摘要

摘要

随着互联网的迅速发展，基于网络的商业项目的数量也越来越多，用户体 验作为帮助企业赢得市场的重要手段，地位日趋重要。网络试衣间，顾名思义就 是用户在互联网上，可以为自己的虚拟网络分身进行造型塑造、换装搭配的试衣 系统。作为一种三维虚拟技术，其拥有实时交互、视觉展现、满足用户心理需求 等方面的优点，因其构建用户体验时体现出的独特魅力，在网络商业项目中带来 了巨大的商机。本论文的研宄工作是基于三维虚拟应用技术的发展和电子商务应 用的需求，开发一个使用三维虚拟技术实现的3D试衣间系统。

本人研宄开发的虚拟现实3D试衣间系统使用跨平台的游戏开发引擎 Unity3D，结合C#程序开发工具，Maya建模工具，iClone模型动作开发工具， 实现了 3D实时试衣换装。本论文研宄包括系统总体方案设计、3D试衣间需求 分析、概要设计、详细设计、系统实现与测试等工作。系统包括了 3D模型的拆 解、加工、生成和打包、换装系统等主要功能。系统设计与开发包括分别建立人 物模型、服装模型、骨骼绑定、蒙皮、动画制作、导入Unity3D引擎、渲染、换 装等主要步骤。使用Maya建立人物模型、场景模型，模型T站姿骨骼绑定，使 用iClone 3DXchange技术和iClone技术结合为模型套用，并使用 iClone 3DXchange插件将模型和动作转成FBX格式文件，再次使用Maya进行 调整，并且导入到Unity3D引擎中。系统具有场景渲染、人物模特渲染、不同服 装发型的渲染、人物模特在场景中的行走以及换装动画、服装发型的更换等功能， 通过GUI界面的控制，使用户可以通过选择，设置出自己想要选择的模特着装 方案，并通过引擎渲染，通过Shader镜面反射渲染，得到模特背后的合理场景， 从而实现360度三维效果的呈现。

关键词：三维虚拟技术；3D试衣间；Unity3D

-I-

Abstract

With the rapid development of Internet, web-based business projects have increased in numbers and user experience has become an important means for companies to win the market. Three-dimensional virtual technology, due to it's realism and interactivity, has important significance in the building of user experience software. The Network Dressing Room happens to be such a highly interactive user experience software. It can help users preview clothing and different outfit pairings in a practical and fun way, thus providing insightful shopping advice. The application of three-dimensional virtual technology to this setting allows for imagination to become reality. One would be able to closely examine all 360 degrees of an apparel without even having to leave the comforts of home. Therefore, building a 3D online virtual fitting room using 3D technology would be an expedient research project.

Numerous investigations and research has led to the decision to use Unity3D in combination with C# technology to achieve the 3D fitting room system. This project includes scene rendering, model characters rendering, different clothing and hairstyles, as well as walking animation. The replacement of clothing remains the main logic of the system. On the basis of Unity3D, MaYa must also be used to buid character models and scene models. iClone technology will reccord movements for the models. iClone 3DXchange will then combine the models and movements together and plug them into the Unity3D engine for final editing and packaging. Through Ui interface control and Shader control users will be able to see in 3D effects the character models wearing the clothes they intend to buy, thus achieving the 3D fitting room system functions.

Keywords: three-dimensional virtual technology, 3D dressing room, Unity3D

-III-

第1章绪论

3D试衣间可以在网购服装或者游戏角色选装时，为用户提供服装、装备效 果预览，为其设计网络分身、选择服饰装备提供服务，具有很强的互动性和实用 性。在传统网络游戏以及页游、手游竞争越来越激烈的今天，用户体验决定了项 目成败，谁能提供更优质的画面效果，给用户更全面、体贴的服务功能，谁就更 具吸引力，占据竞争的优势，因此开发功能强大的3D试衣间，具有很强的实用 性和广泛的前景。本课题旨在研宄开发利用MAYA、iClone等三维制作软件，以 及Unity3D引擎强大的三维、跨平台功能，使用编程语言可以实现虚拟现实的 3D试衣间系统

1.1选题背景及意义

网络试衣间是一种可供用户对自己的虚拟分身进行选择、设计，对服饰、装 备进行自由搭配的展示系统。这种供用户自由换装的理念，最早出现于十几年前 网络刚开始流行的时候，当用户己不满足于仅靠文字ID或上传图片头像来代表 自己的方式时，提供虚拟分身，自由挑选、任意更换造型的创意初现，并为大型 多人在线角色扮演游戏提供了商机，网络使用者为了使自己在造型、穿着上与大 多数人不同，并为游戏的“2D纸娃娃”系统付费，购买服饰配件以使自己区别 于其他玩家。在纸娃娃系统中，角色的每个动作，每套服饰、武器，都只有独立 的一套，当玩家选择了不同的装备时，引擎可以根据其不同的需要，临时将几组 装备图像拼装到一起，进而形成变化丰富的角色外观。

近年来，随着虚拟现实技术的出现及发展，取代传统2D换装系统的3D虚 拟现实技术己趋于成熟，这成为本选题的技术背景。虚拟现实（Virtual Reality) 技术是用计算机模拟出生活中感官所能感受到的世界，利用三维图像技术生成视 觉画面、多媒体技术制造声音，多传感交互技术模拟触觉，生成逼真的三维虚拟 环境，令人如同身临其境一般，是一种由计算机技术辅助生成的高级技术模拟系 统。虚拟现实技术在近年随着3D商业项目的普及得以迅猛发展，从模型的创建 技术到骨骼绑定技术、动作模拟技术、3D游戏引擎技术方面都有了重大突破。 虚拟现实技术从过去大多应用于公司、企业大型商业项目等领域的专业技术，逐 渐转型为业余爱好者、个人也可以学习并用以完成个人作品的普遍技术。在这样 的背景下，虚拟现实三维技术提供了本论文研宄所需的相关技术支持，使3D试 衣间换装系统的研宄得以实现。

如今的网络游戏分为收费游戏和免费游戏，所谓的免费游戏是指游戏时间免

费，但其装备、外装等增值服务会直接收费的运营模式。外装销售所获利润在免

-1- 费游戏盈利中占很大比例，这些游戏里为合成一件“拉风”装备投资上万甚至数 十万人民币的玩家屡见不鲜，3D试衣间可以让用户直观的看到服装与装备的外 观效果，通过换装功能比较发现“高档”外装与普通免费装备的区别，以触发其 购买的欲望。而在收费游戏，即点卡游戏中，游戏运营商往往也会提供给用户大 量装备、外装，造型美观的高档装备会激起玩家多次“刷副本”的欲望，从而增 加游戏时间，消耗更多点卡，3D试衣间的换装功能，可以支持玩家试穿高档装 备，自由设计搭配，激发其拥有高档装备的欲望，从而推动点卡的消费。在这种 市场背景下，研宄3D试衣间技术是很有现实意义，具有巨大商业前景的。强烈 的视觉冲击力使用户更容易激发出购买欲，从而使3D试衣间的商业价值得以体 现。

在这样的技术及市场背景下，本论文研宄的3D试衣间换装系统具有巨大 的商业价值，以及指导用户选装的现实意义，因此本论文选择设计实现网络3D试 衣间作为选题。

1.2国内外的研究现状 1.2.1游戏领域的发展历史及现状

网络试衣间在游戏领域最具商业价值的部分，在于为游戏角色换装的功能。 网络游戏游戏中出现的试衣间换装系统分为2D和3D两种。

最早的2D游戏角色换装功能，出现于1996年5月由Origin公司以其著名 游戏创世纪（Ultima)系列产品为背景所创造出来的世界上第一款图形大型多人 在线角色扮演游戏《网络创世纪（UO)》的“2D纸娃娃”系统中。美工将角 色各部分细分，服装、饰品、武器都绘制出独立的一套，当选择了不同款式的部 件，引擎就会将所选部件拼装在角色身上，以达到换装效果。在此之后，2D游 戏大多采用“2D纸娃娃”技术来实现角色的换装功能。如日本ENIX公司开发 的2DQ版游戏《魔力宝贝》，头饰系统就采用了 “2D纸娃娃”技术。腾讯的 QQ秀试衣间，为QQ用户的虚拟2D模特细分各个部件，分别绘制多款发型、 头饰、五官、服装供用户选择，当用户切换不同的部件时，引擎将其组合起来， 生成预览图，该试衣间系统为用户购买QQ秀提供了方便。QQ秀商城的成功运 作，为腾讯带来了巨大的经济利益，很多用户愿意为拥有与众不同的QQ秀而支 付年费购买红钻会员服务。

近年来随着大型3D网络游戏的盛行，3D试衣间功能也出现在众多火爆的 网络游戏中，例如《魔兽世界》、《剑网3》等游戏中，都出现了试衣间插件， 这种插件链接了游戏内部的数据库，当用户点击装备的名称或者装备图标时，即

-2- 可在弹出窗口中看到自己的虚拟角色穿上该装备的形象，华丽的装备可以刺激玩 家的购买欲望，使其花费更多的点卡为获得该装备而努力。这种3D试衣间需要 有网游内部数据库作支持，每个换装部件都需要单独建模，用户做出不同选择时 插件可以调用并可渲染各部件，整合到游戏角色身上，这种3D试衣间往往角色 模型静止不动，换装时也没有动作和神态变化。

