

毕 业 设 计(论文)

基于Unity的三维建模软件设计与实现

题 目

|  |  |
| --- | --- |
| 院 系 | 控制与计算机工程学院计算机系 |
| 专业班级 | 软件工程专业1401班 |
| 学生姓名 | 黄彦宁 |
| 指导教师 | 王洪涛 |

二○一八年六月

基于Unity的三维建模软件设计与实现

# 摘要

近年来，随着计算机硬件水平和各种真实感图形显示技术的快速发展，对三维世界的描述变得更加方便，各个领域对计算机图形技术的要求已经不再局限于平面模型，人们开始不再满足于仅仅使用计算机显示二维的图像，而更加倾向于使用计算机呈现具有强烈真实感的三维世界。要使用计算机来呈现三维场景，必不可少的就是三维模型。

所谓三维模型，就是建立物体的三维数据及其相互关系，用以描述真实物体的三维特征，从而在计算机中进行渲染来模拟真实的三维物体。三维模型中最基本的信息就是几何和拓扑信息，几何信息主要是各点的三维坐标，拓扑信息主要是点之间的邻接关系，即由点构成的边或面，使用这些信息可以准确描述物体的外部形状和内部构造。三维建模就是创建三维模型的一种常见方式。通过三维建模，设计师可以方便的将自己的设计直观地表现出来。

本文将基于Unity引擎，设计并实现一个三维建模软件，主要完成以下工作：

1）研究三维建模相关的技术背景和研究现状，确定要采用的解决方案。

2）横向比较现有的三维建模软件，归纳出本软件需要实现的功能，设计出良好的软件架构，并确定各功能的具体需求。

3）根据预期的目标与设计，实现基于Unity的三维建模软件，阐述常用建模工具的实现与使用方法，对实现的软件进行测试，记录测试结果并进行总结。

关键词：三维模型；三维建模；Unity；计算机图形学；虚拟现实

DESIGN AND IMPLEMENTATION OF 3D MODELING SOFTWARE BASED ON UNITY

# Abstract

In recent years, with the rapid development of computer hardware and a variety of realistic graphics display technology, the description of the 3D world has become more convenient. The requirements for computer graphics in various fields are no longer limited to the plane model, and people are not satisfied with the image of the only two dimensions. It tends to use computers to present a strongly realistic three-dimensional world. To use computers to render 3D scenes, it is essential to model the 3D models.

The so-called three-dimensional model is to establish the three-dimensional data of the object and its relationship, to describe the three-dimensional features of the real object, and to render in the computer to simulate the real 3D objects. The most basic information in the three-dimensional model is geometry and topology information. The geometric information is mainly the three-dimensional coordinates of each point. The topology information is mainly the adjacency relation between points, that is, the edge or surface composed of points. Using such information can accurately describe the external shape and internal structure of the object. Three-dimensional modeling is a common way to create three-dimensional models. Through 3D modeling, designers can easily express their designs intuitively.

This paper will design and implement a 3D modeling software based on the Unity engine.

1) Study the technical background and research status of 3D modeling and identify the solutions to be adopted.

2) Compare the existing 3D modeling software horizontally, summarize the functions that the software needs to implement, design a good software architecture, and determine the specific requirements of each function.

3) According to the expected target and design, the 3D modeling software based on Unity is implemented, and the implementation and usage of commonly used modeling tools are described. Test the implemented software, record the test results and summarize.

Keywords: 3D model; 3D modeling; Unity; computer graphics; virtual reality

目 录

[摘要 I](#_Toc517513618)

[Abstract II](#_Toc517513619)

[1 绪论 1](#_Toc517513620)

[1.1 三维建模的背景及意义 1](#_Toc517513621)

[1.1.1 三维建模的背景 1](#_Toc517513622)

[1.1.2 三维建模的意义 1](#_Toc517513623)

[1.2 国内外研究现状 1](#_Toc517513624)

[1.2.1 国外研究现状 1](#_Toc517513625)

[1.2.2 国内研究现状 2](#_Toc517513626)

[1.3 课题的主要研究内容及技术路线 2](#_Toc517513627)

[1.3.1 研究的主要内容 2](#_Toc517513628)

[1.3.2 研究的技术路线 3](#_Toc517513629)

[2 三维建模概述 4](#_Toc517513630)

[2.1 三维模型的概念 4](#_Toc517513632)

[2.1.1 三维模型的含义 4](#_Toc517513633)

[2.1.2 三维模型的应用领域 4](#_Toc517513634)

[2.1.3 三维模型的表示 4](#_Toc517513635)

[2.2 不同三维建模方法的研究 5](#_Toc517513636)

[2.2.1 基于面元模型的建模方法 5](#_Toc517513637)

[2.2.2 基于体元模型的建模方法 5](#_Toc517513638)

[2.2.3 基于面体混合模型的建模方法 6](#_Toc517513639)

[2.2.4 本课题使用的建模方法 6](#_Toc517513640)

[3 系统分析与设计 7](#_Toc517513641)

[3.1 系统分析 7](#_Toc517513646)

[3.1.1 可行性分析 7](#_Toc517513647)

[3.1.2 需求分析 7](#_Toc517513648)

[3.1.3 绘制三视图 8](#_Toc517513649)

[3.1.4 辅助绘图 8](#_Toc517513650)

[3.2 系统总体设计 8](#_Toc517513651)

[3.2.1 系统架构设计 8](#_Toc517513652)

[3.2.2 系统功能模块设计 9](#_Toc517513653)

[3.3 系统详细设计 10](#_Toc517513654)

[3.3.1 相关算法 10](#_Toc517513655)

[3.3.2 数据结构设计 11](#_Toc517513656)

[3.3.3 交互设计 12](#_Toc517513657)

[4 系统的实现与测试 14](#_Toc517513658)

[4.1 开发环境 14](#_Toc517513660)

[4.1.1 Unity引擎 14](#_Toc517513661)

[4.1.2 C#语言 14](#_Toc517513662)

[4.1.3 Visual Studio 14](#_Toc517513663)

[4.2 模型控制模块 15](#_Toc517513664)

[4.2.1 线段的绘制 15](#_Toc517513665)

[4.2.2 圆弧的绘制 16](#_Toc517513666)

[4.2.3 多边形和圆的绘制 17](#_Toc517513667)

[4.2.4 柱体和锥体的绘制 18](#_Toc517513668)

[4.2.5 平移 19](#_Toc517513669)

[4.2.6 旋转 20](#_Toc517513670)

[4.2.7 缩放 21](#_Toc517513671)

[4.3 观察控制模块 21](#_Toc517513672)

[4.3.1 固定视角模式 21](#_Toc517513673)

[4.3.2 自由视角模式 22](#_Toc517513674)

[4.3.3 画面居中和模型居中 22](#_Toc517513675)

[4.3.4 平行投影和透视投影 22](#_Toc517513676)

[结论与展望 23](#_Toc517513677)

[参考文献 24](#_Toc517513678)

[致谢 25](#_Toc517513679)

