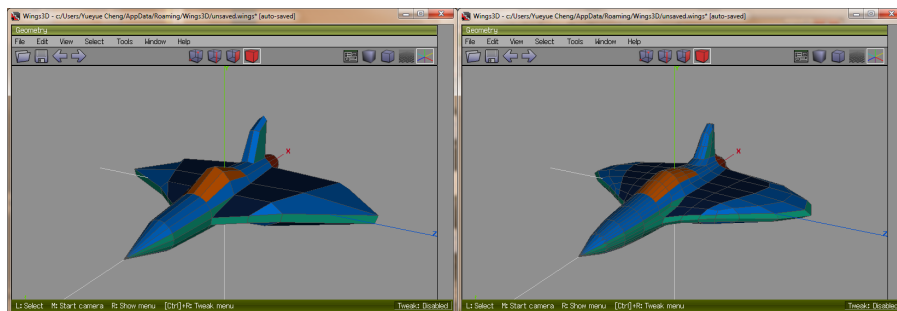


Figure 3.6.: the Smoothed Model (right)

is complete, or while the modeling process. At last the model can also choose some post-processing for optimizing, such as smooth (see the Figure3.6).

3.2.3. Use Case for Game Design

About game design, first we design and program a simple game by game engine Unity while the role models which are needed in game are modeled respectively by the three above modeling software. Then the models are exported in different formats that are supported by the modeling software, and imported to the game in Unity. Afterwards, by observing and comparing the models in different formats from different software, the various encountered problems in the process and the solutions are summarized. Final the purpose of comparison and analysis of various software is reached.

Unity3D

Unity3D is a cross-platform game engine for almost every platform[29] which is used to develop 3D video games, architectural visualization, real-time 3D animation and other types of interactive content for PC, mobile devices and websites.

Unity3D has a hierarchical integrated development environment, visualized and detailed property editor and dynamic game previews. The graphics engine of Unity3D uses Direct3D (on Windows), OpenGL (on both Mac and Windows) and proprietary APIs (on Wii); the sound system which is based on OpenAL library can play compressed audio of Ogg Vorbis;



Figure 3.7.: the screenshot of game Space Shooter

the video system is using the Theora encoder. The scripting of game engine is built on Mono which is an open-source implementation of the .NET Framework, so that uses can program with JavaScript, C# or Boo[30].

Unity3D has a variety of features, for example, it can automatically import and update the resources of project, has a built-in terrain editor to support the patches of trees and vegetation. In addition Unity3D provide a shader with multiple variants to adapt to diversity platforms[31], and multiple techniques of Texture mapping, like bump mapping, reflection mapping and parallax mapping.

Example Game Description

First, the game is called “Space Shooter” and is a vertical scroller Shoot’em up in which the action is viewed from above while the background scrolls down. There are three levels in this game and with increasing the level, the game will be more difficult. Player will have to control a spaceship to fly in the space and to protect himself from random asteroids or the spaceships of enemy, as shown in the following screenshot(see the Figure3.7).

The spaceship can move freely within the range of the screen, forward, backward, to the left or to the right while pressing arrow keys and fire while pressing space bar. Message of game status will be displayed in the upper left of the screen. Asteroids and enemies appear from the top of the screen, and move down. The enemy will also fire at you. Initially, the player has three lives. Lives will increase by through each level and will decrease by impact or attack from the enemy. The player loses the game, when his lives reduce to zero.

On the other side the Player will increase the score by shooting down asteroids and the spaceships of enemy. When the score reaches a certain value, Player can improve the level (1000 points to improve to level 2; 2000 points to improve to level 3). In the final level, the third level, the boss of enemy will appear. The boss has 10 lives and two forts which take turns to fire with several bullets at the same time. The player will win the game after shooting down the boss.

3. Environment and Analysis

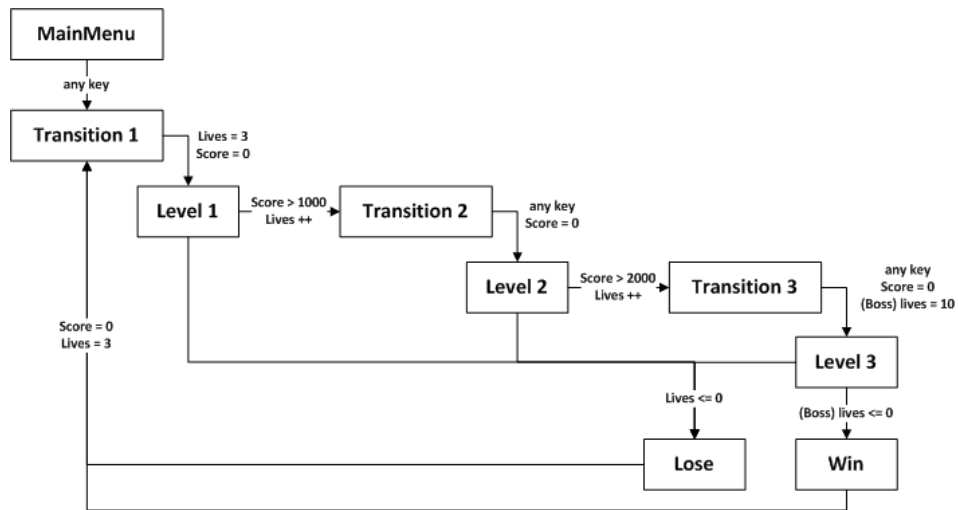


Figure 3.8.: the process of game Space Shooter

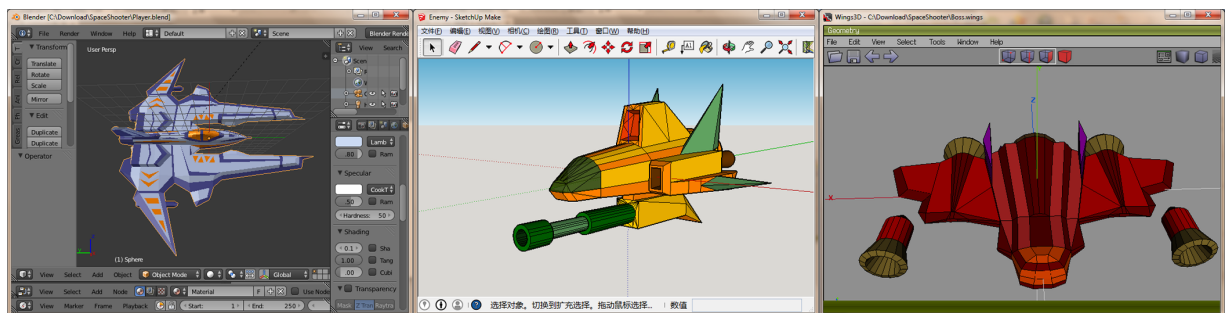


Figure 3.9.: the models of Player, Enemy and Boss

There are nine scenes in the game, including a start scene (main), three game scenes for different levels, three cut scenes to show game information and the last two ending scenes (win and lose). The process of game is shown as the Figure3.8.

Result of Modeling and Analysis

In this game, there are three complex models, which are created respectively by the three modeling software: the Player by Blender, the Enemy by SketchUP and the Boss by Wings 3D. The complete models is shown as the Figure3.9.

Through the whole process from building of the three models to import into Unity3D, some problems were noticed, summarized in the Table3.1, and will be analyzed pertinently next.

- ready-made mesh

Blender provides both 2D and 3D mesh, including: cube, sphere, plane and so on which are commonly used. In contrast, Wings3D and SketchUP provide only 3D or 2D mesh. For example, the plane in Wings3D is actually a very flat block. In SketchUP the 2D

	Blender	SketchUp	Wings3D
ready-made mesh	2D and 3D	only 2D	only 3D
closed surface (automatic)	—	only in the same layer	Subdivision surface is also possible
normal mapping (front/back)	Batch is possible	without batch	—
materials and colors	√	√	Vertex Attribute - Colors to Materials
supported formats[32]	.fbx .stl .3ds .dae .obj	.fbx(only pro) .3ds .dae .obj	.stl .3ds .dae .obj
not a whole model (in export)	—	.3ds .dae .obj	.obj
models shrank (in Export)	.fbx(0.01 of original) .3ds(0.1 of original)	.fbx(0.01 of original) .3ds(0.1 of original)	.3ds(0.1 of original)
selection only (Mesh Renderer in Unity3D)	.fbx .3ds .dae	.dae	.3ds .dae

Table 3.1.: the Comparison of Existing Modeling Software

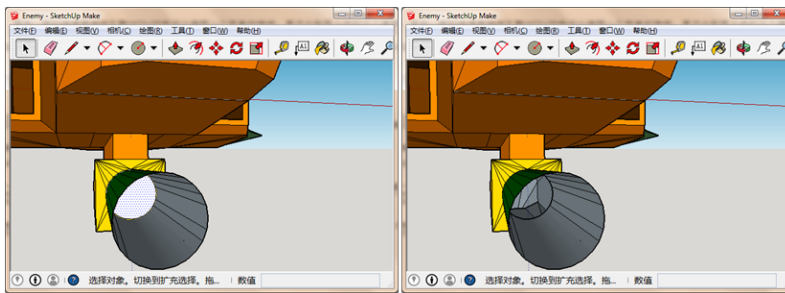


Figure 3.10.: extra white Cross-Section (left)

meshes e.g. circle and plane can also be extruded to increase thickness to reach the effect of cube and cylinder.

- closed surface (automatic)

In video games the 3D models used to be asked in the best shell shape. In order to avoid the error when the surfaces intersect, here these three models will be also completely closed model for the target. At this point Wings 3D works particularly well, because the surface of model in Wings 3D will be closed automatically. Even when the points are not in the same plane, the surface will be closed by the way of subdivision surfaces. In SketchUp the surface or the plane will be automatically generated only when the points in the same plane.

Therefore, when the conditions are met, SketchUp will also generate automatically cross-section in the interior of the model, as the selected white cross-section shown in left of the Figure3.10. Sometimes it brought some troubles, especially when a pure shell model is needed.

3. Environment and Analysis



Figure 3.11.: the Models of Correct Display and Display Error

In contrast, Blender has not similar functions. The user has to do manually if they want to generate a surface with points.

- normal mapping (front/back)

Normal mapping, also called Dot3 bump mapping is an application of bump mapping technology in 3D computer graphics which is used for faking the lighting of dents and bumps to add details without adding more polygons[33]. There are two types of normal mapping in Blender: recalculate outside and recalculate inside. If the normal mapping is not unified, e.g. most surfaces of a model are “recalculate outside” but the rest are “recalculate inside”, the display error will appear after the model was imported into Unity3D (see the Figure3.11).

The similar situation is also occurred in SketchUp. In SketchUp each surface or plane of model is divided into light front side and dark back side, so that different materials can be set separately both two sides. When the surfaces are not all front side outside, the display error will appear in Unity3D too.

At this point, because there is no 2D mesh in Wings 3D, and all of the surfaces are always closed automatically, so there is no problem about “normal mapping” in Wings 3D.

- materials and colors

The model can be set materials and colors in all these three software. The caveat is that the model in Wings 3D exports only its material, so that the colors of the model should be converted to material at first, when the colors need to be exported too (Vertex Attribute - Colors to Materials).

- supported formats

All the three software support several different formats. Here just list the five commonly used formats which are introduced in chapter 2. Different formats have different

influences to the model of, especially in exporting. For example, the model of some formats is no longer a whole model and is divided into several parts by different materials or colors or the volume size of the model is changed. Especially for the model of .obj format, because its materials and colors are saved in the associated files .mtl, the materials and colors won't be displayed directly and correctly when the .obj model is imported in Unity3D.

- supported formats Selection only is very important, when the user only want to export the selected objects. Otherwise all visible and all invisible objects in the entire scene will be exported. In the case of the latter situation, because the exported model is not independent, so some property of the model which is imported into Unity3D, such as materials and colors, cannot be set directly. Meanwhile some function (component) which Unity3D provide to the imported model, cannot act directly on the model, e.g. "Mesh Renderer". This will bring some difficulties to game design and game program in Unity3D.

In this regard SketchUp are lacking, although SketchUp has also the function "selection only". When selection only is enabled in SketchUp, the exported model of some formats will still attach with some invisible objects, so that some of the properties and functions cannot be set directly. However the eligible model of format .dae cannot often be a whole after exporting from SketchUp. So sometimes the exported model from SketchUp needs a third-party software as a relay, such as Blender, when the user want to import an appropriate model into Unity3D.

4. Concept and Design

By summarizing analysis of several modeling software in the last chapter, in this chapter, the requirements of the new tool will be presented. Then according to these requirements we will draw up the design concept of this new tool and the corresponding functions.

4.1. Requirements Analysis

Requirements analysis is critical to the success of a software project[34], including the tasks that are to determine the need, meet for a new or altered product, the possibly conflicting between the various stakeholder, and etcetera.

In this project, the main work and the main purpose is to develop a 3D modeling tool. The primary requirement of this tool is of course the 3D modeling functions:

- based on simple 3D-low-poly-modelling
 - In order to decrease render times a polygon mesh has relatively few polygons.
- basic functions of 3D modeling
 - The ready-made meshes are provided.
 - The models are created with different shapes and colors.
- to ensure the quality of the model
 - The edge of model is optimized, e.g. to achieve the arc with the algorithm of subdivision surfaces.

Secondly, there are also three further important requirements for the tool. According to these requirements, some of the more specific child requirements are expanded after the analysis of section 3.2.

- intuitive GUI
 - The framework and interface of the tool should be designed user-friendly.
 - It is a quick and understandable to change the parameters.
- apply to game design or other aspects
 - The model is exported to Unity3D with .obj format.
 - The model should be a completely closed and pure shell model.
 - By taking Wings3D as a reference, the model should be close surface automatically to avoid the problem of normal mapping.
 - Function “selection only” chooses the model and keeps its independence.

4. Concept and Design

Finally, this tool must also have some general requirements, like all good used software.