综上所述，无论在2D还是3D游戏中，试衣间的换装功能都可供用户进行 装备预览，为其搜集个性服装进行指导，具有巨大的商业价值。

1.2.2服装领域的发展历史及现状

网络试衣间在服装展示方面的概念形成于2001年德国开发的第一个试衣间 系统，它在葡萄牙的一场时装秀中亮相，是由德国弗劳恩霍夫学会的科学家与其 他科研小组共同开发而成的。该系统需要较繁琐的步骤才可以达到效果，首先用 户需提供自身的三维影像，这一步需通过手持的三维扫描仪进行扫描完成，之后 将扫描结果传输给该试衣间系统，用户就可以按照系统中录入的服装目录，选取 各种服饰进行试穿，还可以通过鼠标控制虚拟影像进行简单的举手弯腰等动作， 查看衣服是否合适，此外，顾客甚至可以调控光线等参数进一步观看服装的整体 效果。这套系统在葡萄牙亮相后，很快在网上流行起来。

之后在美国和日本，也陆续出现了相关技术。2006年美国的H&M服装连锁 店开发完成了可以应用于网络的试衣间系统，用户可以通过登陆H&M官方美国 网站，选择试衣间功能。用户可以选择一个与自己体型相似的模特，对皮肤颜色， 发型，身体特征，高度，体重，以及眼睛颜色，鼻子，嘴唇进行选择。选好后就 可以用所有在H&M销售的服装包括泳装、成衣和小装饰品为你的模特进行设 计。当消费者完成满意的组合，存储进入“你的衣柜”，就可以将衣柜中的组合 打印出来，带到商场去购买。

2007年年底，网络试衣间的雏形在中国出现并逐渐流行起来。但由于技术和 资金上的瓶颈，还没有比较完美的作品形成，大多是Flash制作的半成品，使用 平面的服装照片在平面的模特照片身上进行替换，没有立体感，制作较为粗糙， 有的试衣间甚至让人感觉模特的头发像没有用Pfctosfcp处理好就硬贴上去一 样。如何将网络试衣间制作成三维立体，可视性以及功能性更强的产品还有待研 宄，这一片领域上的空白与中国服装网购市场的巨大需求形成强烈对比，很多用 户都期待功能完善、用户体验优秀的网络试衣间问世，这使得对网络试衣间的研 宄在服装展示方向具有很大的市场价值。

-3-

1.3与以往试衣间系统的不同之处

1、 本选题采用虚拟三维技术制作的3D试衣间与以往的2D试衣间的不同点在 于：2D试衣间的虚拟模特、服装、道具都是绘制出来的二维图片，换装的时候 模特是静止的，如果要制作会动的二维模特，不但模特的每个动作需分解重绘， 而且所有供换装的服饰、道具都需要配合分解的动作进行绘制，例如：有100套 换装道具，就需要绘制100次不同着装的模特动作流程，工作量巨大。而3D试 衣间模特以及换装用服饰、道具都是与骨骼绑定的，只需要建立各自的网格模型， 动画进行时骨骼会带动换装的网格模型一起运动，100套换装道具也只对应一套 动作骨骼，减轻了工作量。

2、 而本选题与《魔兽世界》、《剑网3》等3D游戏的试衣插件不同点在于：

(1) 以往的3D试衣系统是用没有骨骼动画的3D模特进行展示，而本论文研宄 的网络3D试衣间的模特是带有骨骼动画的，不单纯显示3D模型换装后的静态 效果，模特在换装的同时会带入表情与肢体动作，更加生动富于趣味性。

(2) 以往的3D试衣系统中，模特头、身体、四肢的网格是一个整体，只有换 装的装备、道具进行细分建模，这种方式只能提供装备道具的着装预览，而无法 改变模特发型、面容。而本论文研宄的3D试衣间系统中，模特自身也进行了网 格细分，头发、脸、身体、四肢分别建模，使得用户不止可以选择装备、道具， 同时可以对发型、眼球、脸型进行挑选。

(3) 以往的3D试衣插件直接调用游戏数据库中的建模资源进行装备替换，不 需要对模型数据进行处理。而本选题需在模型、动画FBX文件导入引擎平台时， 将其含有模型网格、贴图、骨骼、动画等信息转化处理为可调用的数据，按照一 定的命名规则存储于资源包中。

3、 本选题研宄的3D试衣间与现实服装展示网络试衣间的区别在于：服装展示 网络试衣间是2D平面的，模特和服装的立体阴影都是绘制或者拍照得到的，无 法进行三维展示，换装没有动画播放只是生硬的服装套用，用户体验较差。而本 选题所研宄的3D试衣间提供了制作三维服装展示的换装方法，使用这套方法只

-4-

需在模型制作和贴图制作上多投入工作时间，就可制作出商用三维服装展示试衣 间，对服装展示类网络试衣间在三维领域的发展提供了技术支持。

4、本选题研宄的为3D模特制作动画的方法，是利用Maya模型制作工具进行建 模并且绑定骨骼，然后使用lClone 3DXchange以及lClone技术，进行动画制作， 这种工作流程与以往的Maya建模、绑定骨骼、制作动画不同，省去了 IK、FK 控制器、线控等相对复杂的骨骼绑定过程，利用iClone技术可更为直观的进行 动画制作，使制作动画的门槛降低、效率提高。同时本论文研宄了如何用Maya 工具处理由iClone制作的动画，使其成为Unity3D引擎可用的素材。

1.4主要研究内容

论文研宄工作旨在研宄开发一个使用三维虚拟技术实现的3D试衣间系统。 具体研宄工作包括：

1、 通过对现有虚拟现实技术的研宄分析，提出虚拟现实3D试衣间的总体设 计方案，包括Unity3D引擎的选择、建立人物模型的方法、骨骼动画的制作方法、 模型和动画导入Unity3D引擎的方法等；

2、 3D试衣间系统需求分析。通过对试衣间需要解决的问题的分析，提出 3D试衣间系统的功能，并给出用例模型、系统流程等需求模型进行描述；

3、 3D试衣间系统概要设计。包括系统体系结构描述，提出3D试衣间系统 的功能模块划分，给出系统功能结构图、系统分层结构图，给出系统文件管理器 设计描述，包括主要库表的定义、数据字典的描述等；

4、 3D试衣间系统详细设计。给出系统主要模块的算法描述，包括程序流 程图等。描述使用MAYA建立人物模型、场景模型、模型骨骼绑定的流程；使 用iClone3 DXchange技术和iClone技术结合为模型套用并且调整骨骼动画的方 法；使用iClone 3DXchange插件将模型和动画转成FBX格式文件的技巧；以及 将所需素材导入到Unity3D引擎中的方法。描述场景渲染、人物模特渲染、不同 服装发型渲染的实现方法以及人物模特在场景中的动画播放、服装发型更换、360 度三维预览等功能的实现方法。

5、 3D试衣间系统的功能测试。

-5-

1.5论文结构概述

第1章是对选择本论文题目的研宄意义的阐述；国内外相关技术的发展状况; 本人在此次论文中的研宄内容。

第2章将会具体介绍并分析相关的技术特点，其中包括用到了哪些技术，该 技术的特点，选择的理由。

第3章是对总体需求进-行分析与阐述，分析试衣间系统换装功能得以实现所 需要具备的数据基础、技术基础，试衣间的最终实现目标，用户需求，功能需求 等内容。

第4章是试衣间的总体设计。根据需求，阐述设计思想；人物模特模型、骨 骼动画的建立；导入引擎、读取模型资料生成文件管理器资源集合、渲染生成角 色、调用文件管理器中的数据进行换装、存储换装数据的方法。

第5章是试衣间的实现及试衣间的测试。重点阐述实现各个功能模块的实现 方法、流程、部分关键代码、截图效果说明。最后通过整体效果阐述效果分析结 果，并进行测试。

最后一部分是致谢。感谢导师在论文的修改中对我的帮助，感谢老师们在技 术上为我传授的知识。

-6-

第2章3D试衣间总体需求分析

本论文研宄的3D试衣间所涉及的技术面较广，需要对3D建模、骨骼动画、 游戏引擎、程序制作等方面工作进行综合运用，需要分析制定一套可行的开发计 划。这就需要分析3D试衣间的总体需求，分析功能实现所需要解决的问题，因 此总体需求分析对项目进行具有决策性、指导性，在系统开发过程中具有举足轻 重的地位。本章将详细介绍3D试衣间需求分析阶段的工作。

2.1 3D试衣间需求分析

2.1.1模型基本要素需求分析

本论文研宄的3D试衣间设有男、女模特各一名，这两名3D模特的模型、动 画FBX文件，提供了 3D试衣间换装所需的所有基本要素。这些基本要素由以 下几部分内容组成：

1、 3D模特需具有可供用户选择替换的部件，其模型应被细分为若干个部 件并分别建模。如将模特细分为：面部、眼睛、头发、上衣、裤子、鞋子6个 部件。这些部件的模型网格组合在一起时可形成完整的3D模特，不同的部件又 可建立外形不同的模型网格，形成多种换装方案供用户选择。例如：不同外形 的上衣1、上衣2。为了在模型FBX文件导入引擎时，其包含的部件信息可以被 读取并存储，所有方案包含的部件必须全部置入模型文件FBX中。