# 

# 1绪论

## 三维建模的背景及意义

### 三维建模的背景

随着计算机硬件水平和计算机辅助设计技术的发展，以及计算机软件在方法学和理论体系等方面的突破性进展，人们已经不满足于仅仅使用传统的二维方式来表现物体，对于使用计算机呈现物体的三维信息有了更高的期待。

三维模型是物体的多边形表示，通常用计算机或者其它视频设备进行显示。显示的物体可以是现实世界的实体，也可以是虚构的物体。任何物理自然界存在的东西都可以用三维模型表示。

三维建模是指对三维物体建立适合计算机表示的数学模型，是在计算机环境下对三维物体进行处理、操作和分析的基础，也是在计算机中描述客观世界、建立虚拟世界等的关键技术。因此，三维建模是计算机辅助几何设计、计算机图形学、计算机动画、计算机视觉、图像处理、科学计算、虚拟现实和数字媒体创作等众多领域中都需要面对和解决的共同问题和核心技术之一。

### 三维建模的意义

建模可以以最低的成本表现设计人员所设计出来的物件的直观视觉效果，同时也有利于对有设计缺陷的物件进行修改。

传统的二维建模只能表现物体从某个方向观察得到的特征，为了完整、准确地描述一个物体需要从多个角度进行建模，其中涉及许多重复信息，尤其是在修改的时候，难以保持数据一致性。同时，二维图形不够直观，即使提供足够多的角度，也需要较强的空间想象能力才能对物体的三维结构有较好的认知。而三维模型可以更直观地表现物体的外形和结构特征，也更方便多角度、交互式地进行观察。同时，由于单个模型即可表达物体的全部几何特征，没用重复信息，因此即使频繁修改也无需担心数据一致性问题。

## 国内外研究现状

### 国外研究现状

国外目前应用较多的三维建模软件有[1]：欧特克（Autodesk）公司的3ds MAX和Maya；Multigen公司的Creator；Google公司的SketchUp；Microsoft旗下Caligari公司的trueSpace等。

同时，国外还有许多以三维模型为基础实现漫游、观察、分析、决策等的平台软件和实现某一方面功能的应用软件如数字校园、数字小区、数字城市等[2]，如：

1）美国ERDAS公司的IMAGINE Virtual GIS。该系统为三维环境下的可视化分析工具提供了GIS的功能，它能够使用户对三维图像进行实时地查询与交互操作。主要有三维可视化分析、交互可视化、威胁分析以及三维对象的链接集成、实时贯穿飞行，同时能够建立无论数据大小的虚拟世界。

2）美国Skyline软件。Skyline系列软件是基于GIS，RS，GPS和虚拟现实技术的三维可视化地理信息系统。能够利用数字正射影像、数字高程模型、矢量数据、3D模型和非空间属性数据等信息源，创建交互式的三维可视化场景；能够迅速创建、编辑、浏览、处理和分析广域范围的真实三维地表景观、建筑物景观等，并且支持大型数据库和实时信息通讯技术。

### 国内研究现状

国内尚无广泛应用的国产三维建模软件，大多数建模师主要使用3ds MAX等国外的三维建模软件。

国内比较专业的三维可视化系统软件或平台有[2]：

1）适普软件有限公司的IMAGIS Classic。该系统是一套以数字正射影像（DOM）、数字地面模型（DEM）、数字线化图（DLG）和数字栅格图（DRG）作为综合处理对象的虚拟现实GIS系统。该系统结合了三维可视化技术与虚拟现实技术，完全再现管理环境下的真实情况，把所有管理对象都置于一个真实的三维世界中。

2）灵图的VRMap。VRMap 支持OpenGL和DirectX两种国际主流的图形标准，可以充分利用计算机硬件的性能和当前最先进的图形学技术，并且具有完整的空间数据描述体系、强大的三维空间数据处理能力、三维空间分析能力和海量数据管理能力，支持大规模场景的可视化和高度的数据共享与互操作，并拥有VBA、插件和控件开发等功能，适用于各种三维应用系统的开发。

另外，国内的武汉吉奥公司的CCGIS、上海杰图三维展示系统、中视典的VRP产品体系等，在三维可视化方面都有自己独特的功能。

## 课题的主要研究内容及技术路线

### 研究的主要内容

本文在对现有三维建模技术的研究基础之上，对三维建模技术在虚拟现实中的应用、未来发展等问题进行研究，深入了解后设计并实现一个基于Unity的三维建模软件，主要实现如下功能：

1）线段和封闭平面的绘制；

2）基本三维图形的绘制；

3）平移、旋转、缩放等基本操作；

4）物体的任意角度观察；

5）指定物体绘制三视图。

### 研究的技术路线

1）搜集查阅相关文献资料，整理国内外研究三维建模技术的发展现状，了解三维建模的技术要点；

2）以Google公司的SketchUp软件为参考，深入学习其建模相关功能；

3）设计底层数据结构、接口和用户交互；

4）基于Unity进行编码；

5）测试并修复软件中存在的问题。

# 2 三维建模概述

在设计虚拟现实系统之前，首先需要创建一个虚拟环境[3,4,5]。在众多因素中，视觉关系到最直观最形象的用户体验，所以环境构建中实时地逼真地呈现出各种物体就成为一个至关重要的功能需求。

这其中，最关键的就是物体几何信息的描述，这也是三维建模[6]所需要研究的主要问题。



## 三维模型的概念

### 三维模型的含义

三维模型是用来描述物理物体的、由各种几何实体（如三角形、线、曲面等）连接的一组三维点集[7]。作为数据的集合，三维模型可以手工创建、使用算法创建（即程序化建模）或扫描创建。它们的表面可以进一步定义纹理映射。

三维模型广泛应用于任何用到三维图形和计算机辅助设计（Computer Aided Design，CAD）的任何地方[8]。甚至早在三维图形广泛用于个人计算机之前，许多计算机游戏就使用三维模型的预渲染图像作为精灵，然后计算机可以实时渲染它们。设计者可以在不同的方向和视图中看到模型，这可以帮助设计者查看创建的模型是否和他们想象中的一致。这样直观地看到的设计可以帮助设计师或者公司找出产品改进的地方。

### 三维模型的应用领域

如今，三维模型被广泛应用于各种领域[6]：医疗行业使用由MRI或CT扫描的多个二维图像切片创建的精细器官模型；电影业使用三维模型作为动画和写实电影中的角色和物体；视频游戏产业将三维模型作为计算机和视频游戏的资产；科研机构制作高精度的化合物模型；建筑业使用三维模型代替传统的物理建筑模型来展示建筑物和景观；工程界使用三维模型来进行设备、车辆和结构的设计等；地球科学界近几十年来普遍在构建三维地质模型；三维模型还是3D打印等技术的重要基础。