- reliability and robustness
 - The errors' probability has to be kept low and an error has only little effect, if it is unable to avoid
 - Incorrect operation will not lead to crash.
- extensibility
 - The framework of the tool should be able to be expanded easily for additional function.

4.2. Workflow

In section 3.2.2, a simple modeling process has been described. To reference that process, and to combine with requirements of the tool in last section, the modeling process will be designed. The Figure4.1 describes the using process of the tool. Like other 3D modeling tools on the market, this tool provides the function from creating a new model to completing it, then saving and exporting the its data.

4.2.1. Ready-made Mesh

After starting the tool, first the user can create a new or open an existing file. Then the user can use the Ready-made Meshes those are provided by the tool to start modeling.

In order to make the operation as simple as possible, this 3D modeling tool is inspired by the sandbox indie game “Minecraft” (see the Figure4.2). The game Minecraft allows players to build constructions out of cubes which are sized uniformly and textured variously in a 3D procedurally generated world[35]. Therefore the basis Ready-made Mesh that is provided by this tool is also a fixed unit cube. By constantly adding cubes and manipulating them, the user can build their own model, just as people build houses with bricks.

4.2.2. Catmull-Clark Subdivision Surface

Theoretically, if there is a sufficient number of unit cubes, people can build a model of arbitrary shape. But sometimes it requires too much work, especially when you need curved surfaces. Here we can apply the function of subdivision surfaces to solve this problem as much as possible.

The subdivision algorithms include two steps: the topological connection of points in new mesh and vertices' geometric positioning, and the common subdivision algorithms are based on quadrilateral or triangular mesh. Catmull-Clark Subdivision Surface which is the subdivision algorithm for arbitrary quadrilateral mesh is one of the most common schemes to create smooth surfaces in computer graphics. This algorithm is a promotion of bi-cubic uniform B-spline surfaces which was devised by Edwin Catmull and Jim Clark in 1978[37][38].

A B-spline, or basis spline is a spline function to give support for degree, smoothness and

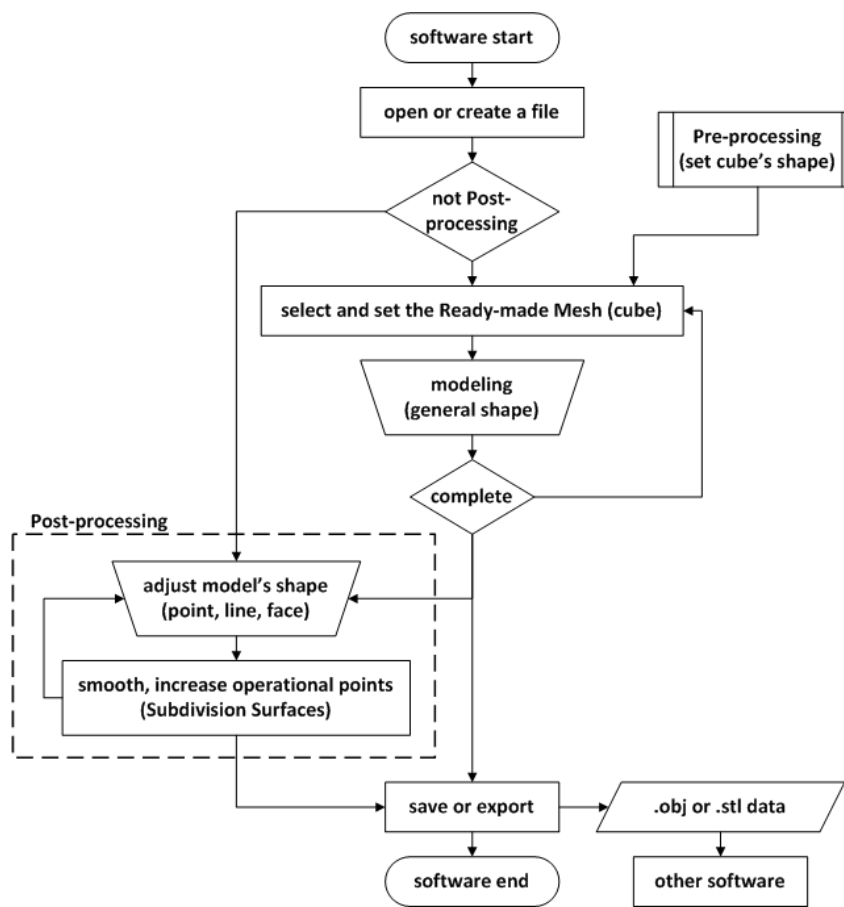


Figure 4.1.: the Workflow of the Tool



Figure 4.2.: the Screenshots of Minecraft[36]

4. Concept and Design

domain partition in numerical analysis. In the CAD and computer graphics, the B-spline constructs spline functions as linear combinations with a set of control points. The B-spline is a generalization of Bézier curve, and can further be promoted into NURBS which can provide more geometry to build more accurate models.

The B-spline curve can be subdivided by subdivision and iteration of its control points, which is useless for curve but can be extended into curve surface to produce a subdivision surface.

The detailed algorithm of Catmull-Clark Subdivision Surface is described as follows[39]:

- All of the vertices in a mesh of an arbitrary polyhedron should be called original points.
- A face point which is the average of all original points of the respective face is added for each face.

– formula of the face point (red points in the Figure4.3):

$$V_F = \sum_{i=1}^n \frac{1}{n} V_i \quad (4.1)$$

- V_i : the vertex of the face
- n : the number of the vertices

- The average of the edge's two original endpoints and its two neighbouring face points is added as an edge point for each edge.

– formula of the edge point (blue points in the Figure4.3):

$$V_E = \frac{1}{2} \left[\frac{1}{2}(W + V) + \frac{1}{2}(V_{F1} + V_{F2}) \right] \quad (4.2)$$

- W, V : two original endpoints of the edge
- V_{F1}, V_{F2} : the two neighboring face points of the edge

- Each original point V moves to the place of the new vertex point V' with following formula.

– formula of the new vertex point (black points in the Figure4.3):

$$V' = \frac{1}{n} F + \frac{2}{n} R + \frac{(n-3)}{n} V \quad (4.3)$$

- n : the number of neighboring edges
- F : the average of all face points for faces touching point V
- R : the average of all edge midpoints for edges touching point V

- For each face point connects the edge points those are added for every edge of the same face.
- Each new vertex point connects the edge points of the edge that the original point is touching to.

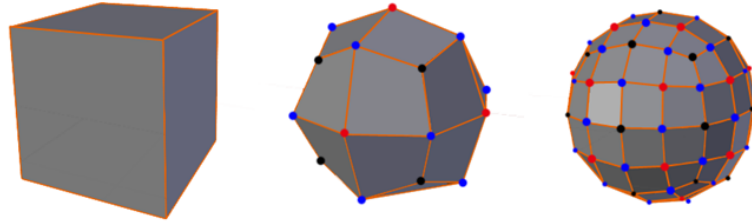


Figure 4.3.: the Effect of Catmull-Clark Subdivision Surface
(no subdivide, subdivide once and subdivide twice)

- The new faces are defined as enclosed by edges.

The Figure4.3 shows the effect of Catmull-Clark subdivision surface, from left to right: the original mesh, the mesh that was subdivided once and twice.

4.2.3. the Pre-processing

The pre-processing is a pre set process in this tool. According to the algorithm of Catmull-Clark subdivision surface and its change, the simplex Ready-made Mesh will be subdivided into variety of different basic shapes to deal with more complex modeling. While modeling the user still manipulate the blocks those are marked with black line, such as we introduced in section 4.2.1 above. The actual effect of the model is indicated by the orange line, and this part is done automatically by the tool itself.

Through different subdivision algorithm, the Ready-made Mesh is divided into three different shapes. The degree of subdivision for mesh in the pre-processing can be once of subdivision, or further twice. All of these can be selected and set in the preset of Ready-made Mesh. The detailed scheme of different shapes will be introduced in the following.

Shape 1: Block

The subdivision algorithm of block is similar to Catmull-Clark subdivision surface, but the edge points and the new vertex point have changed.

- formula of the changed edge point (blue points in the Figure4.4):

$$V_E = \frac{1}{2}(W + V) \quad (4.4)$$

– W, V : two original endpoints of the edge

- formula of the changed new vertex point (black points in the Figure4.4):

$$V' = V \quad (4.5)$$

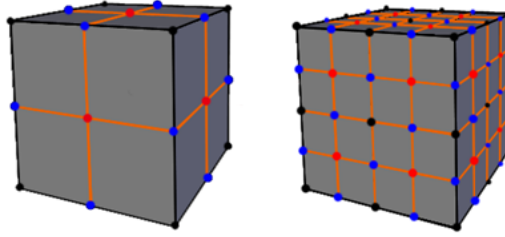


Figure 4.4.: the Effect of Subdivision Surface for Block

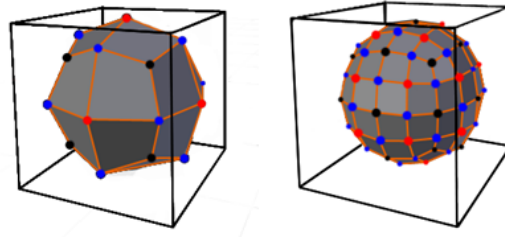


Figure 4.5.: the Effect of Subdivision Surface for Sphere

Shape 2: Sphere

The subdivision algorithm of sphere is the application of Catmull-Clark subdivision surface. Although the shape of the sphere will be more optimized with the increasing of subdivision, on the other hand the volume of the entire body will be relatively smaller. Therefore, the highest number of subdivision in the pre-processing is twice(see the Figure4.5).

Shape 3: Octahedron

The subdivision algorithm of octahedron is also similar to Catmull-Clark subdivision surface, and the edge points and the new vertex point have changed too. Meanwhile the whole method of subdivision will be changed after the first subdivision.

- formula of the changed edge point (blue points in the Figure4.6):

$$V_E = \begin{cases} \frac{1}{2}(V_{F1} + V_{F2}), & \text{subdivide time} = 1 \\ \frac{1}{2}(W + V), & \text{subdivide time} \geq 2 \end{cases} \quad (4.6)$$

- V_{F1}, V_{F2} : the two neighboring face points of the edge
- W, V : two original endpoints of the edge

- formula of the changed new vertex point (black points in the Figure4.6):

$$V' = \begin{cases} F, & \text{subdivide time} = 1 \\ V, & \text{subdivide time} \geq 2 \end{cases} \quad (4.7)$$

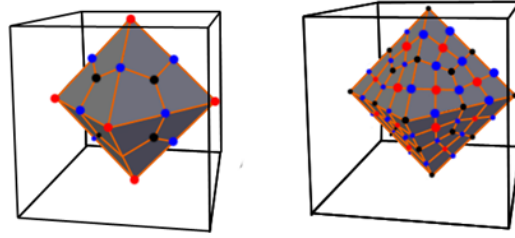


Figure 4.6.: the Effect of Subdivision Surface for Octahedron

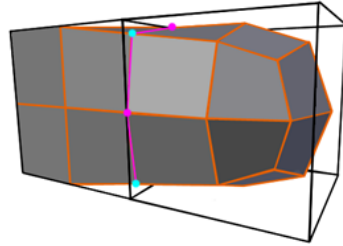


Figure 4.7.: Join of Block and Sphere

- F : the average of all face points for faces touching point V

Join between Cubes

It can be seen as above that the surfaces of block are subdivided, but the shape does not change. In the same way, the shape of octahedron does not change since the second subdivision. Here, the purpose of subdivision is in order to correspond to the subdivision of sphere, so that the cube with different deformation can be jointed together freely and the closing of entire mesh will be ensured, as shown of the Figure4.7.

Each cube is subdivided into one of three different shapes: block, sphere or octahedron which is also as a shape attribute for the cube. When the cubes with different shape joint, each cube is subdivide still in accordance with its own set of shape attributes, except the edge points and the new vertex points in connection part.

- The edge point (magenta points in the Figure4.7) in connection part is calculated by the edge which neighbors two faces of the two cubes each with different shape attributes.

4. Concept and Design

- formula of the edge point in connection part:

$$V_E = \begin{cases} \frac{1}{2} \left[\frac{1}{2}(W + V) + \frac{1}{2}(V_{F1} + V_{F2}) \right], & \text{block and octahedron} \\ \frac{1}{2} \left[\frac{1}{2}(W + V) + (V_{F1} + V_{F2}) \right], & \text{sphere and octahedron} \\ \frac{1}{2} \left[(W + V) + \frac{1}{2}(V_{F1} + V_{F2}) \right], & \text{block and sphere} \end{cases} \quad (4.8)$$

- W, V : two original endpoints of the edge
- V_{F1}, V_{F2} : the two neighboring face points of the edge
- The new vertex point (cyan points in the Figure4.7) in connection part means the point which belongs to several cubes with different shape attributes simultaneously.
 - formula of new vertex point in connection part

$$V' = \frac{1}{m} \sum_{i=1}^m V'_i \quad (4.9)$$

- m : the number of neighboring cubes
- V'_i : the new vertex point for neighboring cube with its own shape attribute
- For example: the new vertex points in the Figure4.7 (cyan points) are the average of the new vertex points V'_1 of block and V'_2 of Sphere:

$$V' = \frac{1}{2}(V'_1 + V'_2) = \frac{1}{2} \left[V + \left(\frac{1}{n}F + \frac{2}{n}R + \frac{n-3}{n}V \right) \right] \quad (4.10)$$

4.2.4. the Post-processing

The post-processing means the phase to optimize further the details of model after the completion of basic model in the main modeling phase. The process is free to choose that users can decide themselves whether the process is needed according to the completeness of model.

The post-processing can optimize further the shape of the entire model according to the algorithm of subdivision surface and its variants after the main modeling phase, so that the shape of the model can be smoother. Meanwhile, different from the pre-processing and the main modeling phase, the structure of cube which is provided by Ready-made Mesh will not continue to use in the post-processing, but the traditional modeling units like point, edge and face will be used.