2、 部件替换结束时，为使换装过程更自然，需要播放一段动画以示换装完 成，因此3D模特文件需包含与部件分类相对应的骨骼动画。如：更换上衣、裤 子、鞋子后，模特分别播放看向新衣服、裤子、鞋子的动画。

3、 替换部件后，新的部件必须可以跟随整个身体进行动画位移，因此需要 将不同方案的全部部件与整体模型绑定于同一套骨骼上。

4、 不同方案的模型网格，分别可具有多套贴图，并按分类存储于工程文件 夹中，以便组合出更多可供用户选择的方案。例如：不同外形的上衣1、上衣2 可以与不同花样的贴图1、贴图2，组成上衣1\_贴图1、上衣1\_贴图2、上衣2\_ 贴图1、上衣2\_贴图2，四种方案供用户选择。

5、 3D试衣间具有性别选择功能，因此需要制作一男一女两套方案的FBX 模型文件，两套文件都需要包含各自不同的一套骨骼、模型网格、贴图、动画

-7-

第2章3D试衣间总体需求分析

本论文研宄的3D试衣间所涉及的技术面较广，需要对3D建模、骨骼动画、 游戏引擎、程序制作等方面工作进行综合运用，需要分析制定一套可行的开发计 划。这就需要分析3D试衣间的总体需求，分析功能实现所需要解决的问题，因 此总体需求分析对项目进行具有决策性、指导性，在系统开发过程中具有举足轻 重的地位。本章将详细介绍3D试衣间需求分析阶段的工作。

2.1 3D试衣间需求分析

2.1.1模型基本要素需求分析

本论文研宄的3D试衣间设有男、女模特各一名，这两名3D模特的模型、动 画FBX文件，提供了 3D试衣间换装所需的所有基本要素。这些基本要素由以 下几部分内容组成：

1、 3D模特需具有可供用户选择替换的部件，其模型应被细分为若干个部 件并分别建模。如将模特细分为：面部、眼睛、头发、上衣、裤子、鞋子6个 部件。这些部件的模型网格组合在一起时可形成完整的3D模特，不同的部件又 可建立外形不同的模型网格，形成多种换装方案供用户选择。例如：不同外形 的上衣1、上衣2。为了在模型FBX文件导入引擎时，其包含的部件信息可以被 读取并存储，所有方案包含的部件必须全部置入模型文件FBX中。

2、 部件替换结束时，为使换装过程更自然，需要播放一段动画以示换装完 成，因此3D模特文件需包含与部件分类相对应的骨骼动画。如：更换上衣、裤 子、鞋子后，模特分别播放看向新衣服、裤子、鞋子的动画。

3、 替换部件后，新的部件必须可以跟随整个身体进行动画位移，因此需要 将不同方案的全部部件与整体模型绑定于同一套骨骼上。

4、 不同方案的模型网格，分别可具有多套贴图，并按分类存储于工程文件 夹中，以便组合出更多可供用户选择的方案。例如：不同外形的上衣1、上衣2 可以与不同花样的贴图1、贴图2，组成上衣1\_贴图1、上衣1\_贴图2、上衣2\_ 贴图1、上衣2\_贴图2，四种方案供用户选择。

5、 3D试衣间具有性别选择功能，因此需要制作一男一女两套方案的FBX 模型文件，两套文件都需要包含各自不同的一套骨骼、模型网格、贴图、动画

-7- 2.1.2 3D试衣间所需的基本数据分析

将FBX格式的模型文件导入Unity3D引擎后，为了可以调用其包含的模型 网格、动画、骨骼、贴图等信息，需要将其处理、存储为可被换装程序识别使 用的基本数据资源，为换装功能奠定基础。换装功能的实现需要收集和存储以 下几种类型的基本数据资源：

(1) 材质资源：换装方案在FBX文件中是由模型网格和贴图来体现的， 在导入Unity3D引擎后，需将它们转化为可调用的材质资源才能被程序使用。 提取所有模型网格的名称，用数据字典进行记录，创建模型网格资源包，例如 female\_top-1.assetbundle;然后提取贴图名称，使之与对应的部件模型网格名称 进行组合，用数据字典进行记录，创建模型网格贴图资源包，例如 female\_female\_top1.assetbundle。这些材质资源包中包含生成角色所需的数据资 源，当组装角色时将被调用。

(2) 骨骼信息资源：模型网格在FBX文件中是通过蒙皮与骨骼建立联系 的，而当它与贴图组成可用于Unity3D的材质时，这种联系也将不复存在。因 此，为了在调用材质时可以映射它们对应的骨骼，需要创建数据字典记录骨骼 名保存于资源包中，用以维系材质与骨骼之间的联系。

(3) 动画数据资源：3D模特入场时会播放行走动画，停止时会播放闲置 状态动画，更换上衣、裤子、鞋子时则播放相应的换装动画，这些动画信息存 在于动画文件中，为了适时调用，需要提取并存储于资源包中。

2.1.3 3D试衣间开发平台需求分析

实现3D试衣间需要有一个可以渲染3D模型、动画，可以塑造较好的空间 感、光感，并且支持换装功能文件管理器数据集合开发的游戏引擎。由于Unity3D 引擎具有性价比的绝对优势、先进易用性、兼容开放的跨平台性、高质量的3D 画面效果等多方面优势，本课题选择了 Unity3D作为项目开发平台。希望可以利 用到它强大的虚拟现实功能，使试衣间的画面质量得以优质的呈现，并利用其方 便、功能强大的开发平台，使整个开发项目顺利进行。

Unity是由Unity Technologies开发的一个让用户轻松创建诸如三维视频游 戏、建筑可视化、实时三维动画等类型互动内容的多平台的综合型游戏开发工具， 是一个全面整合的专业游戏引擎。[1】Unity类似于Director、Blender game engine、 Virtools或Torque Game Builder等利用交互的图型化开发环境为首要方式的软 件其编辑器运行在Windows和Mac OS X下，可发布游戏至Windows、Mac、 Wii、iPhone、Windows phone 8 和 Android 平台。【2】也可以利用 Unity web player

-8- 插件发布网页游戏，支持Mac和Windows的网页浏览，它的网页播放器也被Mac widgets所支持。【3】

Unity3D引擎在以下六个方面具有非常显著的优势，为系统开发提供了优良 的环境，非常适合本选题所需。

(1) 优质画面：其制作完成的游戏，画质可以与虚幻三引擎制作的游戏相 媲美，强大的图形处理能力可以满足大部分游戏的开发需求【4】，这使得3D试 衣间的模特、场景、动画、服装的画质得以保障。Unity3D的渲染管道高效，效 果逼真，着色器系统灵活高性能，提高了引擎能带给用户的整体体验与感受。其 出众的3D渲染能力不逊于虚幻三引擎，而又远超PaperVison3D等引擎。比起现 今开发网络试衣间的主流2D平面软件来，更是具有不可比拟的优势。Unity3D 引擎还可以通过操作其界面导入如今业内流行的大多格式素材文件，包括其它软 件制作的模型、贴图、动画、音频等原始素材，这就使开发3D试衣间系统成为 可能，制作的模型以及相关素材都可以在引擎中使用【5】。这些素材都可以安置 在Unity3D的文件夹资源库中，当需要使用它们时，可以通过Unity3D的主操作 窗口轻松的将资源库中的内容拖入虚拟出来的三维空间内。在这个空间内，可以 组建3D试衣间场景、角色等元素，以创建虚拟模特在试衣间内行走、换装时的 三维动画等三维交互工程。用Unity3D来制作3D试衣间在三维虚拟和画面质量 上将会更容易满足用户对视觉感官上的需求。

(2) 平易的开发环境：Unity3D的开发环境既可以是Windows又可以是 Mac OS X，比较有利于学习与研宄，只需要安装一款Unity3D软件，就可以在 主流系统下开发。而Unity3D的免费版本也可以制作完整的游戏，和收费的专业 版主要区别在于画面滤镜和底层类库的使用权限，因此不需要投入成本就可以进 行开发，对于学习研宄非常有利。与虚幻三等动辄几百万的引擎相比，其价格的 优势显而易见，Unity提供了基础版和专业版两个版本，基础版是完全免费的， 只有针对Flash、IOS、Android等平台收取授权费用。因此从性价比上来看，选 择Unity3D显然更为合适

(3) 非常强大的跨平台性：既然是网络3D试衣间，就必须能发布于网络，而 Unity3D强大的跨平台功能可以符合本选题需要，其用于发布web的插件Unity web player既可以发布支持运行于Windows系统的网页程序，又可发布运行于 Mac 的网页程序，该插件支持 IE、Firefox、Safari、Mozilla、Netscape、Opera 和Camino、Mac OS X的Dashboard工具，几乎所有主流web浏览器，因此选择 Unity3D引擎，就可以解决网络发布的问题【6】。并且Unity3D支持Window、Mac OS X、Web browsers、iOS、Android、PlayStation3、Xbox360、Windows Store、

-9-

Windows Phone、Linux、Blackberry 10、Wii U、PlayStation Vita、PlayStation Mobile、 Samsung Tizen、XboxOne等几乎所有主流平台，开发者可以通过一次开发，进

而以极小的代价发布到多个平台上去【7】。这样就大大的增加了网络3D试衣间的 应用范围，使其可以发挥最大的商用价值。相比之下虚幻三引擎虽然己经预留了 与OS系统对接的研发接口，但其存在前期技术难吃透，后期支持有欠缺的劣势， 导致配置门槛高优化难，应用此引擎很难顾及通过它来实现游戏的跨平台性。而 苹果不支持Flash的局面，造成了 Flash3D在跨平台性上有难以逾越的鸿沟。因 此Unity3D在跨平台性上的优势是虚幻三和Flash3D引擎无法比拟的。