### 三维模型的表示

绝大多数三维模型可分为以下两种：

1）体元模型：这些模型定义了它们所代表的物体的体积，例如岩石。体元模型主要用于工程和医学模拟领域。

2）面元模型：这些模型代表表面，即物体的边界，而不是体积，例如无限薄的鸡蛋壳。几乎所有在游戏和电影中可以见到的模型都是面元模型。

面元模型和体元模型都可以创建具有相同功能的物体。不同之处主要在于创建和编辑模型的方式、不同领域对模型的使用习惯以及模型对现实物体的近似程度。

## 不同三维建模方法的研究

### 基于面元模型的建模方法

该类方法侧重于对三维物体表面信息的描述，主要有如下几种[9]：

1）边界表示法：物体的边界是物体内外部点的分界面，一般用体表、面表、环表、边表和顶点表五层来进行描述。该方法强调物体表面的细节，因而详细记录了构成物体的所有几何元素的几何信息，同时也记录了几何元素相互间的邻接关系（即拓扑信息）。几何信息与拓扑信息分开存储，数据结构清晰，数据内容完整，并且可以唯一定义物体的三维模型，但缺点是对于不规则的三维物体描述起来比较复杂。该方法主要适用于三维空间操作和分析。

2）线框表示法：用约束线来建立一系列解释图形以表达物体的边界和轮廓，实质上就是把物体上相邻的采样点或者特征点用直线相连，这些直线会形成若干个多边形，将这些多边形进行拼接形成一个多边形网格，用这个多边形网格来模拟物体的边界。该方法数据结构比较简单，数据存储量也较小，适合编程实现，但缺点是表达能力取决于线表示所能允许的复杂程度，且对同一物体的表示不是唯一的。此外，该方法也无法计算物体的几何特征。

3）小平面表示法：将物体的外表面分割成一系列微小的平面（如三角面片），同时记录该平面的特征属性。分割得到的小平面的形状和大小取决于被描述物体的形状特征和描述物体时所选取的精度。该方法既可以描述规则物体，也可以描述不规则的复杂物体，适用于多种场合，具有良好的通用性，尤其是对于含有复杂的曲面的物体，可以生成十分逼真的形态，显示效果非常好。然而，由于分割得到的小平面的数量不确定性很大，该方法所需的存储空间也具有很大的不确定性，在曲面复杂、描述精度较高的情况下数据量将变得非常大，相应的各种运算和变换也会变得相当复杂。

4）扫描表示法：将物体描述为某一个截面轮廓沿指定路径平移或绕指定轴向旋转所形成的几何体。在指定的运动过程中，截面的轨迹完整且唯一地定义了一个几何体。该方法将三维物体简化为对二维物体的理解，简化了问题的难度，但是能处理的物体种类有限，对于非规则的物体很难进行描述。

### 基于体元模型的建模方法

该类方法侧重于对三维空间体的表示，如水体、云体等，通过对体的描述实现对三维空间的表示，主要有如下几种[9]：

1）实体几何构造法：由简单的、形状规则的几何体（称为“体素”）通过正则布尔运算来构造复杂三维实体。“体素”经过某种变换（平移、旋转、缩放、组合等）后，通过正则布尔运算建立中间体，再把中间体当成“体素”进行更高层次的变换。该方法非常简单，也适合对复杂的目标采用分治算法，不存储冗余的几何信息，但缺点是不具备拓扑信息，对物体的表示也不具有唯一性。

2）八叉树表示法：类似于平面空间中的四叉树表示法，父结点表示某一块三维空间区域，8个子结点分别表示该块三维空间区域分成的8个卦限。对于每一个结点，若其代表的区域中所有体元都为均质体，则将该类型值存入结点中，否则（即为非均质的区域）将该区域进一步细分为8个卦限分别存为8个子结点，如此循环直到每个结点代表的区域都是均质的为止。该方法是一个递归的分割过程，数据结构简单，适合计算机表达，存储便捷且搜索速度快，对于布尔运算和物体几何特征的计算效率很高，但缺点是不能精确表达物体的边界，且占用的存储空间较大，难以进行几何变换。

3）空间位置枚举法：把物体所在的三维空间分割成形状相似、大小相同的单元，各单元以固定的规则网格相连，根据物体是否占据网格中的某个单元来定义物体的形状和大小。该方法很容易建立各单元的空间索引，便于进行空间搜索，且拓扑关系明确，易于进行布尔运算，但缺点是只能近似表达物体的几何信息，描述精度不高，且难以对单个物体进行坐标变换。

4）四面体格网表示法：基于对边界难以捉摸对象（如污染云）的研究，Pilouk等提出了采用不规则四面体网格（Tetrahedron Network，TEN）模型的建模技术。该技术是不规则三角网（Triangulated Irregular Network，TIN）向三维的推广，以四面体作为基本单元，将任意物体分割成一系列邻接但不交叉的不规则四面体。四面体是面数最少的体元，因而对其进行操作时所需的计算量也是最少的。使用TEN模型可以有效地进行三维插值计算和可视化，四面体之间的邻接关系还可以反映空间实体间的拓扑关系，但缺点是使用TEN模型无法描述三维连续曲面，且生成三维空间曲面非常困难。

### 基于面体混合模型的建模方法

该类方法能够充分利用各种单一的建模方法在表示不同空间实体时所具有的优点，取长补短，对复杂物体进行有效、完整的表达。但混合三维模型数据量大，且经常需要在不同的方法之间进行转换，然而不同模型之间的转换有时只能是近似的，甚至是不成立的。

### 本课题使用的建模方法

基于前述研究，结合对Unity相关API（Application Programming Interface，应用程序编程接口）文档的查阅，本文采用基于面元模型的建模方法，对于建模过程中用到的元素，如直线、圆弧、平面等，使用线框表示法进行描述，在将一系列元素整合为一个完整模型时将它们转换为使用边界表示法进行描述。

# 3 系统分析与设计



## 系统分析

### 可行性分析

1）经济可行性

本软件初期可免费使用，待用户量积累到一定程度后可采用授权使用的方式进行盈利。同时，随着用户量的上升，用户使用本软件制作的模型数量也会增长，可以配套开发一个在线的模型交易平台，官方提供一些常用的模型（视具体情况可采取付费下载的方式），同时允许授权用户经该平台自由交易自己制作的模型，平台按一定比例进行抽成。

此外，本软件开发所用到的开发工具为Unity 2017.3.0f3（Personal授权）和Microsoft Visual Studio 2017 Community。其中Unity 2017.3.0f3（Personal授权）对于年收入100000美元以下的开发者或开发者团队免费，Microsoft Visual Studio 2017 Community亦为免费软件。

从经济方面分析，本软件的开发具有可行性。

2）技术可行性

本软件基于Unity引擎[10]，使用C#语言进行脚本开发，脚本开发环境采用Microsoft Visual Studio[11]，开发完成后使用Unity提供的打包功能发布为Windows通用桌面应用程序。

在现有技术条件下，本软件各功能目的均可以达到，从技术方面具有可行性。

### 需求分析

本软件的使用对象主要是三维建模师。需求分析的方法以实地调查为主，即通过对三维建模师的访谈，同时根据三维建模师对现有三维建模软件的不满意之处，结合对相应软件的实际使用和归纳总结[19]，获取本软件各方面的功能需求。