It can get more accurate model when each point, line and face in model can be modified. For example, because it is difficult to generate automatically a cylinder by the algorithm

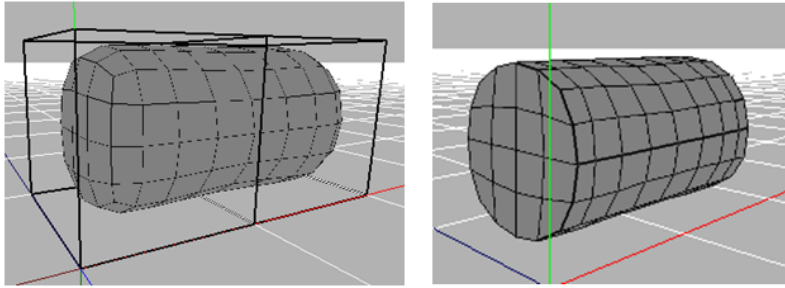


Figure 4.8.: the Approximate Model and the Accurate Model of Cylinder

of Catmull-Clark subdivision surfaces, so the shape of cylinder does not be provided as a Ready-made Mesh in this modeling tool. In the pre-processing and the main modeling phase, this modeling tool can only build approximate cylinder model (see left of the Figure4.8), but in the post-processing the user can change the approximate model into an accurate model of cylinder by modifying the specific points, edges or faces, especial the two sections on either side of the cylinder in this case (see right of the Figure4.8).

4.3. Modeling Functions

In this section, the functions of this modeling tool at present will be introduced briefly.

Add Cube and Batch

The addition of new cube in this modeling tool is in two ways which both cannot be used in the post-processing. One way is to add cube through the toolbar AddCube by the given coordinate where is the location for the new cube. The other way is to extrude new cube next to the selected face of the existing cube. Both ways support batch processing, can add cubes in the same time. The batch processing of cube addition is with the help of toolbar AddCube too. The user determines the batch mode at first and gives the three numbers of new cubes along the axis (x, y and z), and then click the button to finish the addition of cubes (see the Figure4.9). The batch processing of function Extruded is with the help of multiple select. The user can select more than one face of one cube or many cubes, and extrude a new cube next to each selected face in the same time (see the Figure4.10). Adjacent cubes can are joint each other themselves in order to ensure the closed of the entire mesh.

Factors for Catmull-Clark Subdivision Surface

In order to further increase the variety of the cube's shape and to modify its changes more freely, the modeling tool adds three factors which correspond to face point, edge point and new vertex point respectively based on the algorithm of Catmull-Clark Subdivision Surface.

By comparison of the formula about edge point V_E and new vertex point V' of the three different shapes, it can be found the three edge points (blue points) or the three new vertex points (black points) are located on the same line (see the Figure4.11). Then we can

4. Concept and Design

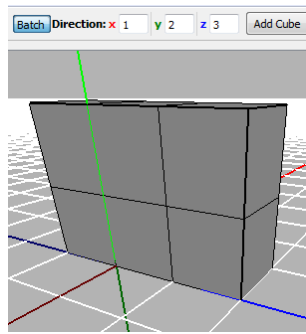


Figure 4.9.: the Batch Processing of Cube Addition

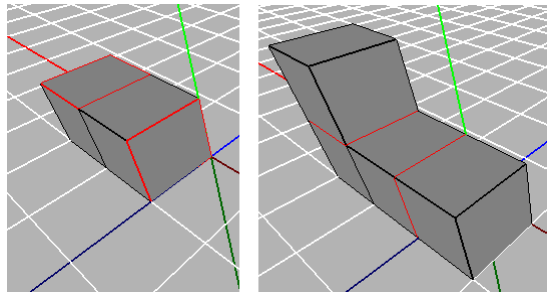


Figure 4.10.: the Batch Processing of Function Extruded

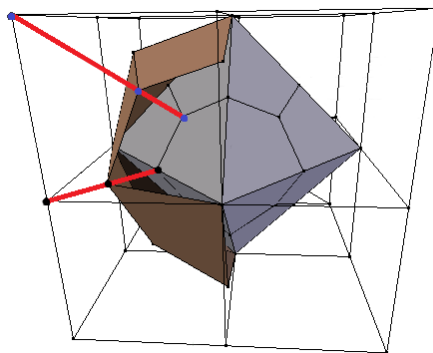


Figure 4.11.: the Edge Points or the New Vertex Points on the Same Line

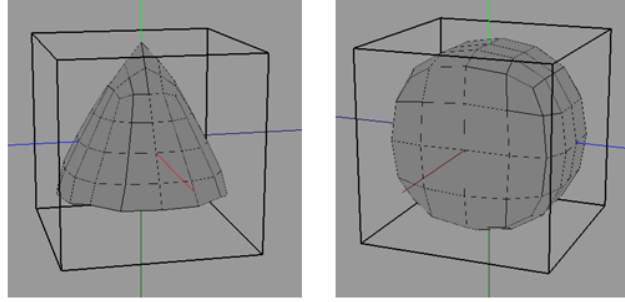


Figure 4.12.: the Variants of Cube's Shape

determine the value interval about the edge points and the new vertex point. And the new formulas are:

$$V_E = \frac{1}{2}(W + V)f_E + \frac{1}{2}(V_{F1} + V_{F2})(1 - f_E) \quad (4.11)$$

$$V' = V * f' + F * (1 - f') \quad (4.12)$$

- f_E : the factor for the edge point
- f' : the factor for the new vertex point

The Range of new face point is set simply from the original face point to the cube's center its formula is as follows:

$$V_F = \sum_{i=1}^n f_F \frac{1}{n} V_i + (1 - f_F) * C \quad (4.13)$$

- f_F : the factor for the face point
- C : the center point of cube

Each face of cube saves its corresponding factors, so the edge points and the new vertex point are controlled by all the neighboring face. Therefore the user can be unified to modify all the parameters in this modeling tool, the user can modify uniformly all the factors of cube, or only the factors of one face to changed the corresponding face's shape. As a result, except the three basic Ready-made Meshes the cube can be changed to more different shapes, for example: cone and tire-shaped 4.12.

Smooth

In this modeling tool, there are two methods of subdivision surfaces, one is Catmull-Clark Subdivision Surface which is played also as the role to smooth surface, the other is simply to increase sub-surfaces without shape's change (see the Figure 4.13). Here the two methods can be combined to use. The parameter of smooth is save as a Boolean in each face, so the user can choose freely which face needs smooth and which does not, and then subdivide faces

4. Concept and Design

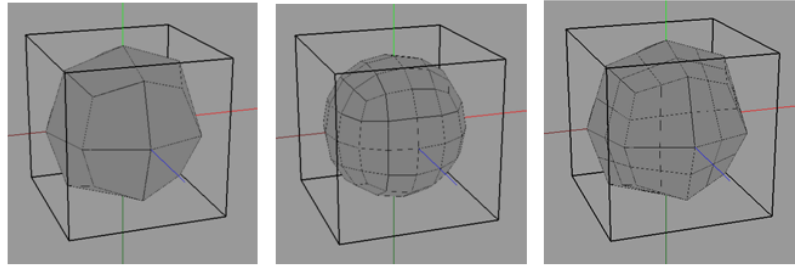


Figure 4.13.: before Subdividing (left), Smooth Subdivision (middle) and not Smooth (right)

together. No matter which phase, the degree of subdivision surface can be adjusted freely, its maximum time for subdivision is two. In the post-processing, the user can determine the effect of subdivision by function "apply". The new subdivided mesh can be subdivided continue.

In the post-processing, one important function of the subdivision surface is to increase the operational object (like point, edge and face) to replace the defect that the new cube cannot be added.

Select

There are total six selection modes in this modeling tool which are controlled by six exclusive buttons, so that only one mode can be chosen in one time. Except the mode of "no select", there are also "vertex select", "edge select", "face select", "body select" and "all select". The user can choose one mode and click the corresponding object by mouse which the user want to choose. The chosen object will be highlighted in red mark, and then user can use the related functions to operate the selected object. When in the mode of selection, the modifier key "ctrl" can be combined with mouse to choose multiple objects.

The mode of "body select" which is used to select cube cannot be used in the post-processing, because the structure of cube has be removed. In contrast, due to restrictions of cube structure, each points and edges cannot be modified freely in the pre-processing and main modeling phase. Therefore the mode of "edge select" is used only in the post-processing and the mode of "vertex select" in the pre-processing and main modeling phase can only display the coordinates of the selected point to help the users to better locating but cannot operate the selected point itself.

Delete

The function Delete which is activated by right-click menu is used to remove the selected cube or cubes in the pre-processing and main modeling phase. Because this function is based on the structure of cube, so this function cannot be used in the post-processing. The entire mesh will be closed automatically even after the cube or cubes which are the part of mesh are deleted.

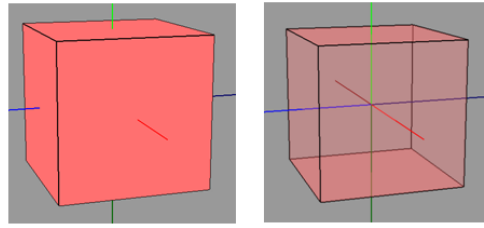


Figure 4.14.: Color and Transparency

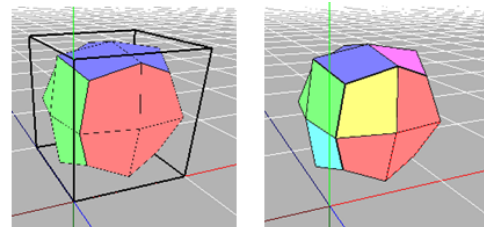


Figure 4.15.: Color for Face of Cube in main Modeling Phase (left) and Color for Face in Post-processing (right)

Color and Transparency

The color and transparency of model can be set (see the Figure4.14). Here, the color and transparency are as the parameters to save in each of the face, so the color of each face can be modify freely. But the parameters of sub-surfaces which are generated automatically by Catmull-Clark Subdivision Surface are inherited from the original face, and cannot change freely in the pre-processing and main modeling phase. To change the color of sub-surface, will have to determine the effect of smooth at first (see the Figure4.15).

Moving and Mirror

Moving and Mirror are peculiar functions in the pre-processing and main modeling phase which are both based on cube structure. Moving is controlled and implemented by toolbar. The user can give the expected distance of moving at first and then move the selected cube or cubes up, down, left, right, forward or backward by 6 directional buttons. Mirror is activated by right-click menu and copy the selected cube or cubes symmetrically according to the axes as center line.

Both the functions will remain the closed of mesh. For example, if a big mesh is split into two by moving a part of cubes, the two small meshes will close themselves respectively and automatically, or if the separate meshes contact with each other, then they joint automatically.

Position and Scale

Position and Scale are peculiar functions in the post-processing, and the objects on which they work can be any points, edges or faces. Position has two different types, one is the

4. Concept and Design

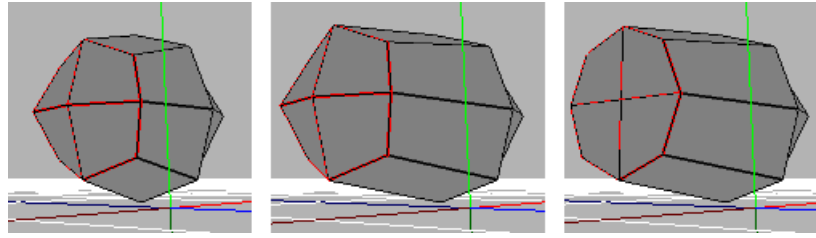


Figure 4.16.: the Original Position (left)
the Relative Moving (middle) and the Absolute Moving (right) along x Axis

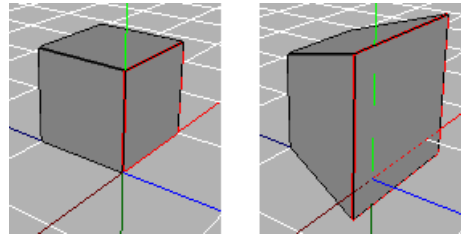


Figure 4.17.: the Function Scale

relative Position, and the other is the absolute Position.

The relative Position will calculate the center point of all the points of the selected objects and the distance between each selected points and the center point, and then move all the selected objects relatively without the change of relative distance between all selected points. The absolute Position is to move direct all selected objects which means each point of the selected objects to the target coordinates (see the Figure4.16).

The Function Scale needs also to calculate the center of the selected object at first, and then move all the points of the selected objects away from the center point or toward the the center point (see the Figure4.17).

Modeling File

As a fully functional modeling tool, there are also a series of related functions for modeling files, such as to create a new file, to open an existing file, to save file and so on. This tool provides two original file formats which are imitated OBJ format, respectively for the cube structure in the main modeling phase and the structure of vertices, edges and faces in the post-processing. As a result, the modeling work in any phase can be saved successfully in order to continue to work at any time.

In addition, the modeling tool can also export the model in OBJ format which is one of common 3D modeling file formats, so that the model can be used by more other software, such as Blender or Unity3D.

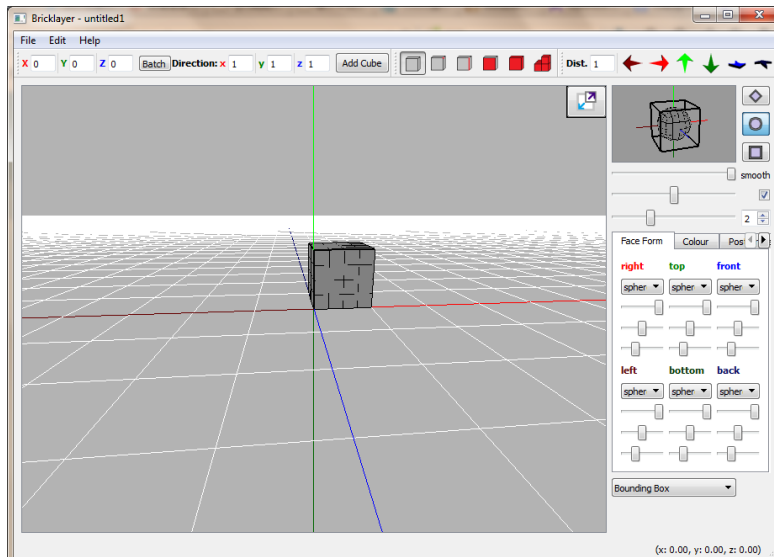


Figure 4.18.: the Screenshot of User Interface

4.4. User Interface

In this project, the user-friendly interface of the modeling tool is also a very important requirement. The GUI (Graphical User Interface) refers the user interface that allows users to operate computer through graphical icons[40]. Compared with the text-based interfaces of early computer, the graphical user interface is more visually and easier to be accepted for users. Otherwise, although this tool is used mainly in the PC at present, considering the possibility of transplant to mobile devices in the future, the simple and reasonable user interface is very necessary. At this point the, Qt provides a rich controls, in order to establish a beautiful and practical user interface. The Figure4.18 shows the screenshot of user interface of this modeling tool in project.