(4) 高性能高效性：一些用Flash开发需要非常繁琐的程序才能完成的项目， 用Unity 3D只需稍微设置一下，用很短的控制脚本就能轻松完成，其运行速度 则大大高于Flash。Unity3D支持画面内三角面数量较多，当场景中角色不多时， 角色包含的三角形数量可以为10000-15000个，若是大型的或网络游戏中，此数 量显然是比较高的【8】，大大高于同为3D引擎的PV3D，PV3D只能支持画面内 最多3000个三角面。具有更高的性能的Unity3D支持将3D试衣间中使用的模 型制作的更为精致。

(5) 强大的编辑能力：完成3D试衣间的换装功能，需要读取模型资料生成 文件管理器、渲染生成角色、调用文件管理器数据进行换装、存储换装数据等流 程，Unity3D支持编程语言JavaScript、C#和Boo【9】，可供实现这些流程所需程 序的编写。其提供层级式综合开发环境，具备视觉化编辑，时预览的强大综合编 辑能力对实现此换装功能提供很多方便。例如:所有显示在属性编辑器中的属性， 都可以直接修改，这不仅包括普通的游戏物体，还包括脚本。在Unity3D中，有 一个Components (组件）的概念，物体的一些属性，包括实施到游戏物体上的脚 本都被作为游戏物体的“组件”显示在属性编辑器面板中。而对于脚本中的变量， 只要被定义为“公共”的，也会显示在属性编辑器中，用户甚至可以在游戏运行 时使用属性编辑器来改变这些变量值，观察游戏的效果，而无须重新打开脚本编 辑器来编辑它们。而对于某些类型的“游戏物体”变量类型，用户可以直接从层 次面板中拖动需要的游戏物体到属性编辑器中，来完成变量的分配【10】。这强大 而方便的编辑能力为本课题的研宄提供了方便

(6) 简单易用：Unity3D拥有可视化编程界面，用户可以直观的在界面上进行 模型拖拽以搭建场景，将写好的程序拖拽到相应模块上进行关联。拥有便利的 导入功能，与PaperVison3D等Flash3D引擎相比，操作更易实现，节省开发时 间。在Flash3D引擎中如要实现把模型加载入引擎中并搭建舞台，调入摄像机 和灯光等功能，需要编写不少程序才能进行。而Unity3D只需直观的将模型拖

-10- 拽到界面中，可视化的添加摄像机、灯光到所需位置即可，甚至不需一点程序 介入。

2.1.4 3D试衣间模型制作环境分析

本课题研宄的3D试衣间系统的场景、细分角色、细分服装都需要模型制作 软件进行制作，由于骨骼动画需要在iClone中套用，而iClone只能使用己绑定 好骨骼的模型，因此在建模工具中必须完成角色模型和服装模型分别与骨骼的 绑定工作，这就都需要具有模型制作能力和骨骼绑定能力的模型制作工具进行 支持。Maya是世界上最为优秀的三维动画的制作软件之一，其应用主要是动 画片制作、电影制作、电视栏目包装、电视广告、游戏动画制作等[11】。MAYA 在制作模型和骨骼绑定方面功能完善，因此本课题选用MAYA作为模型制作工 具。与Maya相比，3DSMAX软件应用主要是动画片制作、游戏动画制作、建 筑效果图、建筑动画等。MAYA和3DSMax都是高端3D软件，两者之间都有很 多相同的功能，像创建模型，渲染材质，动画制作等等.但就运用实际情况而 言,3DMax更加适合于建筑学，室内设计等等，而MAYA在如角色动画、运动 学模拟、完美的材质等方面更为突出【12】。本课题模型需要使用骨骼动画，因此 选择了动画方面更为突出的MAYA作为模型开发软件。

2.1.5 3D试衣间动画制作环境分析

本选题研宄的3D试衣间模特是具有骨骼动画的，而且骨骼动画是可以进行 更新的，因此选择一种简便的制作骨骼动画方式，可以减少工作量，提高工作效 率。因此选择了 lClone作为制作骨骼动画的主要技术，它制作动画的效率非常 高，且对于3D动画新手来说又比较容易上手，使用它可以立竿见影的解决各 种3D动画实现难题。iClone具有功能强大和门槛较低的特点。

iCLone制作骨骼动画效率高的原因在于：

1、 适用范围广、门槛低：无论是3DSMAX还是Maya建立的带骨骼模型，都可 以使用iClone 3DXchange插件转换为iClone可用的模型，直接套用骨骼动画， 然后使用iClone 3DXchange转成FBX格式文件。虽然这种直接导出的FBX格式 文件想要应用于Uinity3D引擎建立的项目中，还需要导入MAYA进行模型大小、 骨骼动画的烘焙处理，但用其制作动画可以省去了在Maya里制作IK、FK、线 控、控制器等复杂的步骤，使制作骨骼动画的门槛大大降低。

2、 简单的骨骼动画制作：iClone可以使用庞大的模型库资源，包括Poser软件 所提供的所有高质量模型，都可应用于iClone，对于不会建模的动画新手来说， 这无疑是福音，即使需要对动作进行调整或者原创动作，其操作界面都非常直观，

-11- 很好理解。对于传统的模型骨骼动画制作来说，资源库是相当难以得到的资源。 若要自己进行动画录制，需要购买价格不菲的动态捕捉器设备，这对于一般的业 余爱好者或者小型工作室，都是一笔难以接受的花费；但是若要依靠自己DIY 骨骼动画的话，无论是在3DSMAX中还是MAYA里操作，都非常复杂，不仅要 先建立并设定好模型的各部分骨骼物件，同时还必须拥有具备骨骼动画制作经验 的动作师，花费大量时间才能调试出自然的骨骼动画，稍有不慎就会让模型动作 出现各类问题。与之相比，在lClone中只要轻松地拖曳资源库中的骨骼动画到 准备好的模型上，在时间轴里加以组合排列，就可以让人物模特完成各种复杂动 作。若想要自定义骨骼动画，也可以逐帧进行动作调试，与3DSMAX和MAYA 中调试的不同之处在于其调控更为简单直观，只需点选骨骼操作面板的节点，在 视图界面中就可对骨骼进行旋转、移动、缩放以更改模型动画，如图2-1所示

2-1 lClone骨髂动画自定义调试面板

2-1 lClone skeletal animation custom debugging panel

3、模型面部定制：lClone可以简单实现3D模型面部定制功能，只需传入2D面 部照片，进行简单的调试后，就可以将模型的面部改为照片所示，该功能不仅仅 是将2D面部照片形成贴图蒙在模型上，还可以对模型面部特征进行调整，如模 型鼻子的高度，眼睛的大小，以适应照片所示人物的真实特征。这就使得iClone 对模型制作的准入门槛真正的得以降低，从模型库下载的模型，通过这步面部定 制的简单修改就可以巧妙的化为己用。虽然这样出来的模型面数较多，且有可能 比较粗糙，但直接提供出的人物脸型网格，可以拿到MAYA中进行修改，比在 MAYA中直接建模制作要方便很多。

2.1.6 3D试衣间开发工作流程分析

3D模型是试衣间系统所需的基本要素，需要使用Maya软件对模型进行制

-12- 作，并且进行骨骼绑定、蒙皮，以得到最原始的模型素材。使用iClone3DXchange 插件对原始模型素材进行处理，以取得iClone可以直接使用的骨骼模型，并且 在iClone中为模型添加骨骼动画，调整好后再次导入iClone3DXchange插件，将 带有骨骼动画的模型转换为FBX格式，以供Unity3D引擎使用。通过C#语言编 写程序，在Unity3D中进行GUI菜单界面制作，以实现用户点击选择换装功能， 模型通过Unity3D进行渲染生成，通过C#语言编写人物模型生成模块、文件管 理器模块、数据字典、程序配置模块完成系统的整体制作。试衣间开发工作流程 图如图2-2所示：

图2-2 3D试衣间开发工作流程图 Figure2-2 3D fitting room system development flow chart

2.1.7 3D试衣间换装流程分析

程序开始运行后，进行程序配置设定，访问文件管理器读取程序必要的模特 基本数据集合，用于Unity游戏引擎渲染。如果访问文件管理器失败，则程序进 入等待数据就绪状态；如果数据成功读取，则检查上一次配置数据是否存在，若 存在就读取配置，下载配置所需的数据资源包；若是第一次打开程序，无法找到 模特配置，就随机生成一套模特部件方案，并且下载相应的数据资源包。当被请 求的资源加载完毕，根据配置生成模特，播放模特走入试衣间的动画，使用淡入 淡出的方式使动画播放更加自然。当用户点击换装按钮进行换装操作时，按钮对

-13-

应的部件下载资源包并替换原部件，下载完毕后播放相应的换装动画并将动画属 性设置为非循环。完成换装后可选择保存现有配置。人物换装结束后，用户即可 退出系统。换装的流程图如图2-3所示：