1）本软件需实现基本二、三维图形的绘制，具体要求如下：

* 1. 指定两个端点，相连形成线段，线段共面且闭合则形成平面图形
  2. 指定两个端点和圆心角，在两端点间绘制指定的圆弧
  3. 指定中心和外接圆半径，绘制正多边形
  4. 指定圆心和半径，绘制圆
  5. 指定中心、底面正多边形外接圆半径和高，绘制正棱柱
  6. 指定中心、底面圆半径和高，绘制圆柱
  7. 指定中心、底面正多边形外接圆半径和高，绘制正棱锥
  8. 指定中心、底面圆半径和高，绘制圆锥

2）本软件需要实现模型的坐标变换，具体要求如下：

* 1. 指定位移向量，将选中模型按该向量表示的方向和距离进行平移
  2. 指定旋转中心、旋转向量和旋转角度，根据以穿过旋转中心且平行于旋转向量的直线为旋转轴，将选中模型绕该旋转轴旋转指定角度
  3. 显示选中模型的可拉伸方向，指定拉伸方向和拉伸比例（不为0），将选中模型沿拉伸方向按指定比例进行缩放

3）本软件需要实现模型的任意角度观察，具体要求如下：

* 1. 可任意调整观察位置和观察方向
  2. 可调整观察的投影方式（透视投影或平行投影）
  3. 可指定特殊的观察点（前视、左视、顶视等）
  4. 可将观察方向指向选中模型

### 绘制三视图

本软件需要实现指定模型的三视图绘制，具体要求如下：

1. 绘制指定模型指定视角的投影图
2. 仅绘制各面的轮廓，不绘制表面纹理
3. 可直接看到的轮廓和被遮挡的轮廓用不同的线型（或颜色）表示

### 辅助绘图

本软件需要实现绘图的辅助功能，具体要求如下：

1. 可自动吸附到坐标原点或附近其他图形的端点
2. 可自动吸附到三个坐标轴上
3. 可自动吸附平行于任意坐标轴的方向

## 系统总体设计

### 系统架构设计

本软件采用分层模块化的结构，如图4-1所示。

各层功能描述如下：

1）交互层提供人机交互界面，响应用户的键盘鼠标输入，并按照当前选定的功能调用下层接口

2）控制层封装下层接口的调用和Unity类库的调用，向上层提供业务接口

3）数据层存储模型的几何信息和拓扑信息，以及相机的位置、朝向等信息，向上层提供数据操作接口

UI管理器

模型控制器

相机控制器

模型数据

相机数据

绘图

观察

数据层

控制层

交互层

三维建模软件

图3-1 软件结构图

### 系统功能模块设计

1）模型控制模块

模型控制模块以模型数据为基础，根据用户的输入进行模型的创建、编辑或删除。

模型数据按顶点表、线表和面表进行存储。其中顶点表存储模型中所有顶点的坐标，线表存储每条线两个端点的坐标在顶点表中的索引，面表存储每个面上所有顶点的坐标在顶点表中的索引、该面上顶点的三角剖分结果和该面的边界顶点序列。即，顶点表中存储的是模型的几何信息，线表和面表中存储的是模型的拓扑信息。

模型控制器调用下层接口进行模型的创建和坐标变换等操作，同时调用Unity的相关接口根据更新的模型数据刷新渲染数据重新渲染模型。

2）相机控制模块

相机控制模块以相机数据为基础，根据用户的输入改变相机的位置、朝向、投影方式等参数。

相机数据存储相机的默认位置、当前位置、朝向和投影方式等信息。

相机控制器调用下层接口修改相机的当前位置和朝向等，并调用Unity相关接口重新渲染窗口。

3）人机界面模块

人机界面模块主要为用户提供不同功能的切换、精确参数设置等功能。对于鼠标输入，从相机位置向鼠标所指方向发射一条射线，若射线与模型发生碰撞，则认为输入点即为碰撞点，否则参考场景中其他辅助绘图信息推断输入点，若无参考信息则认为输入点在水平面上。对于键盘输入，提供若干快捷键和功能键进行使用，同时接收由键盘输入的精确的长度、角度和坐标等数值。

UI管理器响应用户的绘图操作和观察操作，并调用相应控制器的接口进行业务处理。

## 系统详细设计

### 相关算法

**算法0** 相同点判定（输入两点P、Q，判断是否可近似认为是同一个点）

指定阈值e2，若线段PQ的长度小于e2，则认为是同一个点

**算法1**  结点获取算法（输入点P，获取包含P的结点）

Step 1. 搜索结点列表，使用**算法0**判断是否存在包含P的结点，若存在记录其下标，若不存在将下标记为-1

Step 2. 若下标为-1，则将下标记为当前结点列表中结点的数量，然后创建包含P的新结点，将新结点加入到结点列表中

Step 3. 返回结点列表中指定下标处的元素

**算法2** 线段相交判断（输入两条线段P1P2、Q1Q2，判断两条线段是否相交）

Step 1. 若P1P2和Q1Q2的外积为0，则P1P2与Q1Q2平行，返回false

Step 2. 设以线段P1P2为对角线的矩形为R，以线段Q1Q2为对角线的矩形为T，若R和T不相交，则两线段不可能相交，返回false

Step 3. 若Q1、Q2分别在P1P2两侧（即P1Q1与P1P2的外积和P1Q2与P1P2的外积异号）且P1、P2分别在Q1Q2两侧（即Q1P1与Q1Q2的外积和Q1P2与Q1Q2的外积异号），则两线段相交，返回true，否则返回false

**算法3** 线段添加算法（输入线段PQ，修改图中相应的邻接关系）

Step 1. 使用**算法0**判断，若P、Q为同一个点，直接返回

Step 2. 建立新结点字典newNodes，其中每条记录的键表示该点在新线段上的比例，值为该位置的坐标，将newNodes初始化为{(0,P),(1,Q)}

Step 3. 建立待移除关系列表，存储需要取消邻接关系的结点对

Step 4. 建立待添加关系列表，存储需要添加邻接关系的点对

Step 5. 重置所有结点的编号和访问标记，对图进行深度优先遍历，对于每一个结点，执行Step 6

Step 6. 对于当前结点M的每一个相邻的未被访问过的结点N，使用**算法2**判断MN与PQ是否相交，若相交，执行Step 7、8

Step 7. 求出交点R，若newNodes中不包含R的记录，计算r=PR/PQ，向newNodes中添加一条记录(r,R)

Step 8. 若R在不与M或N重合，将(M,N)添加到待移除关系列表，将(M,R)和(R,N)添加到待添加关系列表

Step 9. 将newNodes按键从小到大排序得{(r0,p0),(r1,p1),…,(ri,pi),…}，依次将{(p0,p1),(p1,p2),…}添加到待添加关系列表

Step 10. 从图中删除待移除关系列表中每对结点间的边

Step 11. 对于待添加关系列表中的每对点，使用**算法1**获取对应的结点n1、n2，在n1和n2之间添加边

### 数据结构设计

建模过程中，对模型采用线框表示法，使用的数据结构为无向图[12]，存储图中所有结点的列表，对外提供AddLine和RemoveLine操作，允许向图中添加线段或移除线段。结点存储自身编号、顶点坐标、访问标记和相邻结点的引用，具体数据结构如下：

private class Node {

public int id { get; set; }

public Vector3 position { get; set; }

public bool visited { get; set; }

public List<Node> neighbours { get; private set; }

}

添加线段时，若待添加线段与已有线段相交，则在交点处把两条线段断开，将交点添加为新结点（见**算法3**）。将图中所有边使用Quad-Edge结构[13,14]进行描述，如图4-2，即可获取所有的闭合区域用于面的绘制。