As the show of the Figure4.18, the GUI of this tool is mainly including one MainWindow: which is main application window to provide menu, toolbar, status bar and other controls:

- Menu: three submenus, File, Edit and About
- three Toolbar (see the Figure4.19):
 - AddCube: three LineEdits are used to give the coordinates of the position where the new cube will be added; a PushButton is used as a switch to open the batch mode; three LineEdits are used to give the number of new cubes along the axis in batch mode and finally a PushButton is used to add new cube/cubes.
 - SelectMode: a ButtonGroup with six exclusive PushButtons for each Select Mode (None, Vertex, Edge, Face, Body and All)
 - the function of Move: a LineEdit is used to give the distance of cube's movement and six ActionButtons are used to determine the direction of movement.
- MainWidget: It is main modeling window with axes and grid. The scene of the main modeling window can be rotated, translated and zoomed.

4. Concept and Design



Figure 4.19.: the Toolbars of AddCube (top), SelectMode (lower left) and the Function of Move (lower right)

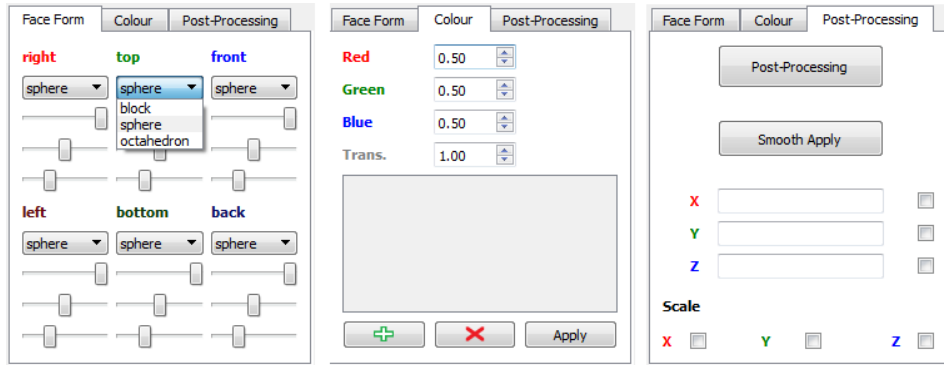


Figure 4.20.: the TabWidget for Toolbars with three Tabs

- MiniWidget: It is preview window for the presetting of new cube for the pre-processing and the main modeling phase and its scene can be rotated. The preview of cube is always centered.
- a PushButton: It is a switch to change the display of MainWidget and MiniWidget.
- a ButtonGroup with three exclusive PushButtons: Its function is to choose the shape of Ready-made Mesh.
- three Sliders: The Sliders are used respectively to modify the three factors (the factor of face point, edge point and new vertex point) of whole cube.
- a SpinBox and a Checkbox: The SpinBox determines the degree of subdivision surface and the Checkbox determines the kind of subdivision surface.
- a TabWidget for Toolbars with three Tabs4.20
 - FaceForm: Each face of cube can be modified individually.
 - * six ComboBoxes: They provide quick options for the basic shape of each face.
 - * eighteen Sliders: They represent the three parameters of each face.
 - Color and Texture:
 - * four SpinBoxes: They represent the RGB color and transparency.
 - * other controls: about texture (unfinished).
 - Post-processing:
 - * two PushButtons: One is used to open the post-processing. Once the post-processing is confirmed to open, it cannot return. The other is used to deter-

mine the degree of smooth.

- * three LineEdits: They are used to give coordinates where the selected objects will be moved to.
- * three CheckBoxes: They are used as switch to determine the relative movement or absolute movement.
- * three CheckBoxes: They are used to determine the direction (axis x, y and z) of Scaling.
- a ComboBox: It used to select polygon mode, for example Bounding Box and Wire Frame.
- a StatusTool: shows the tips and the description of the various functions and the warning of error operations.

5. Programming and Implementation

In this chapter, some specific algorithm and structure of program about this modeling tool will be introduced briefly.

5.1. Cube Structure

The cube structure is a 3D mesh structure which is specifically designed for this project. The following class diagram 5.1 shows simply its structure. For example in one cube, it includes eight vertices, twelve edges, and six faces. Each face of the cube has four edges and four vertices, and each edge contains of two vertices.

There are four classes: Vertex, Edge, Face and Cube. The relationships between Cube and Vertex, Edge or Face are Aggregations which are represented as a single line with a hollow diamond shape in the UML. The relationship Aggregation represents a part-whole or part-of relationship. In the aggregation relationship, one class can be a collection or container of the other classes, but the contained classes can also exist independently from the class of the container??. For example: the class Face is part of the class Cube, but the Face can also exist independently.

Each Edge and Vertex has an ID number, and all these ID numbers are stored as list of integer in order to be called easily by other classes. For instance, each Edge saves the two ID number of its vertices, so that the corresponding vertices of any one edge can be find quickly, rather than to save all the coordinates of vertex in the corresponding edge. Moreover each Face saves four ID numbers of its edges and four ID numbers of its vertices which are all stored in anticlockwise of the target face.

In addition, the class Face also set other related parameters:

- `int neighborID`: when two cubes joint from each other in order to maintain the closed and hollow of the whole mesh, the two overlapped faces will be deleted. This parameter of face is used to record the other face ID which is overlapping itself. If there is not overlapped face, this parameter holds the initial value of -1.
- `bool isDraw`: if the `neighborID` is not -1, which means the overlapped face is existing, this parameter is false and the face does not draw.
- `bool isSmooth`: this parameter is used to determine when the all the faces are subdivided, this face is subdivided by Catmull-Clark subdivision surface or just simple subdivision surface.
- `Color *rgbaValue`: it includes four double-precision floating points which represent RGB colors and transparency of the face.
- `float facePointFactor, edgePointFactor, newVertexPointFactor`: as the section 4.3 has

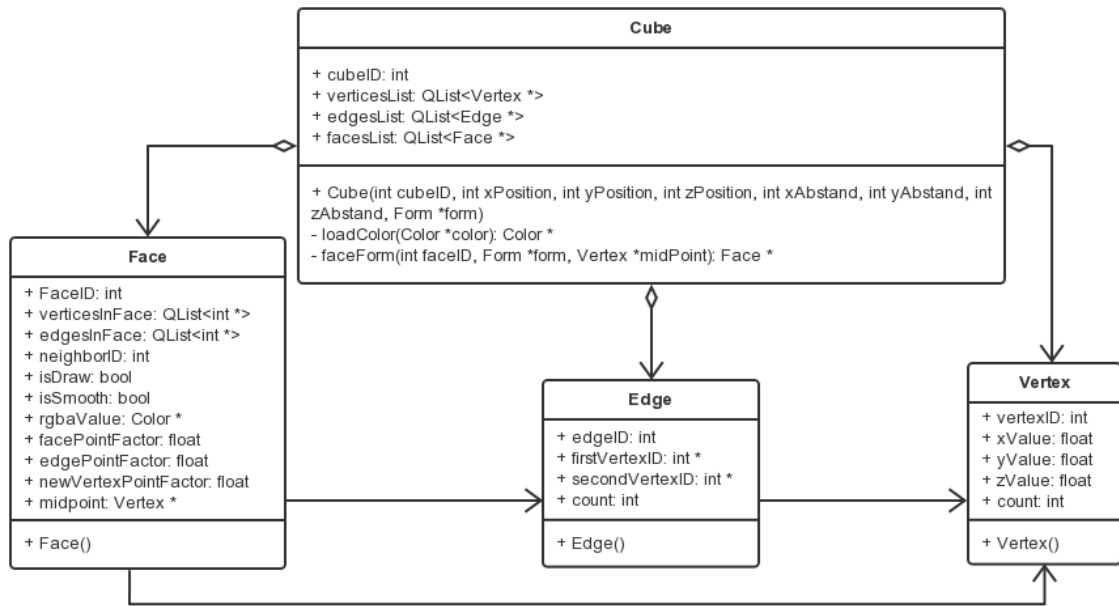


Figure 5.1.: the Cube Structure in Class Diagram

been introduced, this three factors are used to adjust the changed formula of subdivision surface to increase more different shape of each sub-faces which are generated by subdivision of the original face.

- Vertex *midpoint; this point is the center point of the cube which the face belongs to. Its main function is to calculate the new face point by changed formula of subdivision surface according to the section 4.3.

Each vertex or edge has set count value. The function of count value is similar to the parameter “neighborID” and “isDraw” in class Face, and is used to count the number of replications about the vertex or the edge when cubes joint each other. When the vertex or the edge becomes the internal point or line of the entire mesh because of the increasing cube, in order to maintain the closed and hollow of mesh, this point or line will not be drawn, that means the count value for vertex cannot be greater than or equal to 8 and for edge cannot be greater than or equal to 4. If the cube is deleted, the count value of corresponding vertices and edges will also be subtracted accordingly.

All information of the vertex, edge and face are also linked and saved in the corresponding cube simultaneously. In addition, there are three methods in the operation of class cube. The public method Cube() set all attributes of the new cube which are added in the given coordinates “xPosition”, “yPosition”, and “zPosition”. The two private method loadColor() and faceForm() are used to set the color and shape for each face of the cube.

5.2. Algorithm of Catmull-Clark Subdivision Surface

In the chapter 4, we have introduced the algorithm of Catmull-Clark subdivision surface and its variants; here its implementation process will be described briefly. All vertices, edges

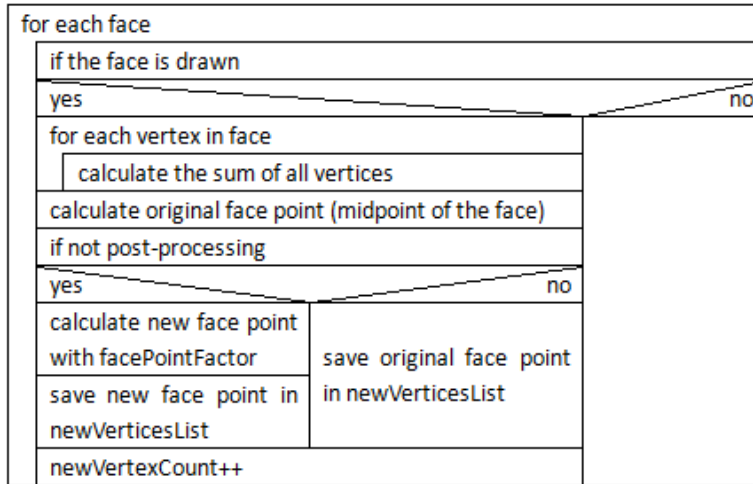


Figure 5.2.: the Calculation Process of Face Point

and faces will be numbered and their number will be saved as lists in class Mesh without duplicate. The subdivision of surface is will operate on all faces, edges and vertices which stored in the class Mesh. The following are Nassi-Shneiderman diagrams for the specific process[41].

First step is to calculate face point of each face which is the center point of all vertices in the face, and then save it as the original face point. When it is not in the post-processing, it also need the factor of face point and the center point of the cube which both are saved in class Face to calculate new face point according to the changed formula of subdivision surface, and save the new face point as the final result5.2.

The second step is to calculate edge point of each edge. According to the flowchart5.3, the program searches each face to find two neighboring faces which are touching to the edge, and saves their original face point, besides the midpoint of the edge is also needed. Because the required parameters from subdivision surfaces, like the factor of edge point, are saved in class Face. So to calculate the final edge point, at first it needs to calculate the two temporary edge points which are respectively corresponded the two neighboring face. And the final edge points is the average of the two temporary edge points

The third step is to calculate the new vertex point, and its process is similar to the calculation process of the edge point. The temporary new vertex point of each neighboring face is needed to calculate respectively at first by the saved parameters in corresponding class Face, and then calculate the average of all temporary new vertex points to get the final new vertex point. It should be noted that except neighboring faces the calculation of new vertex point requires also all the neighboring edges. Here, every vertex will has maximum five neighboring faces and neighboring edges5.4.