图2-3换装流程图

-14-

2.2 3D试衣间功能需求分析

3D网络试衣间具有场景渲染、人物模特渲染、不同服装发型的渲染、换装 渲染、模特行走以及换装动画的播放、试衣镜、读取存储配置等功能，本节将对 其功能需求进行分析。

2.2.1 3D试衣间换装功能分析

换装功能是3D试衣间的核心功能，用户需要运用该功能进行换装体验。换 装功能需求主要包括下面几部分：

(1) 用户菜单界面：用户可以通过点选菜单界面、按钮控制换装功能的实 现。

(2) 数据字典：存储换装部件模型网格、骨骼、贴图、动画的名字与彼此 之间的相关联系，组成一本逻辑上的数据字典，用于数据资源的分类和管理， 供角色生成模块调用。

(3) 数据文件管理器：存储承载数据集合的资源包。将模特按试衣间换装 功能分为多个部分，每个部分的模型以及服装需要按一定规律建立文件名，存储 在模型数据集合资源包中，以供用户选择菜单界面时调取。

(4) FBX文件分类处理器：FBX文件作为3D试衣间的基础提供者，需要 被区分为模型文件和动画文件两种。虽然同为FBX文件，但模型文件需要向程 序提供的是其包含的模型网格、骨骼、贴图信息，而动画文件需要提供的仅仅是 其包含的动画信息。因此需要制作FBX文件分类处理器，在命名的时候区分模 型文件和动画文件，以便程序在存储信息时，可以分类存储模型、动画数据，并 且删除不必要的重复信息。

(5) 角色生成器：将数据字典以及数据文件管理器中存储的逻辑关系列表、 模型数据集合资源包进行组合、调用，渲染生成模特。

2.2.2 3D试衣间菜单界面功能分析

3D试衣间的换装功能需要使用菜单界面来进行整体操控，菜单界面应具有 以下分类，供用户选择和操作：

(1) 模特性别选择：可选择男、女模特作为用户虚拟化身，展示供不同性别角 色使用的服饰、装备。

(2) 模特可换装部件分类：提供面部、眼睛、头发、上衣、裤子、鞋子6个可

-15- 更换的部件分类，供用户选择自由搭配。

(3) 换装配置:保存配置功能与删除配置功能，用于存储和读取己生成的性别、

换装配置，使用户可以基于上一次的配置对模特衣服进行改进，也可以删除整套 配置从头开始。

2.2.3场景、模特、动画功能分析

3D试衣间需制作场景、模特模型，为系统提供模型基本要素，具体需要制 作：

(1) 试衣间场景模型：包括房间、地板、试衣镜、地毯、试衣架等设施

(2) 模特模型：一男一女两个模特

(3) 模特要求：模特模型需细分建模，由面部、眼睛、头发、上衣、裤子、鞋 子6个部分组成。每个部分制作两个及以上的网格或贴图，供换装选择。

(4) 动画功能：在模特入场、站立、更换上衣、裤子、鞋子，这五种情况下， 模特需要播放相应的动画，因此需要准备五个动画FBX文件，每个动画文件， 都需要将细分的模特所有部件与骨骼相绑定并赋予动画。

2.2.4 3D试衣间的试衣镜功能分析

3D试衣间要实现360度观看模特的功能，需要在模特背后放置一个试衣镜， 用以反射模特背向用户的一面。这需要使用Umty3D Pro所提供的镜面反射 Shader渲染方法，在试衣镜模型上调用此方法，使得试衣镜模型具有镜面反射 功能。

2.3 3D试衣间数据模型 2.3.1 3D试衣间系统用例图

本系统共包含性别选择、服装选择、面部、头发、眼睛、衣服、裤子、鞋、 服装配置、配置保存、配置清除、参数设置、图片质量、屏幕分辨率、是否全屏， 共十五个用例。

-16-

第3章3D试衣间的软件与模型设计 3.1试衣间概要设计 3.1.1系统体积结构图

从3D试衣间所需的主要功能上，对系统的功能框架设计。如下图4-1所示:

3D试衣fuj系统

参数设置 性别选择 服装选择 服装配置

图3-1 3D试衣间的系统功能图

Figure3-1 3D dressing room ‘s Function Chart

系统由四大功能模块组成，分别为参数设置、性别选择、服装选择和服装配置。

(1) 参数设置：用户可以在此界面对3D试衣间系统的图像进行配置。这个模 块提供三种功能：

a) 对图片质量进行选择，分别有六种画面质量可供用户选择。

b) 屏幕分辨率选择，为用户提供多种分辨率，以适应其操作平台。

c) 全屏模式选择，用户可选择是否全屏，不勾选窗口模式为全屏，勾选则 为窗口

(2) 性别选择：用户可以根据实际情况选择模特的性别。

(3) 服装选择：包括六种选择类型，用户可以通过对这六部分的点选，进行服 装搭配，其分类为:面部、头发、眼睛、衣服、裤子、鞋。

(4) 服装配置：用户可以选择配置保存来存储己选好的整套服装方案，保存后 当再次进入该试衣间系统时，模特的初始服装搭配将更换为保存好的这套方案。 当选择配置清除选项时，则回归模特初始设置的默认服装方案。

-19-

3.1.2系统层次结构图

本试衣间系统实例主要借助鼠标与用户进行交互。用户通过鼠标点击为模特 选择不同的衣、裤、鞋子，从而产生各种颜色、各种款式之搭配，为用户提供直 观上的参考。为了达到以上的目的，试衣间程序实例中必须使用大量预设的模型 基本组成元素，如不同款式的上衣与鞋裤。因此，程序实例需要与一个模型基本 元素的文件管理器进行交互。根据实际情况的需要，这个文件管理器可能是存在 于本地的文件系统中，也可能存在于互联网的远程数据库中。整个系统的交互关 系，

如下图3-2所示:

图3-2系统层次结构图 Figure3-2 System hierarchy diagram

模型基本元素包括男女模特以及与之配套的各种衣裤模型，因此程序实例需 要对从文件管理器中取得的原始数据进行加工处理，这种处理主要包括对原始模 型数据的分类与索引，从而使得程序能够更高效地使用。通过将模型各部分数据 进行组合从而生成最终的服装搭配方案，组合后的完整人物模型的加载与渲染工 作由程序实例调用Umty3D引擎完成。

明确了系统的各项功能以及它与各外部实体间的交互关系，我们就可以对系 统进行更加详细的规划和设计，进一步将系统划分为具体的功能模块，并设计模 块间的交互关系。

3.2试衣间文件管理器设计

3.2.1模型文件管理器模块的设计

-20-

3.1.2系统层次结构图

本试衣间系统实例主要借助鼠标与用户进行交互。用户通过鼠标点击为模特 选择不同的衣、裤、鞋子，从而产生各种颜色、各种款式之搭配，为用户提供直 观上的参考。为了达到以上的目的，试衣间程序实例中必须使用大量预设的模型 基本组成元素，如不同款式的上衣与鞋裤。因此，程序实例需要与一个模型基本 元素的文件管理器进行交互。根据实际情况的需要，这个文件管理器可能是存在 于本地的文件系统中，也可能存在于互联网的远程数据库中。整个系统的交互关 系，

如下图3-2所示:

图3-2系统层次结构图 Figure3-2 System hierarchy diagram

模型基本元素包括男女模特以及与之配套的各种衣裤模型，因此程序实例需 要对从文件管理器中取得的原始数据进行加工处理，这种处理主要包括对原始模 型数据的分类与索引，从而使得程序能够更高效地使用。通过将模型各部分数据 进行组合从而生成最终的服装搭配方案，组合后的完整人物模型的加载与渲染工 作由程序实例调用Umty3D引擎完成。

明确了系统的各项功能以及它与各外部实体间的交互关系，我们就可以对系 统进行更加详细的规划和设计，进一步将系统划分为具体的功能模块，并设计模 块间的交互关系。

3.2试衣间文件管理器设计

3.2.1模型文件管理器模块的设计

-20-

如上所述，模型文件管理器中包含模特的身体各部分模型。除此以外，模型 文件管理器中还应该包括多个模特的模型数据，以满足用户改变模特性别的功能 需求。某个特定模特的模型数据应该互相兼容，以达到自由相互替换目的；而不 同模特的模型数据之间则互不兼容，即模型数据只属于某个特定模特，不能在多 个模特之间共享。

模型文件管理器还应包含模特的骨骼和动画数据以供程序检索和使用。虽然 一个模特具有多套模型数据，但由于这些数据可以相互替换，因而互相兼容，所 以可以共用同一套人体骨骼和动画效果，即在文件管理器中只需为一个模特保存 一套骨骼和动画数据。

因此，模型文件管理器的逻辑组织

如图3-3所示：

图3-3模型文件管理器逻辑组织图 Figure3-3 Model database logical organization chart

其中，模型文件管理器与模特是一对多的关系；模特与其骨骼和动画数据是 一对一的关系；而模特与其各部分可相互替换的模型数据则是一对多的关系。

3.2.2数据字典模块的设计

当用户选定模特并开始为其搭配服装时，更换服装并重新渲染就是程序进行 的一个主要操作。这是本系统的一个最主要的功能，也是最典型的操作。在这个 过程中，程序需要频繁地在大量的模型数据中定位用户所选定的数据，并将模型 数据传递给Unity游戏引擎进行渲染。由于渲染是实时进行的，因此不应在此过 程中产生过长的延迟时间，否则程序中换装效果展示的实时性就无法保证。所以， 在综合考虑检索模型数据的频繁性和渲染的实时性之下，应该将模型数据载入内 存，并将其索引，以利于搜索，从而降低检索数据的延迟时间。