对于完整模型，可将其制作成组件，采用边界表示法，存储顶点表、边表和面表。顶点表中存储所有顶点的坐标，边表存储每条边两个端点在顶点表中的索引，面表存储面上所有顶点在顶点表中的索引以及面的三角剖分结果。

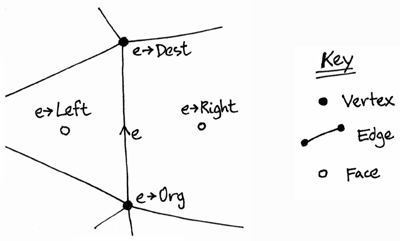
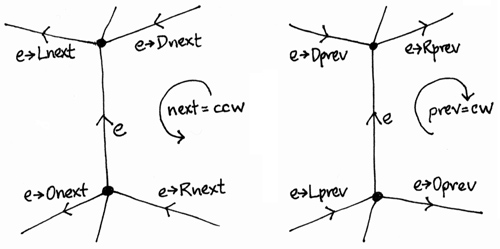
 

图3-2 Quad-Edge结构示意图

### 交互设计

本软件的交互基于Unity引擎内置的UGUI系统，对各UI组件进行组合并添加事件响应函数，同时对鼠标输入进行相应，实现人机交互。

用户界面参考市面上常见的其他三维建模软件，提供建模工具箱、属性窗口和主绘图场景，前期开发阶段各UI组件均使用默认的样式。

工具箱中各功能的切换使用单选框和文本组合实现，同一时刻只允许选中一种功能。每种功能被选中后触发相应的事件处理程序，通过调用下层接口实现功能。

属性窗口中使用文本显示属性名，使用输入框显示（或接受输入）属性值，对于有取值范围的属性支持添加滚动条实现指定范围内的快速取值，对于坐标的输入，将x、y、z三个方向的分量分别置于三个输入框内，如图4-3所示。

图3-3 坐标的输入

主绘图场景中使用红、绿、蓝三种颜色分别绘制x轴、y轴和z轴。在主绘图场景中，使用射线法来确定鼠标指向的位置，即从相机位置向鼠标所指方向发射一条射线（下称“鼠标射线”），依据如下规则推断鼠标位置：

1）若鼠标射线与某模型上某面（包含边界）相交，认为鼠标位置在交点处；

2）若鼠标射线与某模型上某顶点的距离足够小，认为鼠标位置在该顶点处；

3）若鼠标射线与原点的距离足够小，认为鼠标位置在原点处；

4）若鼠标射线与某坐标轴的距离足够小，认为鼠标位置在该坐标轴上距鼠标射线最近的点处；

5）若鼠标位置为线段的终点，且该线段与某坐标轴近似平行，认为鼠标在经过该线段起点且与该坐标轴平行的直线上距鼠标射线最近的点处；

6）若鼠标位置为线段的终点，且该线段与某参考线段近似平行/垂直，认为鼠标位置在经过该线段起点且与参考线段平行/垂直的直线上距鼠标射线最近的点处；

7）若鼠标位置为线段的终点，认为鼠标位置在经过该线段起点的水平面上；

8）认为鼠标位置在经过原点的水平面上。

以上推断规则从上到下优先级依次降低，若符合某条规则即认为该条规则推断出的位置即为鼠标位置，否则按下一条规则进行推断，直到确定鼠标位置为止。

# 4 系统的实现与测试



## 开发环境

### Unity引擎

Unity是一个支持2D和3D图形的多用途游戏引擎[15]，使用C#语言编写脚本。另两种曾经支持的脚本语言为：Boo（已于Unity 5发布时废弃）和JavaScript（于2017年8月发布Unity 2017.1后开始废弃进程）。

Unity引擎使用了如下的图形API：Windows和Xbox One上的Direct3D，Linux、Mac OS和Windows上的OpenGL，安卓和iOS上的OpenGL ES，浏览器上的WebGL和视频游戏控制台上的专有API。此外，Unity还支持iOS和Mac OS上的低级API Metal以及安卓、Linux和Windows上的低级API Vulkan。

在2D游戏中，Unity允许导入精灵并提供一个先进的2D世界渲染器。在3D游戏中，Unity允许设置纹理压缩、Mipmap和引擎支持的所有平台的分辨率，并提供对凸点映射、反射映射、视差映射、屏幕空间环境遮挡（Screen Space Ambient Occlusion，SSAO）、使用阴影映射的动态阴影、渲染至纹理和全屏后处理效果的支持。Unity还为开发人员提供如下服务：Unity Ads（广告服务），Unity Analytics（程序分析服务），Unity Certification（认证服务），Unity Cloud Build（云构建服务），Unity Everyplay（屏幕录制和分享服务），Unity IAP（苹果内购服务），Unity Multiplayer（多玩家服务），Unity Performance Reporting（性能报告服务）和Unity Collaborate（协作服务）。

Unity支持使用CG（微软高级着色语言的修改版本）创建自定义的顶点、片段（或像素）、镶嵌、计算着色器以及Unity自身的表面着色器。

### C#语言

C#是一个包含强类型、命令式、声明式、函数式、泛型、面向对象（基于类）和面向组件等多种编程规则的编程语言[16]，最早由微软在其.NET计划中开发，后来被ECMA和ISO接受为ECMA-334标准和ISO/IEC 23 70:2006标准。

C#是为公共语言基础设施（Common Language Infrastructure，CLI）设计的语言之一，是一个通用的、面向对象的语言[17]。最新版本C# 7.3已于2018年随Visual Studio v15.7.2一起发布。

### Visual Studio

Visual Studio是微软出品的集成开发环境[18]（Integrated Development Environment，IDE），可用于开发计算机程序、网站、网络应用程序、Web服务和移动应用程序等。Visual Studio使用微软的开发平台，例如：Windows API，Windows Forms，Windows Presentation Foundation，Windows Store和Microsoft Silverlight。Visual Studio既可以生成本地代码，也可以生成托管代码。

Visual Studio包含一个支持IntelliSense（代码补全组件）和代码重构功能的源代码编辑器，且其集成的调试器既是源代码级的调试器,又是机器码级的调试器。其他内置工具包括代码分析器、用于构建GUI应用程序的表单设计器、Web设计器、类设计器和数据库模式设计器等。Visual Studio还支持可以扩展几乎任意层级功能的插件，包括增加对源代码控制系统（如svn和git）的支持、增加针对特定领域语言的工具集（如编辑器和可视化设计器）或是针对软件开发生命周期中其他方面的工具集（如Team Foundation Server客户端：团队资源管理器）等。