Finally, it needs to generate new sub-surface by connecting the calculated face points, edge points and new vertex points5.5. Here, because each face is constituted of four vertices, each original face will be subdivided into four sub-faces that each is still constituted of four vertices. In addition, the sequence of list which stores the vertices or edges of new sub-surface should be noted. Here the vertices and edges of each sub-surface are linked in the lists in

5. Programming and Implementation

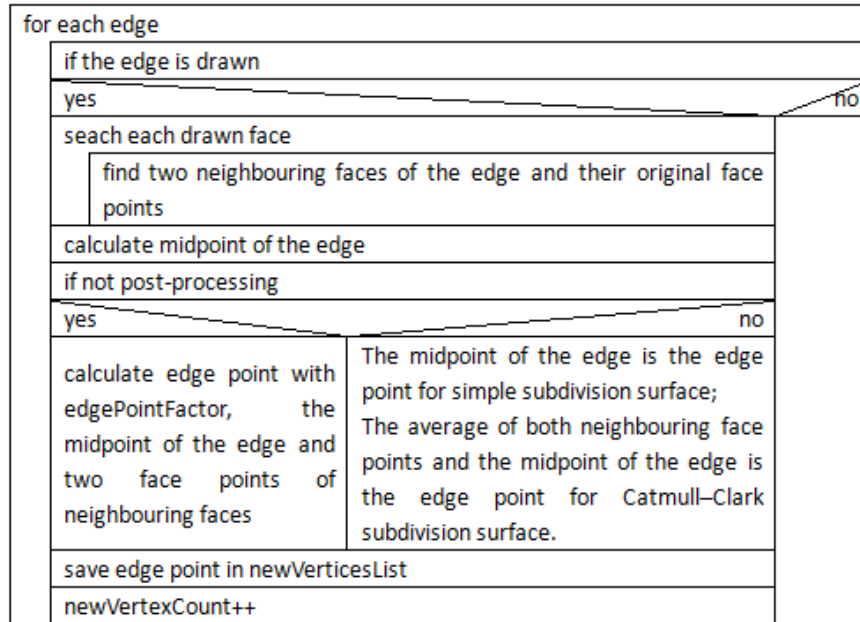


Figure 5.3.: the Calculation Process of Edge Point

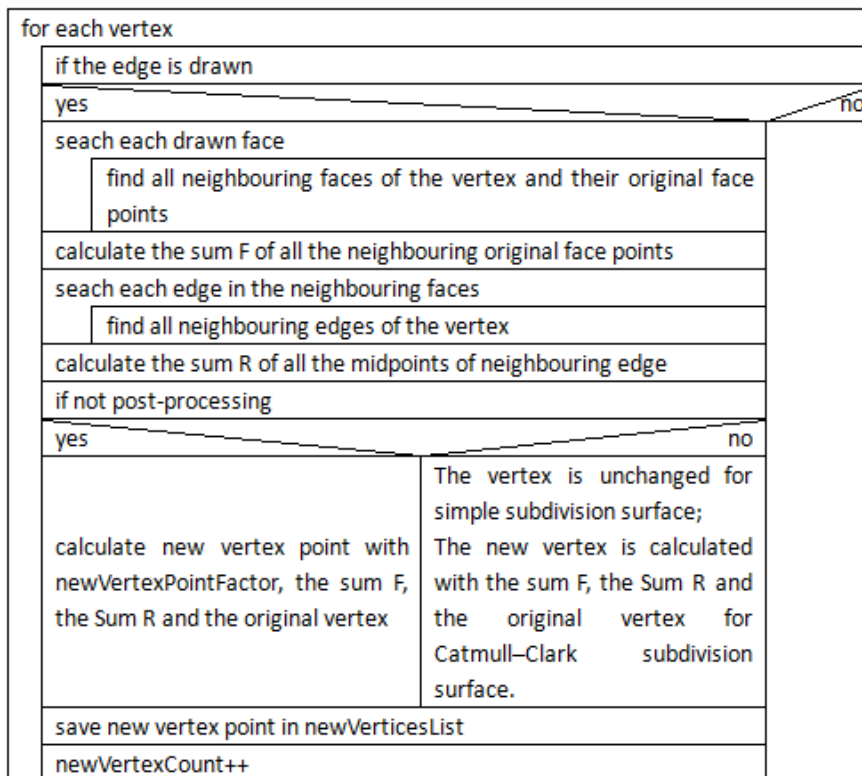


Figure 5.4.: the Calculation Process of New Vertex Point

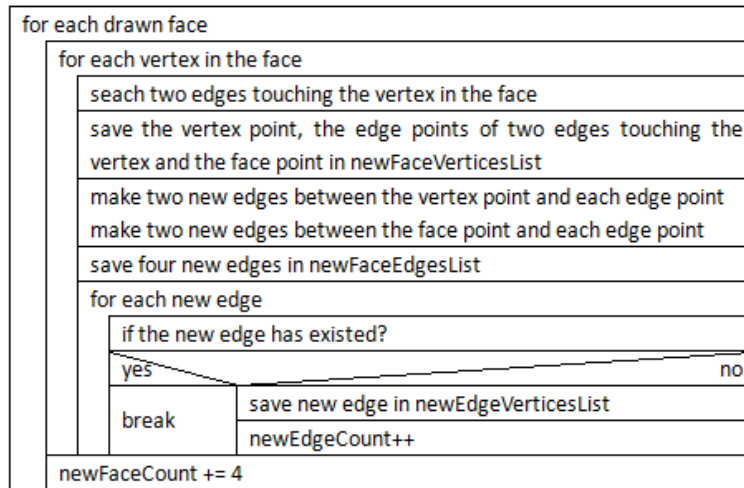


Figure 5.5.: the Generation of New Sub-surface

anticlockwise of the corresponding sub-surface.

5.3. 3D Modeling File Format

File Format OBJ

In order to increase the compatibility of this modeling tool, this tool is designed that can export the model in a variety of common 3D model formats. The first selected file format which has been successfully completed at present is the .obj format which has been introduced in section 2.3. In this modeling tool, since the completed model is perfect closed shell-model, so it does not consider the information of edges when the model is exported in the file format OBJ, but only need to record the information of all the faces and vertices which constitute those faces. And because the accompanying file .mtl of OBJ here is not completed yet, the related information about color and texture are also negligible at this moment.

It is to make an example of a simple cube (see the Figure5.6). Except the beginning and descriptive text, this file record the coordinate positions of all eight vertices in the cube, and each line of text about vertex starts with the keyword “v”. The line about face is with the keyword “f” at beginning. There are in total six faces of the cube and each with the numbers of four vertices in the face. For instance the vertex’s numbers of first face is 1, 4, 3 and 2, that means the Face 1 is made by Vertex 1, Vertex 2, Vertex 3 and Vertex 4 which are record in preceding part of the text. The sequence of the four vertices is anticlockwise along the face.

the Original File Format

This modeling tool provides its original file format to its modeling capabilities. Its file format has two different types which both are inspired by file format of OBJ.

5. Programming and Implementation

```

Datei Bearbeiten Format Ansicht ?
# OBJ File Exported from Bricklayer 1.0
mtllib a.mtl
o Cube
v 0.000000 0.000000 0.000000
v 1.000000 0.000000 0.000000
v 1.000000 0.000000 -1.000000
v 0.000000 0.000000 -1.000000
v 0.000000 1.000000 0.000000
v 1.000000 1.000000 0.000000
v 1.000000 1.000000 -1.000000
v 0.000000 1.000000 -1.000000
usemtl Material
s off
f 1 4 3 2
f 1 2 6 5
f 2 3 7 6
f 3 4 8 7
f 4 1 5 8
f 5 6 7 8

```

Figure 5.6.: the File Format OBJ

```

Datei Bearbeiten Format Ansicht ?
# BRL File Saved from Bricklayer 1.0
type 1
cube 0.000000 0.000000 0.000000 0.5 0.5 0.5 1 1.000000 1.000000 1.000000 0 0.5 1 0.5 1 1.000000
0.500000 0.300000 1 0.5 0.5 0.5 1 1.000000 1.000000 1.000000 0 0.5 0.5 0.5 1 1.000000 1.000000
1.000000 0 1 0.5 0.5 1 1.000000 0.500000 0.300000 1 0.5 0.5 1 1 1.000000 1.000000 1.000000 0

```

Figure 5.7.: Type 1 of the Original File Format

The first type is to save model in cube structure. First, this text of file is also beginning with “#”, and the second line record the type of the original file format, then each line represent one cube and starts with the keyword “cube”. For example one cube (see the Figure5.7), the file records the information of one cube. First three values which are marked red are the position of the cube which means the coordinates of the starting vertex of the cube. Later the information are the parameters of the cube’s six faces which are separated by orange lines. The four values in the blue box represent RGB colors and transparency, and the three values in the green box are the three factors which are used to subdivision surface. The last is Boolean value to determine whether this face needs to smooth.

The second type is to save model with vertices, edges and face5.8. This type is closer to the file format OBJ, but adds colors and other related information. The each line of main content starts with the keyword “v”, “e” or “f” to represent the vertex, edge or face respectively. The Information of vertices include the coordinates of the point, and the edge’s information record the ID number of both points in the edge. The Information about face is including the ID number of its vertices which are in the red box, the ID number of its edges which are in the blue box, the RGB color and transparency which are in the green box and finally the Boolean value for smooth.

```

Datei Bearbeiten Format Ansicht ?
# BRL File Saved from Bricklayer 1.0
type 2
v 0.000000 0.000000 0.000000
v 1.000000 0.000000 0.000000
v 1.000000 0.000000 -1.000000
v 0.000000 0.000000 -1.000000
  :
e 0 1
e 1 2
e 2 3
e 3 0
e 4 5
  :
f 0 3 2 1 0 3 2 1 0.5 0.5 0.5 1 0
f 0 1 5 4 0 9 4 8 0.5 1 0.5 1 1
  :

```

Figure 5.8.: Type 2 of the Original File Format

6. Discussion and Conclusion

In this chapter, we will analyze and discuss the performance of this model tools by practical use case. After summarizing the advantages and disadvantages we will give the future direction of development and solutions for the question which are found as far as possible.

6.1. Practical Use Case

In the previous chapters, the design concept, functions and some implementation methods of this modeling tool have been introduced. Next we will use this tool to build a specific model. The referenced model is selected the simple spaceship model that is modeled by SketchUp in chapter 3 (see the Figure3.9).

6.1.1. the Description of Modeling Process

The first is to build the basic shape of the model of simple spaceship in main modeling phase. The first cube is added by the given coordinates which are input direct through the toolbar AddCube. Here the batch function can also be selected to add more cubes as the main hull of the spaceship in one time. Then the user can expand model sequentially and quickly by the second method of cube adding which is based on extruding of the existing cubes. While the modeling process, the shape of cube can be preset and viewed in preview window before it will be added, or can select the existing cube to change the shape of corresponding selected face. Consider the automatically change of cubes' shape when the surrounding cubes joint each other, it is better to modify the shape of cube after it has added, especially the time when a lot of cubes with complex shape change are needed. In addition, in the main modeling phase, it provides function Mirror. This model of spaceship is symmetrical, so using the Mirror function can greatly improve the speed of modeling. The Figure6.1 is the basic shape of the finished model in main modeling phase.

Later the entire model will be subdivided once for the setting of details. For example the barrel of spaceship is modeled by Sphere form to realize similar cylinder, and the top of spaceship is modeled by Octahedron form to realize ramp (see the Figure6.4). If the more detailed model is needed, it can subdivide the model one more time. The Figure6.5 shows respectively the look of model by subdividing surface once and twice. Here has a small problem about the Sphere form of cube (see left of the Figure6.2). The original Catmull-Clark Subdivision Surface does not produce jagged edges (see the Figure6.3). Here this tool has imported the concept of factors (edgeFactor and newVertexFactor) to modify the formula of subdivision surface and to make the location of the new generated edge points and new vertex points more freely. But when the tool realizes the effect of the original Catmull-Clark Subdivision Surface with this concept, the factors have to be used as the

6. Discussion and Conclusion

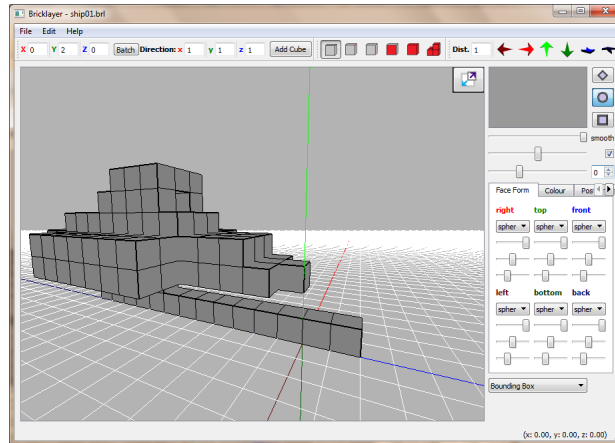


Figure 6.1.: the Basic Shape of Model without Subdivision Surfaces

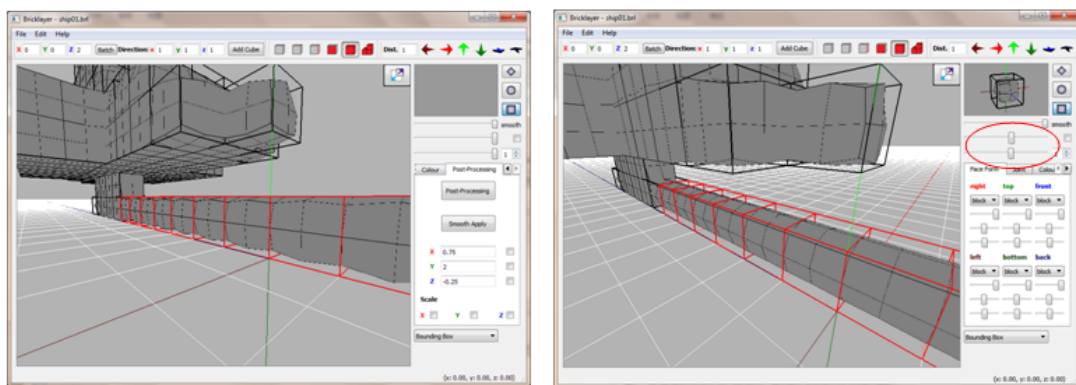


Figure 6.2.: the Jagged Edges (left) and the Smooth Edges (right)

approximation. And this error is caused the current situation. However this problem can also be solved easily by the controls of factors (see right of the Figure6.2). Selecting the cubes and modifying factors (edgeFactor and newVertexFactor) uniformly, when both are basically the same, the jagged edges is eliminated.

In the status of keeping subdivision, it is more intuitive when the new cube is directly added and its shape is modified in real-time. But the complicated calculation of subdivision surfaces will give the system burdens, the more time of subdivision is the greater burden on the system correspondingly, and the reaction speed of the entire tool will slow down.