-21-

具体来说，本系统采用字典的索引方式分层地组织和管理数据，基于其通过 名字查找数据的概念，这个存储和处理数据的逻辑实体在本系统中就称为数据字 典。该数据字典为程序中的其他模块提供了快速检索模特模型数据的功能，以达 到高效使用结果数据的目的。其中的数据排列组织方式与模型文件管理器中数据 的逻辑关系保持一致，这样有利于迅速地通过模特名字检索到其特定的模型数 据。根据上小节叙述的模型数据间的逻辑关系，本系统将数据字典中的数据分层、 分类地进行管理，

其中数据的索引关系如图3-4所示：

图3-4数据字典引索关系图

Figure3-4 Primer cord diagram data dictionary

其中，模特与模型数据之间仍然是一对多的映射关系，与模型文件管理器中 数据的逻辑关系是一致的。但在两者之间加入了一个中间层，进一步将模特的模 型数据划分为面部、眼睛、头发、上衣、裤子和鞋子共六大类，系统中所有的模 型数据被分门别类地存储，这样设计可以使数据字典模块的检索功能更易于使 用，查找数据更为便捷。

具体来说，本系统采用字典的索引方式分层地组织和管理数据，基于其通过 名字查找数据的概念，这个存储和处理数据的逻辑实体在本系统中就称为数据字 典。该数据字典为程序中的其他模块提供了快速检索模特模型数据的功能，以达 到高效使用结果数据的目的。其中的数据排列组织方式与模型文件管理器中数据 的逻辑关系保持一致，这样有利于迅速地通过模特名字检索到其特定的模型数 据。根据上小节叙述的模型数据间的逻辑关系，本系统将数据字典中的数据分层、 分类地进行管理，

其中数据的索引关系如图3-4所示：

图3-4数据字典引索关系图

Figure3-4 Primer cord diagram data dictionary

其中，模特与模型数据之间仍然是一对多的映射关系，与模型文件管理器中 数据的逻辑关系是一致的。但在两者之间加入了一个中间层，进一步将模特的模 型数据划分为面部、眼睛、头发、上衣、裤子和鞋子共六大类，系统中所有的模 型数据被分门别类地存储，这样设计可以使数据字典模块的检索功能更易于使 用，查找数据更为便捷。

3.3试衣间详细设计

3.3.1核心模块的设计

本系统的核心功能是在用户面前呈现一个虚拟的“自己”，并提供给用户一

-22- 个服装库，使用户可以从库中为假想的“自己”选择服装，系统会实时地将用户 的选择反映在“自己”身上，从而迅速而直观地，以类似“照镜子”的方式向用 户展示出不同的服装搭配所产生的不同效果，并为用户评估和衡量服装的搭配对 “自己”的外在风格和内在气质的影响提供参考，最终辅助用户挑选出一套或多 套最满意的服装。

具体地，对于用户来说，屏幕上的“自己”实际上是一个三维的、立体的人 物模型，用户可以将这个模特假想为“自己”，并为“自己”配制一套合适的服 装。为了使模特能够尽量贴近实际用户的审美，真实地体现中意的服装穿在这个 虚拟化身身上的效果，应该能够通过本系统对模特的外形进行调整。为此，本系 统允许用户对模特进行一定限度的定制。比如，用户可以根据实际情况首先选择 模特的性别，并进一步选择模特的头发颜色、发型和眼睛颜色，这样有利于最大 限度地按用户所需达到其满意的效果。

用户完成模特外形的定制后，就可以为其选择和搭配服装了。用户可以从菜 单中为模特选择上衣、裤子和鞋子进行搭配，不同款式和颜色的服装和鞋子都可 以在同类型间相互替换，即不同的上衣之间可以自由地相互替换，裤子和鞋子也 相类似。各种服装来源于服装库，这个服装库应该能够动态增长，即可以随时添 加新款的服装到库中，从而更新服装库为用户提供更多的选择。

综上所述，本系统将使用一个三维立体的模特为用户试衣，这个模特的身体 各部分，如头发、眼睛、上衣、裤子和鞋子都可以在同类型间相互替换，以达到 定制模特外形和为模特换装的目的。而从程序的角度上说，模特上衣和鞋裤的替 换实际上就是其上半身和下半身的模型替换，这本质上与替换模特头发和眼睛的 概念是相同的。因此，服装库具有相对广义的内涵，它既包括模特上半身和下半 身模型，同时也包括模特的头发和眼睛这些外形元素的模型，确切地说，服装库 实际上应该是一个人物模型的文件管理器。

3.3.2模块关系图

根据以上所述，用户可以通过本系统对模特的服装进行选择和搭配，因此本 系统首先要具备与用户进行交互的功能，这一功能主要由用户界面模块，即服装 选择菜单来实现。当用户通过菜单为模特选择了某种搭配的服装后，程序需要通 过一种人物模型的生成算法将模特各部分的模型数据进行组合和渲染，该功能主 要由系统中的模型生成模块完成。组合完成的完整模型由Unity游戏引擎进行渲 染并呈现在客户端视区中。整个系统的模块依赖关系如图3-5所示：

-23-

图3-5模块关系图

Figure3-5 Module diagram

其中主程序模块包括一些必要的程序流程控制逻辑和用户界面子模块。用户 界面即服装选择菜单，用户通过该菜单为模特搭配服装。菜单中的每一项对应组 成模特某个部分的基本数据，由系统中的基本数据模块进行管理。具体来说，人 物模特分为头发、面部、眼睛、上身、下身和脚这六个基本组成部分，每部分都 是一个基本数据模块，该模块包含有不同风格的人物外形或者服装的多种可选模 型，各个模型间可以自由相互替换，从而达到模特换装的程序功能。而数据集合 模块，顾名思义，即代表模特各部分基本数据的集合，用于存储系统中所有模特 的基本数据，并与文件管理器中的数据存储形式保持一致，以方便读取。基本数 据和数据集合这两个模块都由人物模型生成模块所引用，用于辅助该模块生成和 渲染人物模型。其中，人物模型生成模块又包括如下三个子模块：

(1) 模型文件管理器：模型文件管理器模块根据用户在菜单中的选择，访问本 地或远程数据库获取相应的模型基本数据，如模特上下身服装等，存储于模型数 据集合模块中。

(2) 数据字典：为了更方便、高效地使用模型基本数据，本系统通过数据字典 模块实现对这些数据的分类和管理，以供人物模型生成模块调用，其操作的基本 单位为单个基本数据。

(3) 程序配置：将用户认为满意的一整套服装搭配保存起来，就构成了一个程 序配置。该配置包括模特各个部位的模型基本数据，将这些数据组合起来就构成 了一套完整的服装配置。该套服装配置可以在下一次程序启动后自动加载，这样 用户在每次启动程序后就可以看到自己上一次保存的服装配置。用户可基于上一

-24- 次的配置进行改进，也可以删除整套配置从头开始。

至此，本系统中的各个模块就划分完成，模块的功能和模块间的相互依赖关 系也基本明晰。

3.3.3模块交互图

将系统功能模块化之后，需要进一步建立模块间的交互关系，并确定在交互 过程中模块间必要的数据传递。基于上节的分析，本系统具体的模块交互关系如 图3-6所示：

图3-6模块交互图 Figure3-6 Module interaction diagram

如上节所述，本系统分为主程序模块、人物模型生成模块和Unity游戏引擎 三大模块。模块间的交互主要体现为这三个模块之间的数据传递和处理。

主程序和人物模型生成模块间存在两种交互方式：

(1)当系统加载后，主程序将上一次的服装配置参数传递给人物模型生成模块， 该参数由用户上一次运行程序时保存，并以文件的形式保存在文件系统中以供程 序在下一次运行之初将其加载。具体的配置参数包含一套完整的模特基本数据， 其中每一项基本数据都对应模特身体的一个组成部分，通过人物模型生成模块进 行组合，并传递给Unity游戏引擎进行模特整体的渲染，渲染成功后呈现给用户 上一次保存的模特，以便用户继续之前的工作；

-25-

(2)用户通过点击菜单，进行更换模特操作，或为选定模特更换服装操作的过 程中，调用人物模型生成模块对模特进行重新组合，并将组合后的模型基本数据 传递给Unity游戏引擎，由游戏引擎渲染模特和服装呈现给用户。

另外本系统中，人物模型生成模块通过模型文件管理器子模块访问远程或本 地文件管理器，将所有模特的数据一次性读入数据集合中，再由人物模型生成模 块以数据集合为载体，将数据传递给数据字典子模块，在数据字典模块中进行数 据的分类和排序，并将处理完毕的数据以基本数据的形式存储在系统内存中，以 供人物模型生成模块后续的访问和检索，这样使系统运行过程中无需再访问文件 管理器，从而提高程序运行效率。

3.4 3D试衣间模型设计 3.4.1人物模特模型设计

人物模特需要进行建模，首先在Photoshop中，用画笔工具配合手绘板画出人物 的正面以及侧面图，不需要非常的严格，因为在建模当中会适当调整审美与姿势， 但需要画出人物大体比例，供建模参考。如图3-7所示：

图3-7 3D试衣间的人物模特绘制 Figure3-7 3D dressing room ‘s character models draw

在MAYA中，分别建立模型，将整个人物模型，细分为头发、面部、眼睛、上 身、下身、脚这六部分，以配合程序调取模型各部分进行换装操作，每一部分需 制作若干个不同款式的素材，以供用户选择。Unity3D是一款高性能引擎，其允