Visual Studio支持36种不同的编程语言，并且只要存在特定语言的服务，其源代码编辑器和调试器对几乎任何编程语言都能提供不同程度的支持。内置的语言包括：C，C++，C++/CLI，Visual Basic .NET，C#，F#，JavaScript，TypeScript，XML，XSLT，HTML和CSS。其他语言如Python，Ruby,Node.js和M等的支持可以通过添加插件实现。曾经的版本还支持过Java和J#。

Visual Studio最基础的Community版本是完全免费的。最新的Visual Studio版本是2017。微软已经在2018年6月6日宣布“在接下来几个月内”将会公开Visual Studio 2019的发布时间。

## 模型控制模块

### 线段的绘制

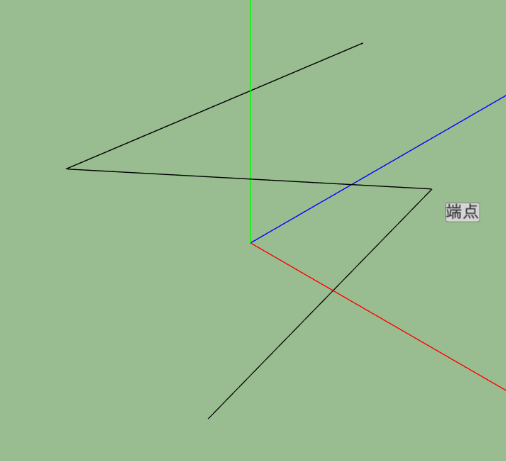
用户可以选择工具箱中的线段工具进行线段的绘制。通过鼠标单击依次确定线段的起点和终点，在两点间绘制一条线段，如图5-1所示。

图4-1 线段的绘制

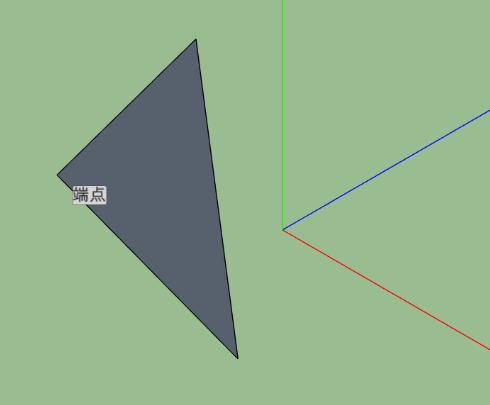
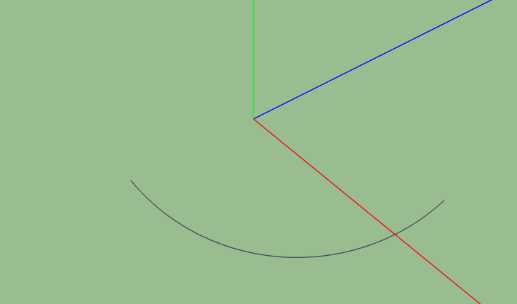
在启用线段工具后，推断系统会自动对鼠标位置进行推断，并给出相应的推断提示文字。绘制线段时，通过单击鼠标将当前推断的鼠标位置作为线段的起点，此后从起点到推断的鼠标位置处会绘制一条辅助线，用于预览将要绘制的线段。继续单击鼠标，完成本条线段的绘制，并自动以本条线段的终点为起点开始下一条线段的绘制。当绘制的线段构成闭合的平面图形时，将绘制出这个图形，如图5-2所示，同时结束本次绘制，绘制结束后将不再显示辅助线。绘制过程中如果按下键盘Esc键将取消本次绘制。

图4-2 绘制闭合的线段

### 圆弧的绘制

用户可以选择工具箱中的圆弧工具进行圆弧的绘制。通过鼠标单击依次确定圆弧的圆心、半径、起点和圆心角，绘制一条圆弧，如图5-3所示。

在启用圆弧工具后，推断系统会自动对鼠标位置进行推断，并给出相应的推断提示文字。绘制圆弧时，通过单击鼠标将当前推断的鼠标位置作为圆心，此后从圆心到推断的鼠标位置处会绘制一条辅助线，用于预览将要绘制的圆弧的半径。继续单击鼠标，将当前推断的鼠标位置作为圆弧的起点，此后从圆心向推断的鼠标位置方向会绘制一条辅助线，用于预览将要绘制的圆弧的圆心角，同时绘制另一条辅助线，用于预览将要绘制的圆弧。继续单击鼠标，完成本次绘制，绘制结束后将不再显示辅助线。绘制过程中如果按下键盘Esc键将取消本次绘制。

图4-3 圆弧的绘制

### 多边形和圆的绘制

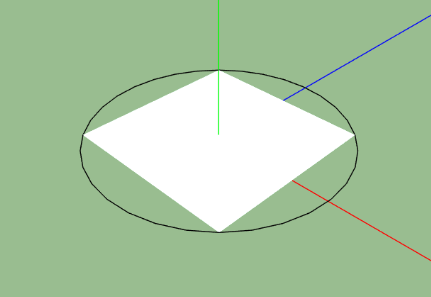
用户可以选择工具箱中的多边形工具进行多边形的绘制。通过鼠标单击依次确定多边形的中心和外接圆半径，绘制一个多边形，如图5-4所示。

图4-4 多边形的绘制

在启用多边形工具后，推断系统会自动对鼠标位置进行推断，并给出相应的推断提示文字。绘制多边形时，通过单击鼠标将当前推断的鼠标位置作为多边形的中心，此后以该点为圆心、圆心到推断的鼠标位置的距离为半径绘制辅助线，用于预览将要绘制的多边形的外接圆。继续单击鼠标，完成本次绘制，绘制结束后将不再显示辅助线。绘制过程中如果按下键盘Esc键将取消本次绘制。

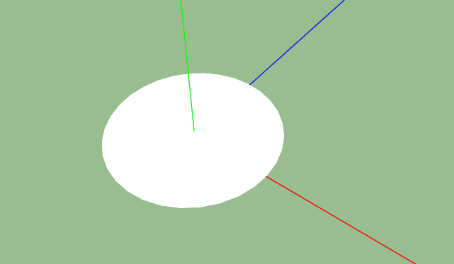
圆的绘制过程与多边形的绘制过程相似，如图5-5所示。

图4-5 圆的绘制

### 柱体和锥体的绘制

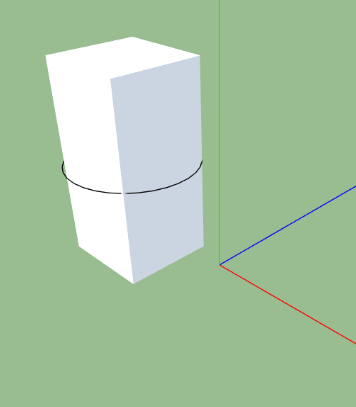
用户可以选择工具箱中的棱柱工具进行棱柱的绘制。通过鼠标单击依次确定棱柱的中心、底面外接圆半径和高，绘制一个棱柱，如图5-6所示。