Following is the post-processing. After confirming to begin the post-processing, the tool will transform the model which is showed in main modeling window directly into the post-processing mode. At the moment, the shape of transformed model has no change, but the cube structure for model has removed (see the Figure6.6). In the post-processing it also provides the function of subdivision surfaces. The user can transform the model with original cubes into the post-processing and then subdivide surfaces, or transform the model directly with the status of keeping subdivision. Here, because the model is not much complicated, it will be subdivided once and then transformed directly into the post-processing.

So far a spaceship with rough shape has basically appeared. In order to make it look

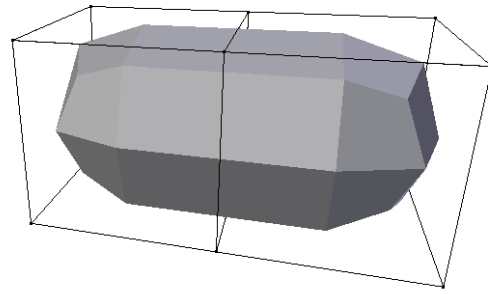


Figure 6.3.: the Original Catmull-Clark Subdivision Surface

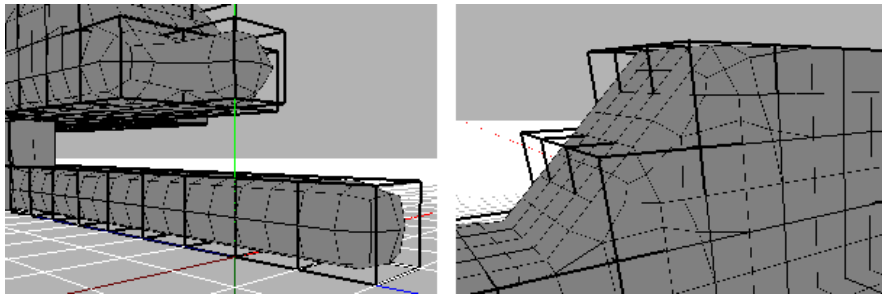


Figure 6.4.: the Barrel (left) and the Top (right) of Spaceship

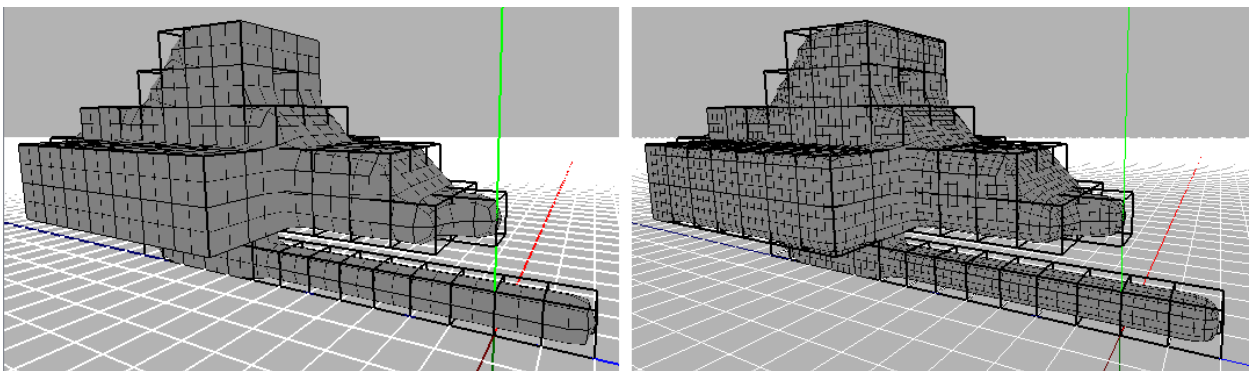


Figure 6.5.: the Model by Subdividing Surface Once and Twice

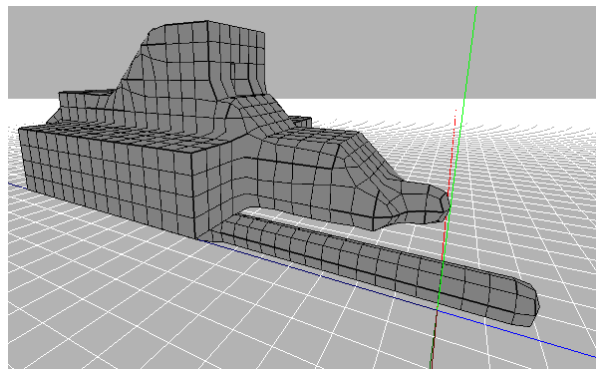


Figure 6.6.: to Transform the Model in the Post-processing

6. Discussion and Conclusion

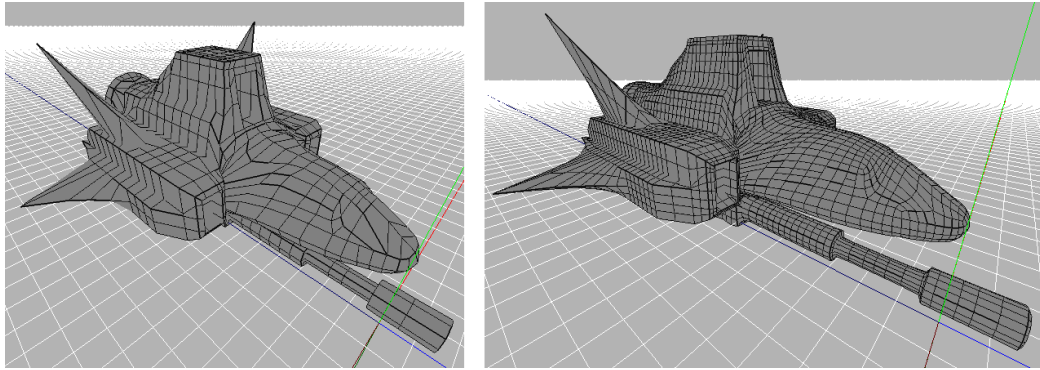


Figure 6.7.: the Finished Model (left)
and the Model by Subdividing Surface One More Time (right)

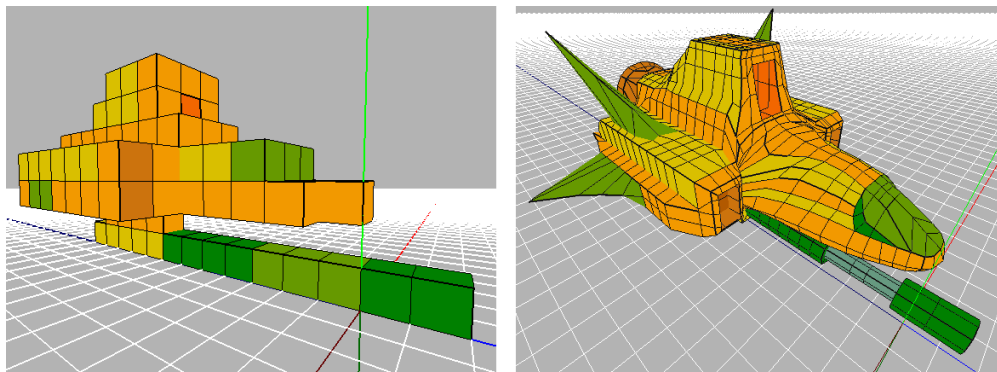


Figure 6.8.: the Color Settings in the main modeling phase (left)
and in the post-processing (right)

more beautiful, in the post-processing, the details of model which means each vertex, edge or face can be re-edited by modification of the coordinate for position. It should be noted that, although the model is a perfect closed shell-model, if the positions of vertices were moved excessively without plans, it will still cause the cross or superposed lines and faces to damage the shell-model. There are two useful functions in the post-processing: Scale and the change of Position which play an important to improve the efficiency of modeling. The finished model is shown in the Figure6.7.

Alternatively for color settings, it can be set in the main modeling phase and be finer changes in the post-processing, or also can be set directly in the post-processing as shown in the the Figure6.8.

6.1.2. Model Export

To complete a model is not usually the end; the key point is how to use it. Sometimes the completed models will be exported, and then will be imported into other higher-end modeling software or game engine for CG, animation or game production, and sometimes the model will be used for rapid prototyping. This modeling tool also can export the model in OBJ file, although associated file .mtl about colors and textures has not complete yet,

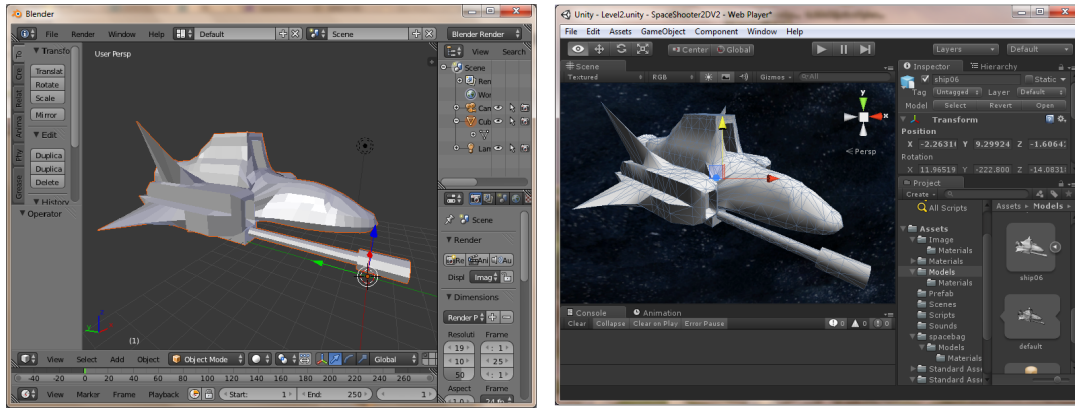


Figure 6.9.: the Exported Model is Re-imported in Blender (left) and Unity 3D (right)

the export of model shape has realized perfectly. The Figure6.9 is exported model which is re-imported successfully by the modeling software Blender and the game engine Unity 3D.

6.2. Lessons Learned

6.2.1. Performance Evaluation

This modeling uses a total of 195 cubes. With the help of referenced model the entire basic shape of model is completed fast and easy, and took the time no more than 10 minutes. In keeping the state of subdivision once, the modifying of model in details took more than 10 minutes. And because the loads on the system, the response speed of modeling tools is slow down with increasing cubes. Here after the number of existed cubes reaches more than 80, it is difficult to add new cube in this state directly with mouse event which is means to extrude new cube from the selected face of existed cube by mouse. However, the following two ways to add new cube have not any influence. One is to add new cube by the toolbar with the given coordinates. The other is to return back the state without subdivision to add cube, and then to subdivide again. In the case of keeping state of subdivision twice, after 25 cubes the adding of new cubes will become difficult with function Extrude. Therefore, it is difficult to modify the detail of whole model in real-time, when the model are subdivided twice.

In addition, although this modeling tool does not provide the function that the model can be subdivided more than two times in the main modeling phase, we have still tested the state of subdivision third times and fourth times. When third times, the threshold of cubes is 10, and when fourth times, the situation is even worse (see the Figure6.10).

It can be seen from the above data, although there are also the influence of system configuration or some other reasons, the main reason to impact the running speed of this tool is the load from subdivision surface algorithm. How to improve the algorithm of subdivision surface will be an important key to improve the performance of this modeling tool.

The post-processing of this modeling took the time more than seven hours, the specific consuming time for each parts of the model are recorded as follows table6.1:

6. Discussion and Conclusion

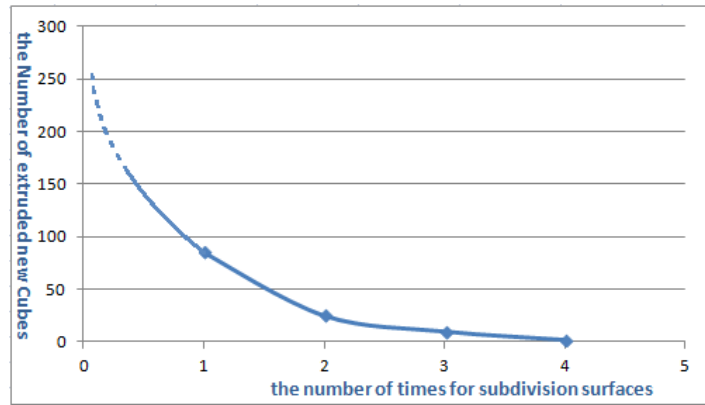


Figure 6.10.: the Threshold of Cubes with Increasing Subdivision Times

Part	Time
the cannon:	15 minutes
the prow:	1.5 hours
the roof:	40 minutes
the broadsides (x2):	more than 4 hours
the stern with the engine:	40 minutes

Table 6.1.: the Consuming Time for each Parts of the Model

The required time of each part of spaceship in the post-processing is greatly affected by the previous basic shape. For example the cannon needs only a little modifying and did not too much time, because its shape is simple and the degree of completion of its shape is already high in the main modeling phase. And with the help of other functions like Scale and Position, the process of modifying can be even more accelerated. For instance, we can select all points in a cross-section of the barrel and scale them to increase the distance between them. Then the thick nozzle of cannon and the thin intermediate barrel are distinguished obviously. But the part like the two broadsides which were almost not modified in the main modeling phase, will take a lot of time to deal with in the post-processing. It also increases the difficulty of modifying accordingly that the operable objects cannot be free to increase or to decrease in the post-process.