-26-

1. 用户通过点击菜单，进行更换模特操作，或为选定模特更换服装操作的过 程中，调用人物模型生成模块对模特进行重新组合，并将组合后的模型基本数据 传递给**Unity**游戏引擎，由游戏引擎渲染模特和服装呈现给用户。

另外本系统中，人物模型生成模块通过模型文件管理器子模块访问远程或本 地文件管理器，将所有模特的数据一次性读入数据集合中，再由人物模型生成模 块以数据集合为载体，将数据传递给数据字典子模块，在数据字典模块中进行数 据的分类和排序，并将处理完毕的数据以基本数据的形式存储在系统内存中，以 供人物模型生成模块后续的访问和检索，这样使系统运行过程中无需再访问文件 管理器，从而提高程序运行效率。

3.4 3**D试衣间模型设计** 3.4.1**人物模特模型设计**

人物模特需要进行建模，首先在**Photoshop**中，用画笔工具配合手绘板画出人物 的正面以及侧面图，不需要非常的严格，因为在建模当中会适当调整审美与姿势， 但需要画出人物大体比例，供建模参考。如图3-7所示：

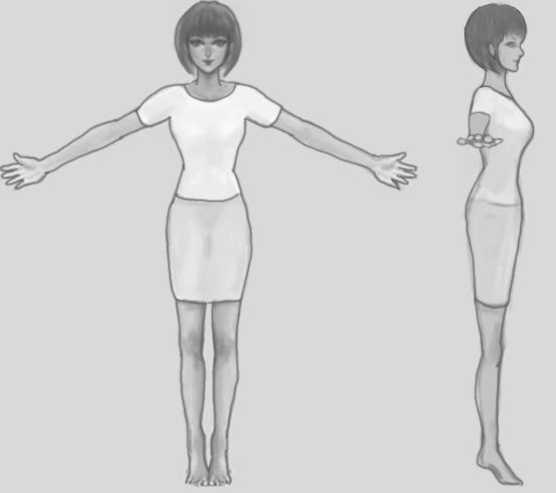


图3-7 3**D**试衣间的人物模特绘制 **Figure**3-7 3**D** **dressing** **room** **‘s** **character** **models** **draw**

在**MAYA**中，分别建立模型，将整个人物模型，细分为头发、面部、眼睛、上 身、下身、脚这六部分，以配合程序调取模型各部分进行换装操作，每一部分需 制作若干个不同款式的素材，以供用户选择。**Unity**3**D**是一款高性能引擎，其允

-26- 许整个屏幕出现的模型最多面数可达15000个三角面，因此在控制模型面数上比 较轻松，但为了使程序运行速度更快，建立模型的时候还是本着面数更低原则来 执行，因此六部分的模型组合在一起总共面数控制在了 10000以下。模型建模如 图3-8所示：

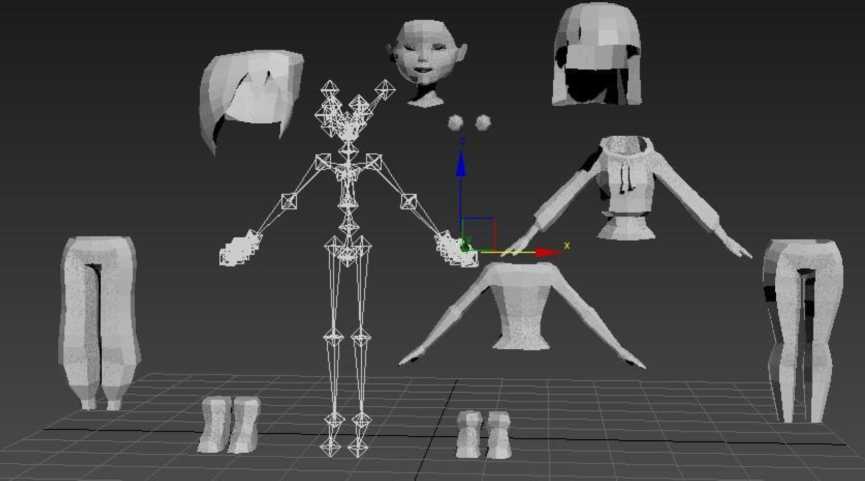


图3-8 3**D**试衣间的人物模特制作

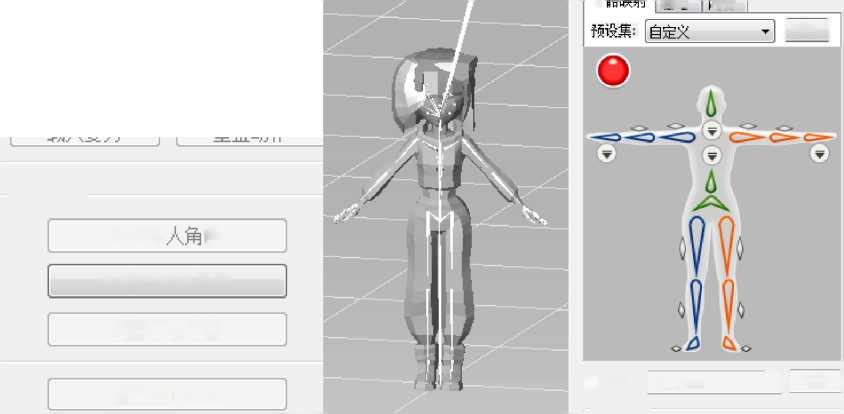
**Figure**3-8 3**D** **dressing** **room** **‘s** **Character** **models** **produced**

3.4.2**人物模特骨骼设计**

试衣间的人物模特设计为可以活动的模特，当进入系统后，模特将从场外走 入场内，每当用户点击菜单，选择不同款式服装后，都将调取模特不同的骨骼动 画，例如：当换了上衣后，模特会惊奇的伸开双臂，左右摆头看自己的新衣服， 以表示更换了上衣。此设计是为了增加换装趣味性，以提升用户体验。因此，必 须设计一套骨骼，并且将模特六部分可换装部分绑定于其上。为了可以让动画进 行的自然流畅，选择了 MAYA软件与iClone 3DXchange插件相结合，进行骨骼 动画的辅助制作，为人物模特选择基础骨骼，并且使用Skin进行蒙皮，只有这 种蒙皮方式可以被Unity3D接受。

在MAYA中选择导出FBX文件，这时将得到一个带骨骼的模型，但并没有 任何动作。打开iClone 3DXchange软件，导入这个没有动作的基础骨骼人物模 型FBX文件，然后选择右边控制面板上角色菜单下的转成非标准角色按钮。

-27-

□启用[00\_Tpose ;] | 予S览 |

导骼内苕

应用至iCbne

如图3-9所示:

曰mm 八」食

1菪除1

角色设定

转成非色  
转成非标准角色

调整标准骨骼

图3-9将骨骼模型文件转成非标准角色 **Figure**3-9**The** **skeleton** **file** **into** **a** **non-standard** **characters**

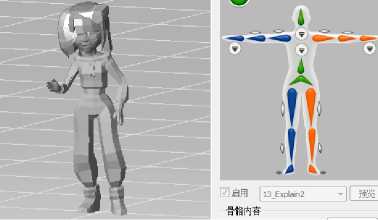
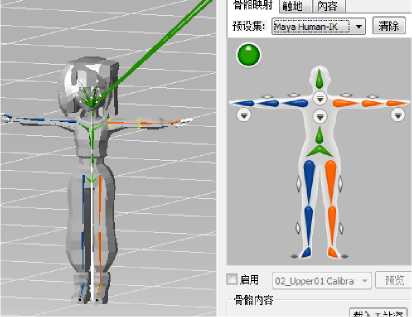
选择骨骼，将站姿调成**T**字形，在右侧面板骨骼映射选项中选择预设集： **MAYAHUMAN**-**IK**骨骼就自动转换成**iClone**可用的骨骼形式。勾选启用选项， 就可以预览将动作库中的动作赋予该骨骼。成功后，点击转换键，就得到一具可 以使用**iClone**动作库中各种动作资源的骨骼。如图3-10所示：

图3-10将动作赋予骨骼人物模型 **Figure**3-10 **The** **action** **gives** **bones**

-28-3.4.3**人物模特动作设计**

为了增强换装的趣味性，给人物模特设计了六套动画，人物骨骼动画文件必须存 为**FBX**格式，以便**Unity**3**D**识别。六套动画分别为：

1. **Idle**:模型闲置时的动作，帧数为100帧
2. **ltem**\_**boots**:换鞋子后的动作，帧数为100帧
3. **ltem**\_**pants**:换裤子后的动作，帧数为100帧
4. **item**\_**shirt**:换衣服后的动作，帧数为100帧
5. **wlk**:走路动作，这个动画只有40帧，只走了两步；
6. **walkin**:走进试衣间时的走路动作，走了五步，然后转身向前，帧数为100 帧

动画设计文件如图3-11:

[**\_Female@idlel.FBX**](mailto:_Female@idlel.FBX)

**Female@item\_boots.fbx** 费 **Female@item\_pants.fbx Female@item\_shirt.fbx** 费[**Female@walk.fbx**](mailto:Female@walk.fbx)[**Female@walkin.fbx**](mailto:Female@walkin.fbx)