图4-6 棱柱的绘制

在启用棱柱工具后，推断系统会自动对鼠标位置进行推断，并给出相应的推断提示文字。绘制棱柱时，通过单击鼠标将当前推断的鼠标位置作为棱柱的中心，此后以该点为圆心、圆心到推断的鼠标位置向量在屏幕水平方向上的分量的长度为半径绘制辅助线，用于预览将要绘制的棱柱的底面外接圆。继续单击鼠标，以中心到当前推断的鼠标位置向量在屏幕水平方向上的分量的长度为将要绘制的棱柱的底面外接圆半径，以该向量在屏幕竖直方向上的分量的长度为将要绘制的棱柱的高，完成本次绘制，绘制结束后将不再显示辅助线。绘制过程中如果按下键盘Esc键将取消本次绘制。

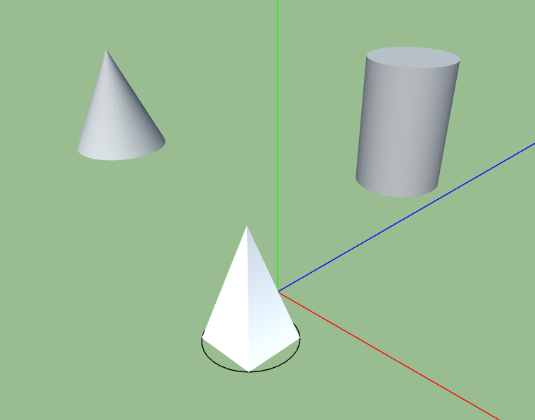
圆柱、棱锥、圆锥的绘制过程与棱柱的绘制过程相似，如图5-7所示。

图4-7 圆柱、棱锥和圆锥的绘制

### 平移

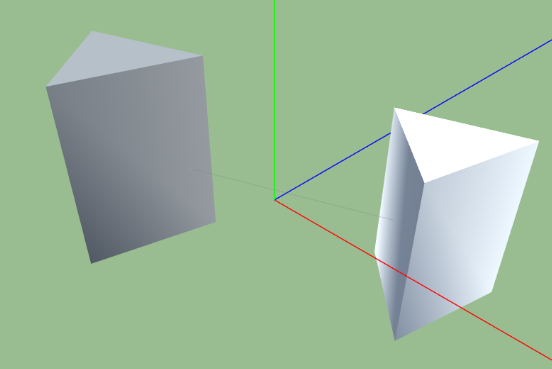
用户可以选择工具箱中的平移工具进行模型的平移。通过鼠标单击依次确定平移向量的起点和终点，实现选定模型的平移，如图5-8所示。

图4-8 平移

在启用平移工具后，推断系统会自动对鼠标位置进行推断，并给出相应的推断提示文字。平移模型时，通过单击鼠标将当前推断的鼠标位置作为起点，此后从该点到推断的鼠标位置处会绘制一条辅助线，用于预览平移的方向和距离。继续单击鼠标，将当前推断的鼠标位置作为终点，选定模型将沿着从起点到终点的向量进行平移，平移结束后将不再显示辅助线。平移过程中如果按下键盘Esc键将取消本次平移操作。

### 旋转

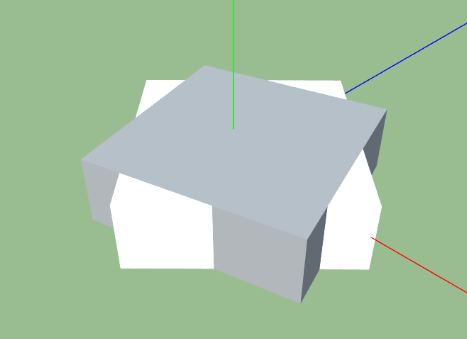
用户可以选择工具箱中的旋转工具进行模型的旋转。通过鼠标单击依次确定旋转轴和旋转角，实现选定模型的旋转，如图5-9所示。

图4-9 旋转

在启用旋转工具后，推断系统会自动对鼠标位置进行推断，并给出相应的推断提示文字。旋转模型时，通过单击鼠标确定旋转中心点，此后从该点到推断的鼠标位置处会绘制一条辅助线，用于预览旋转的起始位置。继续单击鼠标，将当前推断的鼠标位置作为旋转的起始位置，此后从旋转中心点到推断的鼠标位置处会绘制一条辅助线，用于预览旋转的结束位置。继续单击鼠标，将当前推断的鼠标位置作为旋转的结束位置，以旋转中心点、起始位置和结束位置所在的平面为旋转平面，以经过旋转中心点且垂直于旋转平面的直线为旋转轴，以起始位置、旋转中心点和结束位置的夹角为旋转角，将选定的物体绕旋转轴逆时针进行旋转，旋转结束后将不再显示辅助线。旋转过程中如果按下键盘Esc键将取消本次旋转操作。

### 缩放

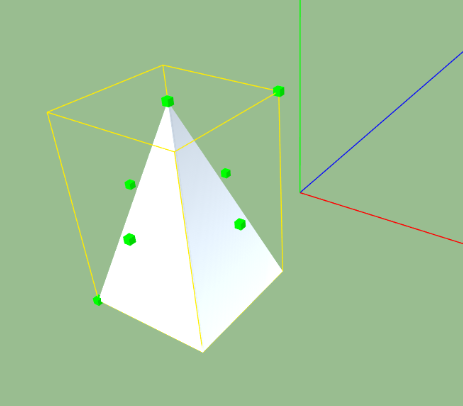
用户可以选择工具箱中的缩放工具进行模型的缩放。通过鼠标单击依次确定缩放方向和缩放比例，实现选定模型的缩放，如图5-10所示。

图4-8 缩放

在启用缩放工具后，选定模型的周围将出现一个包围盒和四对缩放手柄，位于包围盒面上的三对缩放手柄分别用于控制模型在平行于三条坐标轴方向上的缩放比例，位于包围盒顶点上的一对缩放手柄用于控制模型整体的缩放比例。缩放模型时，通过鼠标单击选中一个缩放手柄，移动鼠标位置调整缩放比例。继续单击鼠标，将选定模型沿着选中的缩放手柄所控制的方向按指定的缩放比例进行缩放。缩放默认以与选中的缩放手柄相对的缩放手柄为缩放中心，若按下键盘Ctrl键则以模型中心为缩放中心。缩放过程中如果按下键盘Esc键将取消本次缩放操作。

## 观察控制模块

### 固定视角模式

用户可以从工具箱中切换视角，使用前视、左视和顶视等固定视角观察场景。在固定视角下，除了顶点吸附（推断规则第2条）以外，鼠标位置的推断会被限制到对应的坐标平面上。用户可以平移观察位置或调整观察距离，此时仍将保持固定视角。若用户旋转观察角度，则会进入自由视角模式。