In addition the selecting the operable objects is also a big reason for the time consumption. Although this tool offers a variety of select model and the function of multiple selection too, it consume too much time to select operable objects one by one of mouse click. Meanwhile because of the touching and the covering between objects, it also increases the difficulties for selection of operable objects. For example, the face is causing certain hindrance to the selection of vertices. Therefore, the objects which are displayed on the edge of the model are easier to be selected than in the middle (see the Figure6.11). Sometimes, in order to better select the objects, the users have to adjust the position and angle of the view for model to move the object which will need to be selected to the edge in order to avoid the obstruction. Here, in order to improve this situation, this tool adds the option of wire frame in edit mode (see the Figure6.12). A wire-frame model is the model which is showed by using lines connecting vertices of the model. In the edit mode of wire frame, because the

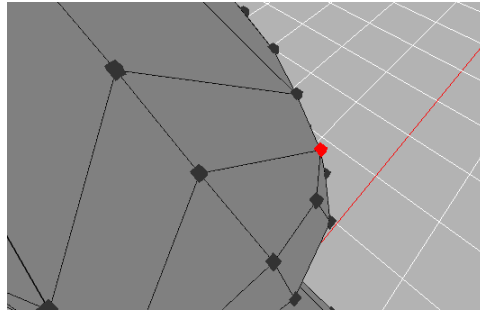


Figure 6.11.: the Problem of Selection

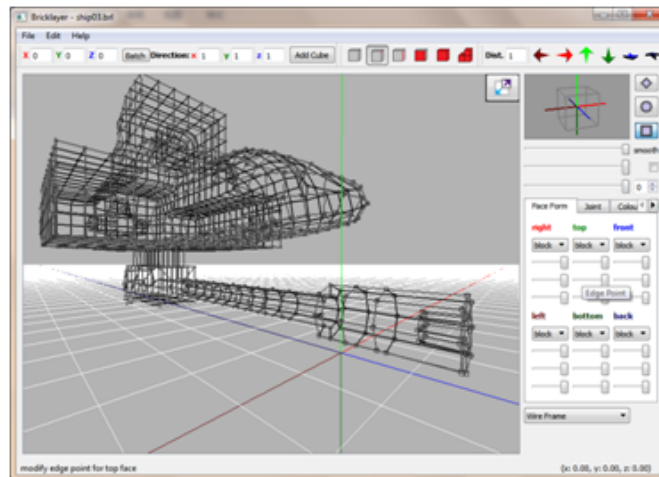


Figure 6.12.: the Model in the View Mode of Wire Frame

fillings of face are removed and the model is displayed by the form of lines, so the selection for object becomes relatively easier. Further the edit mode of wire frame also helps people to better view the structure of the entire model.

The function of subdivision surface is also provided in the post-processing. But because there are already a lot of objects in the post-processing, it is very difficult to modify objects in real time by keeping the state of surface subdivision. On the contrary, it is possible to apply the effect of subdivision surface immediately after subdivision without keeping the state of surface subdivision, and it is also possible to subdivide again and to apply the effect again.

6.2.2. Good Points

Although this is just a simple 3D modeling tool, but it has necessary functions of a complete tool from creating a new file to saving a changed file, and even file export for other applications. In theory, this tool can complete a variety of models. By the modeling example in previous section, several prominent advantages of this tool are revealed while the process of modeling.

- **simple and user-friendly interface**

6. Discussion and Conclusion

With the help of a rich variety of controls which are provided by Qt Creator, this modeling tool provides a simple user-friendly interface. The main controls all equip with simple and clear graphical icons and provide shortcuts of keyboard and corresponding descriptions about the function which are shown in the status bar. The both display windows are drawn with auxiliary lines, such as 3D coordinates and grid lines. At the same time, the two windows can also switch to each other to allow users to check the model better. In addition the Toolbars are grouped by similar functions and layered in tab form to switch conveniently of different group of tools.

- **A completely closed and pure shell model is automatically generated.**

It can be said that is the most important feature in this tool. The shell-model has been widely used in various fields, such as animation and video game. Especially the 3D printing technology which is popular today requires the printed model that must be completely closed and the pure shell-model. 3D printing, also known as additive manufacturing is one of rapid prototyping technical methods[42]. It is based on digital model file, and can quickly generate 3D solid models of almost any shape under the computer control and management. Today the widely accepted file format by the 3D printing software and 3D printers is STL which has been introduced in section 2.3.

Moreover, because the model which is generated by this tool is automatically closed, and all the faces of the model are drawn in accordance with anticlockwise, the normal mapping of the model is unified automatically. Thus if the exported model is re-imported by other software, such as Unity 3D, it can also be correctly displayed as the Figure6.9, and there is no error for display (see the Figure3.11).

- **quick and simple modeling in the main modeling phase**

The uniform modeling units which are the same size can help the user to better grasp the overall scale of the model. Especially the people who have a relatively poor sense of space can also get started quickly. And the modeling mode which is assembling model like building blocks is also more accord with people's thinking habits.

- **precise modifying of each operable object in post-processing**

Relative to the main modeling phase which receives a certain limit, this tool provides a relatively free post-processing, in order to ensure a wide variety of models which are expected by user can be established successfully.

6.2.3. Problems and Solutions

Of course, after actual modeling by using this tool, we also found a lot of flaws and problems. The following will discuss these problems and put forward improvement of method.

Overload and Slow Reaction Speed of Tool

By the analyzing in subsection 6.2.1, it can be known that the calculation amount of subdivision surfaces is the main cause to slow down response speed of this tool.

Solutions:

Except to improve the configuration of the system itself, an effective way is to improve the algorithm for subdivision surfaces. This tool uses variants of Catmull-Clark Subdivision Surface, and makes each face of the entire model to save its own parameters. On the one hand, this change can subdivide each face according to its own parameters in order to maintain the features of each face. For example the smooth faces and the pointed faces can be existed at the same time while the entire model is subdivided. But on the other hand, it also increases the amount of calculation, for instance the edge points. The original formula of edge point is the Formula4.2.

But the new formula of edge point must take different parameters of two neighboring faces into account, calculate the two edge points for the two neighboring faces respectively and find the average as the final answer. This way, the amount of calculation has doubled. If it counts the new vertex point, the amount of calculation will increase even more.

In contrast, by the introduction about factors in section 4.3, it can be seen the edge points and the new vertex point face which are generated by the faces of different forms can be represented in a uniform formula, for example an edge point with two neighboring faces of different forms:

$$\begin{aligned} V_E &= \frac{1}{2} \left[\frac{1}{2}(W + V)f_{E1} + \frac{1}{2}(V_{F1} + V_{F2})(1 - f_{E1}) + \frac{1}{2}(W + V)f_{E2} + \frac{1}{2}(V_{F1} + V_{F2})(1 - f_{E2}) \right] \\ &= \frac{1}{2} \left[\frac{1}{2}(W + V)(f_{E1} + f_{E2}) + \frac{1}{2}(V_{F1} + V_{F2})(2 - f_{E1} - f_{E2}) \right] \end{aligned} \quad (6.1)$$

- f_{E1}, f_{E2} : the factors of the two neighboring faces for the edge point

As a result, the program can reduce the repetitive calculations as long as the factors are previously saved. In the post-processing, the evaluation of edge points and the new vertex point needs also to depend on all neighboring faces' parameters.

Here it can also use similar way to improve the program by saving the respective coefficients beforehand, extracting the common divisor and unified calculating, in order to reduce the number of cycle which is increased by individual calculating in the program.

Limit and Low Accuracy of the Model in the Main Modeling Phase

The basic shape of the model in the main modeling phase affects directly the modeling work in the post-processing, so that to improve degree of completion of the model in the main modeling phase is very important.

Solutions:

In the main modeling phase, this tool provides a fixed-size modeling unit cube and the variants of Catmull-Clark Subdivision Surface to change the model into a variety of different shapes. But these are not enough. For example the case in the Figure6.13, when a larger sphere is required, currently it is difficult to accomplish the main modeling phase. The effect of subdivision surfaces is based on faces. As the Figure6.13 shown, there are the two cuboids

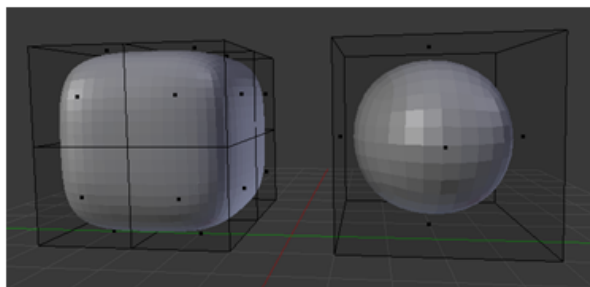


Figure 6.13.: the Comparison of two Cuboids after Subdivision

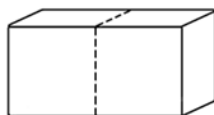


Figure 6.14.: the Entity Cube

with the same volume which the left is composed of eight small cubes and the right is a whole cube. The results of the two cuboids are completely different after subdivision.

At the moment, this modeling tool can only achieve the effect of the left. If the user wants to complete the right, must control the position of the face points, the edge points and the new vertex points by complex adjustment of factors to simulate the shape of a sphere. Or the user can scale the small shape which can be completed in the main modeling phase to a big one in the post-processing. Both methods have some complexity and uncertainty, and may not be able to achieve the perfect effect.

Here, in order to solve this problem, the concept of the entity cube which is relative to the unit cube is proposed. The cube structure which is owned by this tool currently is unit cube structure. On the basis of this unit cube structure the tool increase entity cube structure whose volume is multiple of unit cubes. As shown of the Figure6.14, the solid line represents the entity cube, and the broken line represents unit cubes. The vertices of entity cube are entity vertices; the line between both two vertices is the entity edge and the face which is constituted of entity edges is entity face.

The specific implementation process is based on the structure of the unit cube, to add parameter for judgment of entity cube in class Vertex, and to add two new class EntityEdge and EntityFace (see the Figure of class diagram6.15). Except the two endpoints the class EntityEdge also records the ID of all unit edges which are part of the entity edge. In the same way, except the entity vertices and the entity edges which are composing entity face, the class EntityFace records the ID of all unit faces which are part of the entity faces too. The list of entity vertices, entity edges and entity face are added class Mesh in the total. Thus, the jointing and deleting of the cubes remain the same and are based on the unit cubes (see the Figure6.16), but when the model are subdivided, it is based on the entity vertices, the entity edges and the entity face which are recorded in class Mesh. It should be noted that in order to ensure the calculation of face point (the midpoint of face), all the entity faces are set to maintain a rectangle to avoid irregular shapes. Therefore, after jointing and deleting the leaved irregular faces will split themselves automatically into smaller rectangular entity

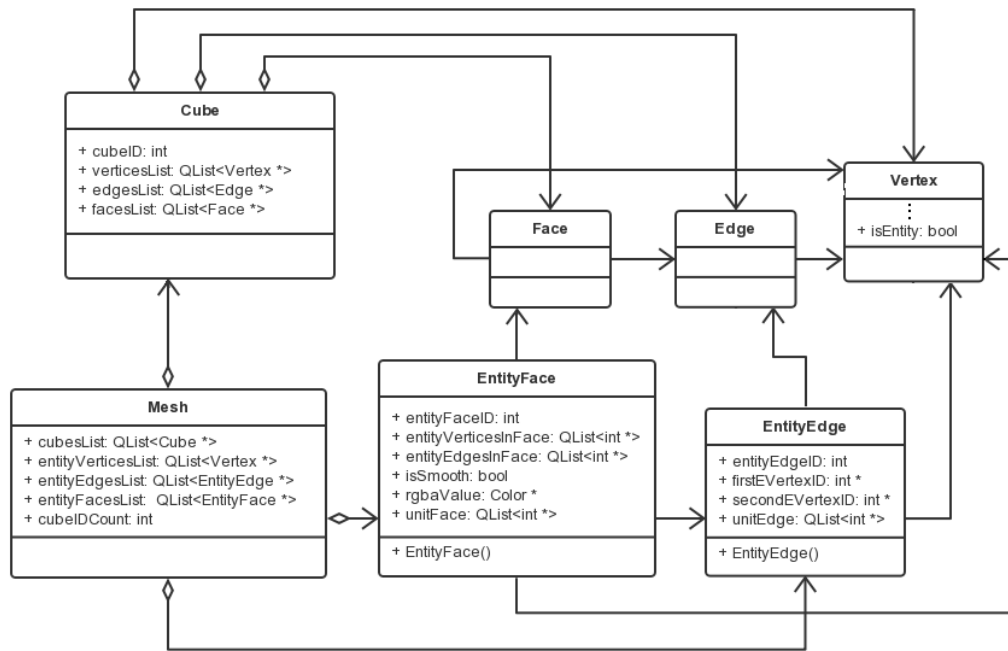


Figure 6.15.: the new Cube Structure in Class Diagram

faces. And the new entity vertices and the entity edges which are generated by the new generated entity faces are added into list of class Mesh and all the entity faces that they belongs to.

On the basis of the structure of the entity cube may also derive some other functions, for example Merge and Split which can transform edge between the unit edge and entity edge (see the Figure6.17). Besides, other derived function about scale of cube can modified the size of the entity cube and its modified magnitude remains the multiple of unit cube.

At the same time this function can also help to solute some other problems. It reduces the repeat works by adding directly a large volume of the cube. It also reduces the burden of subdivision surface and improves the response speed of the tool by removing the excess objects and subdividing only entity object. And the new cubes whose size can be modified freely will make the modeling process more intuitive and convenient.

Repeated Operation, Especially the Selection Process

The entire modeling process, there are many boring repeated operation, for example, the addition of cubes, the modifying for the edge of model, the selection operation and adjustment the angle of display in order to better select objects.

Solutions:

First, the addition of cubes in batch and the multiple selections of mouse have great help to quickly add cubes and uniformly modify the edges of model. If the entity cube which is proposed and discussed above is completed, it would further contribute to solve this problem.