### 自由视角模式

自由视角模式下默认的观察角度称为等轴视图，在该视图下三条坐标轴间均成120°夹角。用户可在工具箱中快速切换到等轴视图。

此外，用户在自由视角下可以通过鼠标操作自由改变观察位置和观察角度：

1）通过滚动鼠标滚轮实现观察距离的调整，用向前、向后滚动滚轮来模拟观察者的前进、后退；

2）通过按住鼠标中键并移动鼠标实现观察位置的调整，用鼠标位置的上、下、左、右移动来模拟观察位置的上、下、左、右移动；

3）通过按住鼠标右键并移动鼠标实现观察角度的调整，用鼠标位置的上、下移动来模拟观察者的抬头、低头，用鼠标位置的左、右移动来模拟观察者的左转、右转。

将上述操作进行组合，即可实现任意角度观察。

在自由视角下，鼠标位置的推断将不会受限于某一特定平面。

### 画面居中和模型居中

用户可以选择工具箱中的画面居中工具和模型居中工具来快速调整观察位置和观察角度。画面居中可使相机立即朝向坐标原点，模型居中可使相机立即朝向选定模型。

### 平行投影和透视投影

平行投影是在一束平行光线照射下形成的投影。透视投影是指空间任意直线均通过一固定点(投影中心)投射到一平面(承影面)上而形成的透视关系。在平行投影下，各条线段均保持其原本的平行、垂直关系；在透视投影下，物体存在近大远小的变化，线段间原本的平行、垂直关系可能会改变。

用户可以从工具箱中切换两种投影方式。

# 结论与展望

本课题的论文中，首先对三维建模进行了介绍，随后分析了现有的三维建模软件和三维建模方法，针对虚拟现实背景下的三维设计和演示环境，提出基于Unity的三维建模软件的设计思路，并进行了系统分析。通过查阅资料进行横向比较，建立了合适的软件架构，并在此基础上实现模型的基本操作和相应的用户交互。经过测试，本软件基本实现了任务书中所提出的要求。

本文对三维建模的原理、建模方法和结构进行了深入的研究，针对模型的拓扑关系采用基于Quad-Edge数据结构的无向图，同时利用结点存储几何信息，实现了良好的数据存储结构，便于进行表面的检索和模型的几何变换。

今后的工作主要是对本软件进行改进，扩展建模功能，优化用户交互，同时提高存储和渲染效率。可进一步改进的方面举例如下：

1）本软件实现了模型几何信息和拓扑信息的创建、存储与修改，并提供了基本的变换操作。在此基础上，后续可以添加更高级的操作如实体布尔运算等，使建模过程更加友好和便捷。

2）本软件实现了对模型结构信息的处理，未对模型的外观信息（纹理、贴图、光照等）进行处理，后续可以增加对模型外观信息的处理，使模型更加逼真和美观。

3）现有的三维模型存储格式仅存储了模型渲染相关的信息，后续可扩展模型的存储格式，使得模型可以包含更多额外信息，从而可以方便地从设计图转换为施工图。

4）Unity引擎具有良好的跨平台能力，本软件基于Unity引擎进行开发，后期可以方便地迁移到虚拟现实平台，实现虚拟现实环境下的三维设计与演示场景的搭建。

# 参考文献

[1] 梁鹏帅, 冯冬敬. 三维可视化的研究现状和前景[J]. 图书情报导刊, 2009, 19(7):134-135.

[2] 徐社美, 董娜. 三维可视化建模的研究现状[J]. 中国水运(下半月), 2008, 8(9):105-107.

[3] 王晓, 李虎军, 林学华. 三维场景建模技术研究与实现[J]. 福建电脑, 2007(11):53-54.

[4] 王志杰. 基于OpenGL三维虚拟场景建模技术研究[D]. 河北工业大学, 2007.

[5] 邢姝凡, 王香茗, 王艳丽. 数字城市建设中的三维建模方法研究[J]. 测绘与空间地理信息, 2018(5).

[6] 郑佳荣, 王强, 占文锋. 三维建模方法研究现状综述[J]. 北京工业职业技术学院学报, 2013, 12(4):5-7.

[7] 黄磊, 刘卫军, 李芳. 基于AUTOCAD和3DMAX的某建筑物三维建模[J]. 科学技术创新, 2018(7).

[8] 李菊, 刘洋. 基于3DMAX和AutoCAD技术的建筑三维建模研究[J]. 自动化与仪器仪表, 2016(5):185-186.

[9] 毕硕本, 张国建, 侯荣涛,等. 三维建模技术及实现方法对比研究[J]. 武汉理工大学学报, 2010(16):26-30.

[10] https://en.wikipedia.org/wiki/Unity\_(game\_engine)

[11] https://en.wikipedia.org/wiki/Microsoft\_Visual\_Studiot

[12] 黄竞伟, 康立山, 陈毓屏. 一个新的无向图画图算法[J]. 软件学报, 2000, 11(1):138-142.

[13] Guibas L J, Stolfi J. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams[C]// ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, Usa. DBLP, 1983:221-234.

[14] Boguslawski P, Gold C. Atomic Operators for Construction and Manipulation of the Augmented Quad-Edge[C]// International Conference on Computer Information Systems and Industrial Management Applications. IEEE, 2007:125-128.

[15] Xie J. Research on key technologies base Unity3D game engine[C]// International Conference on Computer Science & Education. IEEE, 2012:695-699.

[16] Hejlsberg A, Wiltamuth S, Golde P. C# Language Specification[M]. Addison-Wesley Longman Publishing Co. Inc. 2003.

[17] Andrew Troelsen. Pro C# 2005 and the. NET 2.0 Platform[J]. 2005.

[18] Randolph N, Gardner D, Anderson C, et al. Professional Visual Studio 2010[J]. Silverlight & Beyond Xml, 2010(5):págs. 16-17.

[19] Hong D F, Yang G D, Wang Z H. Construction of Virtual Campus Based on ArcScene and SketchUp[J]. Computer Technology & Development, 2008.

# 致谢

[岁月](http://emotion.taomoo.com/rgsy" \t "_blank)如梭，白驹过隙，转眼间四年的大学生活已然接近尾声。回首往昔，挑灯夜战和奋笔疾书的日子恍如昨天，欢笑与泪水也都历历在目。作为一名华电学子，母校优良的学习风气教会了我何为务实，老师们严谨的科研态度教会了我何为求真。

在此，我尤其要感谢我的导师王洪涛老师。在本次毕业设计的实现和论文的写作过程中，王老师倾注了大量的心血，从选题到开题报告，从写作提纲到整篇论文，一遍又一遍地指出每一稿中存在的问题，严格把关，细致入微。王老师的循循善诱和谆谆教诲，使我受益匪浅。

通过本次毕业设计，我充分接触到了一个软件从前期调研、需求分析，到设计、实现，再到测试、维护的完整生命周期。我也深刻地认识到，在这个过程中，任何一个环节都马虎不得，任何一点点失误或疏忽，都需要后面的环节付出十倍百倍的代价来弥补。经过本次毕业设计的锻炼，我更加认识到自己的渺小和不足，在今后的学习生涯中，我会更加努力去夯实基础，同时更多地结合实践，知行合一，不负老师和同学们对我的期望。

毕业论文的写作是对大学四年学习和实践的整理和检验，毕业论文的顺利完成为过往的学习生活画上了圆满的句号，同时也意味着新的学习生活的即将开始。值此[毕业](http://paper.taomoo.com/wmbysj)论文完成之际，我谨向所有关心、爱护、帮助我的人们表示最诚挚的[感谢](http://paper.taomoo.com/wmgxx)与最美好的祝愿。最后，也感谢大学四年坚持不懈的自己，一分耕耘一分收获。