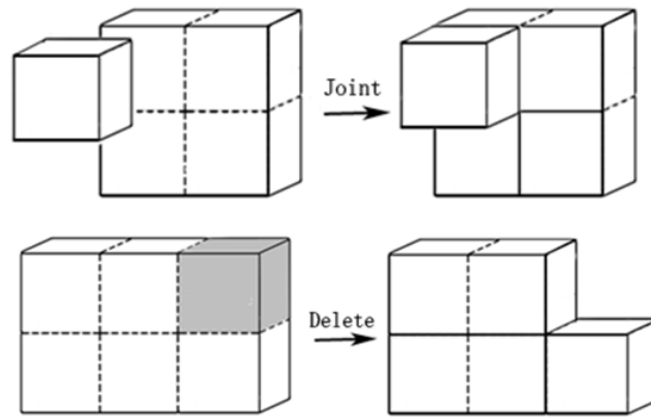


Figure 6.16.: Joint and Delete

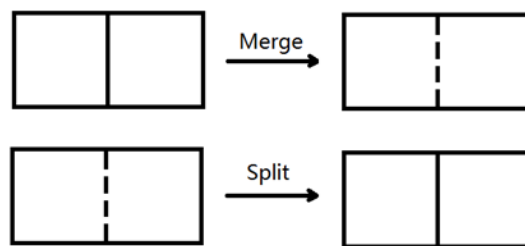


Figure 6.17.: Merge and Split

Then about the problem of selection, the most effective solution is to increase the function of Border Select. The Border Select means to draw the marquee in the display window by mouse, and the user can select all objects in the marquee or on the boundaries of marquee. The marquee is usually rectangular, and sometimes can be irregular shape that is drawn by the path of mouse's movement. In addition, to improve the display method of the model can also help to solve the problem, for example, the edit mode of wire frame which is aforementioned.

Cannot Increase Operable Objects in the Post-processing Freely

Although in the post-processing the new operable objects can be increased by subdivision surface, the subdivision of the entire model will increase too much unnecessary objects to increase the burden on the system and to reduce reaction speed of the tool.

Solutions:

The function which can only subdivide the selected faces will be programmed. Its process is similar to the Catmull-Clark Subdivision Surface (see the Figure6.18). It can to further promote this function, for example to equally subdivide edge into three and face into nine.

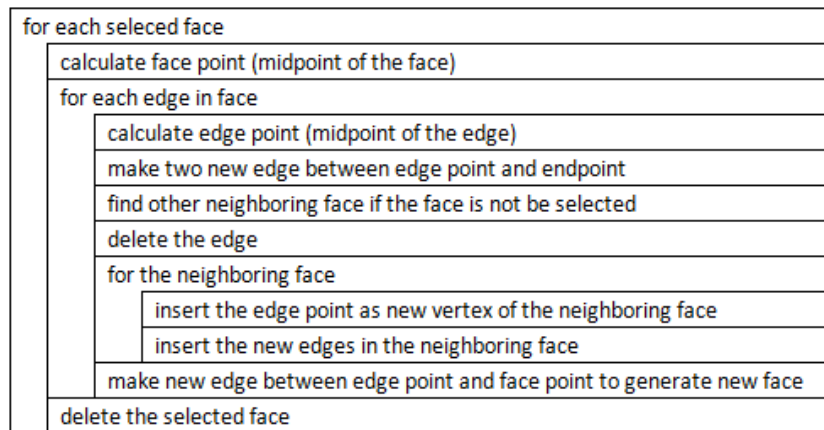


Figure 6.18.: Subdivision of the Selected Face

Cannot Undo and Redo

For each application tool about editing, the functions Undo and Redo are very important to ensure that the entire work is going on smoothly.

Solutions:

The intermediate values will be increased and saved. During the modeling process, the intermediate state of model will be saved and reappeared in order to achieve the functions Undo and Redo.

6.3. Future Work

After improving the basic modeling approach, it can add some more additional auxiliary functions for this tool, for example the functions such as Mirror, Move and Scale which play a big help while the process of the actual modeling. The similar features can be developed in the future, including Merge, Spilt and Scale of Cube which are discussed in section 6.2.3. And the functions about texture and materials should be developed too.

At present although this tool provides the function of color, but there is no way to export it. The future work will develop and improve the exported file format .mtl. Meanwhile other exported file formats should also be increased like the commonest format .fbx and the format .stl which is used for 3D printing.

In addition, there are many functions such as Move and Scale which need the help of controls to realize functional in this tool. In order to make the modeling more easily in the future, it can increase more mouse events. The functions which are realized by mouse action, shortcuts and context menu can reduce the unnecessary controls and make the user interface more concise. And this is also helpful when the tool transplant into the mobile device later.

A. Version and Configuration of System and Related Software

PC: Lenovo ThinkPad E520

- operating system: Windows 7 Home Premium
- processor: Inter(R) Core(TM) i5-2410M CPU @ 2.30GHz 2.30GHz
- RAM: 4.00GB
- system type: 64 Bit

the Programming Software:

- Qt Creator 3.0.0 based on Qt 5.2.0(MSVC 2010)
- Plugins: Doxygen 0.3.8 for program comment

the Modeling Software:

- Blender: Blender 2.70a
- SketchUp: SketchUp 14.0.4900
- Wings3D: Wings3D 1.5.3

the Game Engine:

- Unity 4.2.0f4

B. Contents of CD

1. the preparatory work: example game and example models
2. source code of 3D modeling tool and release version
3. the example model by the tool
4. screenshots for the example model
5. the paper and source code of LaTeX
6. electronic bibliographies

List of Figures

3.1.	the Screenshot of Blender with two Modeling Windows	11
3.2.	the Screenshot of Wings3D with Context-Sensitive Menu	12
3.3.	the Cylinder (left) and the Extruded Cylinder (right)	13
3.4.	the new Cone (left) and the Connected Mesh (right)	13
3.5.	Split two Faces (left) into four (right)	14
3.6.	the Smoothed Model (right)	14
3.7.	the screenshot of game Space Shooter	15
3.8.	the process of game Space Shooter	16
3.9.	the models of Player, Enemy and Boss	16
3.10.	extra white Cross-Section (left)	17
3.11.	the Models of Correct Display and Display Error	18
4.1.	the Workflow of the Tool	23
4.2.	the Screenshots of Minecraft[36]	23
4.3.	the Effect of Catmull-Clark Subdivision Surface (no subdivide, subdivide once and subdivide twice)	25
4.4.	the Effect of Subdivision Surface for Block	26
4.5.	the Effect of Subdivision Surface for Sphere	26
4.6.	the Effect of Subdivision Surface for Octahedron	27
4.7.	Join of Block and Sphere	27
4.8.	the Approximate Model and the Accurate Model of Cylinder	29
4.9.	the Batch Processing of Cube Addition	30
4.10.	the Batch Processing of Function Extruded	30
4.11.	the Edge Points or the New Vertex Points on the Same Line	30
4.12.	the Variants of Cube's Shape	31
4.13.	before Subdividing (left), Smooth Subdivision (middle) and not Smooth (right)	32
4.14.	Color and Transparency	33
4.15.	Color for Face of Cube in main Modeling Phase (left) and Color for Face in Post-processing (right)	33
4.16.	the Original Position (left) the Relative Moving (middle) and the Absolute Moving (right) along x Axis	34
4.17.	the Function Scale	34
4.18.	the Screenshot of User Interface	35
4.19.	the Toolbars of AddCube (top), SelectMode (lower left) and the Function of Move (lower right)	36
4.20.	the TabWidget for Toolbars with three Tabs	36
5.1.	the Cube Structure in Class Diagram	40
5.2.	the Calculation Process of Face Point	41

List of Figures

5.3.	the Calculation Process of Edge Point	42
5.4.	the Calculation Process of New Vertex Point	42
5.5.	the Generation of New Sub-surface	43
5.6.	the File Format OBJ	44
5.7.	Type 1 of the Original File Format	44
5.8.	Type 2 of the Original File Format	45
6.1.	the Basic Shape of Model without Subdivision Surfaces	48
6.2.	the Jagged Edges (left) and the Smooth Edges (right)	48
6.3.	the Original Catmull-Clark Subdivision Surface	49
6.4.	the Barrel (left) and the Top (right) of Spaceship	49
6.5.	the Model by Subdividing Surface Once and Twice	49
6.6.	to Transform the Model in the Post-processing	49
6.7.	the Finished Model (left) and the Model by Subdividing Surface One More Time (right)	50
6.8.	the Color Settings in the main modeling phase (left) and in the post-processing (right)	50
6.9.	the Exported Model is Re-imported in Blender (left) and Unity 3D (right) . .	51
6.10.	the Threshold of Cubes with Increasing Subdivision Times	52
6.11.	the Problem of Selection	53
6.12.	the Model in the View Mode of Wire Frame	53
6.13.	the Comparison of two Cuboids after Subdivision	56
6.14.	the Entity Cube	56
6.15.	the new Cube Structure in Class Diagram	57
6.16.	Joint and Delete	58
6.17.	Merge and Split	58
6.18.	Subdivision of the Selected Face	59

Bibliography

- [1] Trien V Do and Jong-Weon Lee. 3darmodeler: a 3d modeling system in augmented reality environment. *International Journal of Computer Systems Science and Engineering*, 4(2):145–154, 2008.
- [2] History of autodesk 3ds max. <http://area.autodesk.com/maxturns20/history>.
- [3] Jiansuo Yang, Yuansheng Liu, and Wenjuan Zhang. The method of constructing a virtual scene with 3d models in carmaker. In *International Industrial Informatics and Computer Engineering Conference*, 2015.
- [4] Sanjin Jeginovic, S Rizvic, and A Chalmers. Interactive 3d models—from 3ds max to vrml. Central European Seminar on Computer Graphics, 2004.
- [5] Sharad Mehta, Akshay Vishnoi, and Arpan Gupta. Development of a multi-component cad model of human elbow structure. In *Medical Imaging, m-Health and Emerging Communication Systems (MedCom), 2014 International Conference on*, pages 146–151. IEEE, 2014.
- [6] Johan WH Tangelder and Remco C Veltkamp. A survey of content based 3d shape retrieval methods. *Multimedia tools and applications*, 39(3):441–471, 2008.
- [7] Dariush Derakhshani and Randi L. Derakhshani. *Introducing 3ds MAX 2008*. John Wiley & Sons, 2008.
- [8] Kenneth C Finney. *Advanced 3D Game Programming All in One*. Course Technology Press, 2005.
- [9] Brenda Brathwaite and Ian Schreiber. *Challenges for Game Designers*. Charles River Media, Inc., 2008.
- [10] Bob Bates. *Game Design*. Thomson Course Technology, 2004.
- [11] Richard Rouse. *Game Design Theory and Practice*. Wordware Publishing Inc., 2000.
- [12] Obj specification. <http://www.martinreddy.net/gfx/3d/OBJ.spec>.
- [13] Wavefront OBJ. Summary from the encyclopedia of graphics file formats. <http://www.fileformat.info/format/wavefrontobj>.
- [14] Mtl specification. <http://paulbourke.net/dataformats/mtl/>.
- [15] Fbx file structure. http://wiki.blender.org/index.php/User:Mont29/Foundation/FBX_File_Structure.
- [16] Autodesk fbx. <http://www.autodesk.com/products/fbx/overview>.
- [17] Okino Computer Graphics. 3d studio .3ds geometry export. http://www.okino.com/conv/exp_3ds.htm.
- [18] Iso/pas 17506:2012, 2012. http://www.iso.org/iso/catalogue_detail.htm?csnumber=59902.

- [19] Munir Eragubi. Slicing 3d cad model in stl format and laser path generation. *International Journal of Innovation, Management and Technology*, 4(4):410–413, 2013.
- [20] Marshall Burns. *Automated Fabrication: Improving Productivity in Manufacturing*. Prentice-Hall, Inc., 1993.
- [21] Mark Summerfield. *Advanced Qt Programming: Creating Great Software with C++ and Qt 4*. Prentice Hall Press, 2010.
- [22] Jasmin Blanchette and Mark Summerfield. *C++ Gui Programming with Qt 4, Second Edition*. Prentice Hall Press, 2008.
- [23] Qt Project. Signals & slots–qtcore 5.1. <http://doc.qt.io/qt-5/signalsandslots.html>.
- [24] A Penev. Computer graphics and geometric modelling—a hybrid approach. *International Journal of Pure and Applied Mathematics*, 85(4):781–811, 2013.
- [25] Blender wiki. <http://wiki.blender.org/index.php/Extensions:Py/Scripts>.
- [26] Wings3d features. <http://www.wings3d.com/>.
- [27] Peter G Ellis, Paul A Torcellini, and Drury B Crawley. Energy design plugin: an energyplus plugin for sketchup. *National Renewable Energy Laboratory*, 2008.
- [28] What makes sketchup tick(features) ? <http://www.sketchup.com/products/sketchup-pro/features>.
- [29] Jon Brodtkin. How unity3d became a game-development beast, 2013. <http://insights.dice.com/2013/06/03/how-unity3d-become-a-game-development-beast/>.
- [30] Jeff Craighead, Jennifer Burke, and Robin Murphy. Using the unity game engine to develop sarge: a case study. *Computer*, 4552:366–372, 2007.
- [31] Unity documentation. <http://docs.unity3d.com/>.
- [32] Kenton McHenry and Peter Bajcsy. An overview of 3d data content, file formats and viewers. *National Center for Supercomputing Applications*, 1205, 2008.
- [33] Jonathan Cohen, Marc Olano, and Dinesh Manocha. Appearance-preserving simplification. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 115–122. ACM, 1998.
- [34] Alain Abran, Pierre Bourque, Robert Dupuis, and James W Moore. *Guide to the software engineering body of knowledge-SWEBOK*. IEEE Press, 2001.
- [35] Daniel Short. Teaching scientific concepts using a virtual world-minecraft. *Teaching Science-the Journal of the Australian Science Teachers Association*, 58(3):55–58, 2012.
- [36] Minecraft. <http://www.minecraft.net/classic/play>.
- [37] Edwin Catmull and James Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-aided design*, 10(6):350–355, 1978.
- [38] Jos Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 395–404. ACM, 1998.

- [39] Kenneth I Joy. Bicubic uniform b-spline surface refinement. *Visualization and Graphics Research Group, University of California*, 2000.
- [40] Wendy L Martinez. Graphical user interfaces. *Wiley Interdisciplinary Reviews: Computational Statistics*, 3(2):119–133, 2011.
- [41] Isaac Nassi and Ben Shneiderman. Flowchart techniques for structured programming. *ACM Sigplan Notices*, 8(8):12–26, 1973.
- [42] Invention Addict. The benefits of rapid prototyping, 2 2012. <http://www.inventionaddict.com/2012/02/29/the-benefits-of-rapid-prototyping/>.