图**3-11 3D**试衣间的人物模特动画设计 **Figure3-11 3D dressing room ‘s Character model animation design**

3.5 iClone**骨骼动画的制作**

对于非3**D**软件专业人员来说，制作带有骨骼动画的人物模型，是非常困难的 事情，**iClone**这款软件就是为了非专业人员轻松导出骨骼动画量身定做的。在本 试衣间系统中，模特有6个骨骼动画：模型闲置时的动作，换鞋子后的动作，换 裤子后的动作，换衣服后的动作，走路动作，走进试衣间时的走路动作。这些动 作如果按传统的制作方法，一般有两种制作方案：

-29-

3.5 iClone骨骼动画的制作

对于非3D软件专业人员来说，制作带有骨骼动画的人物模型，是非常困难的 事情，iClone这款软件就是为了非专业人员轻松导出骨骼动画量身定做的。在本 试衣间系统中，模特有6个骨骼动画：模型闲置时的动作，换鞋子后的动作，换 裤子后的动作，换衣服后的动作，走路动作，走进试衣间时的走路动作。这些动 作如果按传统的制作方法，一般有两种制作方案：

-29- 1.穿上一套录制骨骼动画的装备，使用录制动作的仪器，导入软件，进行编辑。 2使用MAYA等3D制作软件，一帧一帧的摆动作，改变模型动态。

这两种方案，前者需要花费大量金钱，录制动作的装备仪器动辄几十万，而后者 则需要具有大量骨骼动作制作经验，具有可以摆出自然的动作的技术，这两种方 案都非常专业，不适合业余爱好者与学习者。而如果使用iClone以及其专属插 件iClone 3DXchange搭配工作，贝lj只需要从iClone内置动作库或者网上的动作 素材库中选择适当的骨骼动画，然后在iClone中进行可视化调整，截取所需的 动作帧，制作完毕，导入iClone 3DXchange中，进行模式转录，最终生成可以 用于Unity3D的FBX格式动画文件，并用于Unity3D。

首先选择刚才己在iClone3DXchange中转换为iClone可识别的骨骼动画的人 物模特，之后在iClone 3DXchange右侧面板中，选择应用至iClone，模型将自动 转入到iClone中。在iClone上方菜单选择Animation动作按钮，左侧就会出现动

作库，每个选项中都有大量骨骼动画可供选择，如果对动画有特殊需求，还可以 去网上下载所需动作，然后拷贝到iClone动作库下的文件夹中，就会显示在左 侧动作库中。点击鼠标左键，可将动作库中的骨骼动画拖拽到屏幕中间的人物模 特身上，只需拖拽完成，模型就被赋予了骨骼动画。如图3-12所示：



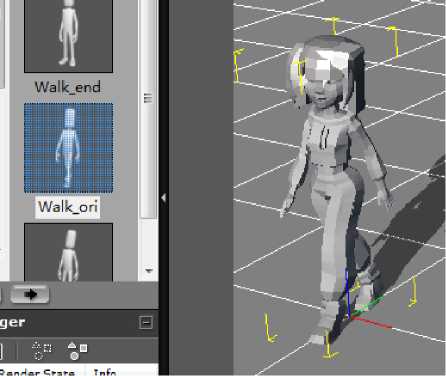


图3-12 iClone骨骼动作库的使用 Figure3-12 iClone bone action library use

-30-

之后可以在屏幕右侧选择Motion Puppet对骨骼动画进行调整，只需调节滑竿 就可以对模型的各部分进行调整，可调节的选项有：膝盖的弯曲程度、挺胸与驼 背、肩膀高低、抬头低头、胳膊抬起和洛下、关节的朝向、步子的大小。调节各 个选项以制作出独特而符合系统所需的骨骼动画效果。如图3-13所示：

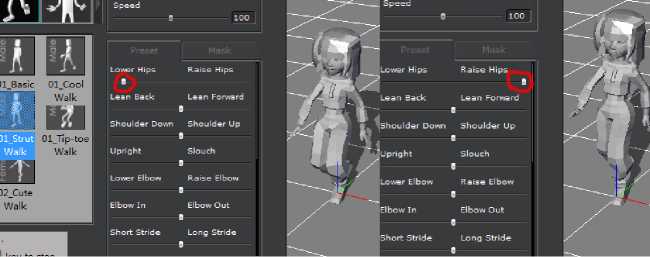


图3-13调节骨骼动画 Figure3-13 Regulating skeletal animation

调节好动画后点击录制，录取所需的动画帧，最后，点选Edit in 3DXchange 按钮，进入iClone专用格式转换插件3DXchange,进行FBX格式输出，由于 Unity3D中只能使用FBX格式的含骨骼动画模型，因此这一步是必不可少的， 至此试衣间的模型与骨骼动画素材准备完毕如图3-14所示：

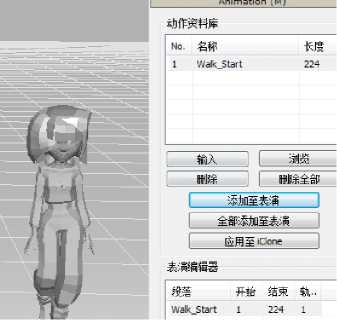




图3-14导出fbx格式动画 Figure3-14 Exporting fbx format animation

-31-

3.6 3D试衣间界面设计

3D试衣间最主要的功能是为虚拟化身换装，因此设计一个让用户能够简单 上手，随心所欲的组合搭配服装的操作菜单界面非常重要。针对用户需求，对于 3D试衣间系统界面的布局设计主要遵循：简单易于上手，易于用户观察的规则 进行。

界面的生成主要由Umty3D的GUI样式代码编写实现，菜单界面样式使用 垂直的布局方式，集合试衣间的全部功能于一个垂直的面板范围内，放置于屏幕 左侧，按钮呈半透明状。当试衣间系统启动，菜单界面将渲染生成。

本系统菜单功能分为：模特性别选择、面部选择、眼睛选择、头发选择、上 衣选择、裤子选择、鞋子选择以及配置的保存与删除。将这些功能集中在同一个 面板上，有助于用户更方便的点选和观察功能，直观易用不会将功能遗漏。

人物模特将从系统的左下走入屏幕中央，因此将菜单放置于屏幕左上方，这 样的布局设置既不会影响画面整体效果，又遵循了从左到右的视觉规律，让菜单 界面可以出现在用户的视觉重心上。按钮呈半透明状，这样不会破坏画面的整体 性，使整个界面更美观。

3.7本章小结

本章主要对试衣间程序的功能、结构、模块以及人物模特骨骼、模型、动画 的总体设计做了详尽的描述，并对程序的界面、系统交互、用户体验设计进行了 阐述。相比其它平台上的2D试衣间，本试衣间系统拥有3D的模特，动作，令 整个试衣间系统具有更高的视觉效果，提高趣味性，成为本系统的特点

**第**4**章**3D**试衣间功能的实现与测试**

第4章3D试衣间功能的实现与测试

4.1 3D试衣间用户界面的实现 4.1.1用户菜单界面的制作

3D试衣间菜单界面由“更换模特”、“面部”、“眼睛”、“头发”、“上 衣”、“裤子”和“鞋子”共七组组件和“保存配置”、“删除配置”两个按钮 组成。每组组件在菜单中占一行，由“<”、“> ”两个按钮和组件名字符组成。 如图4-1所示：



图**4-1**用户菜单界面  
**Figure4-1 The interface of user menu**

用户菜单界面需要使用两种Umty3D引擎的GUI元素来实现绘制，即：

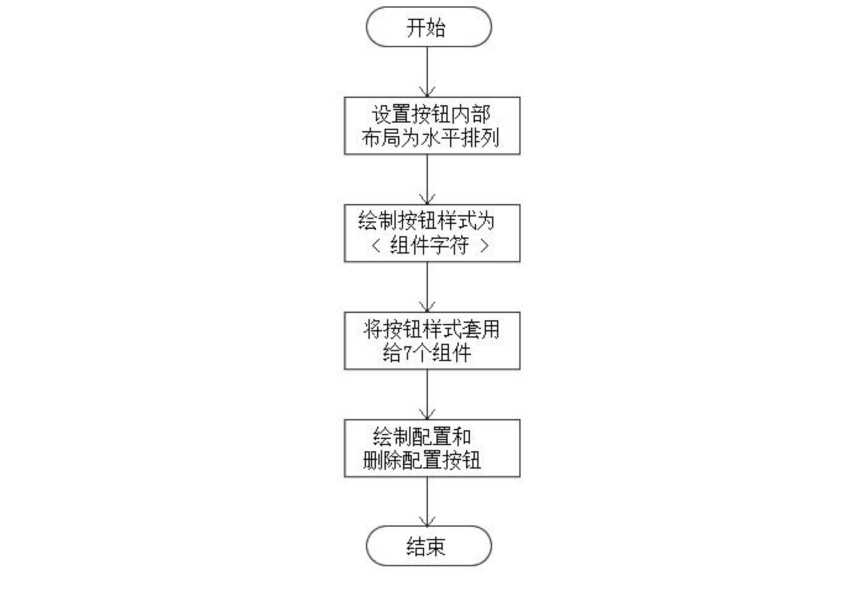
1. GUILayout.Box组件，用于显示只读的文字。
2. GUILayout.Button组件，用于绘制Unity3D引擎提供的标准样式按钮，用户

可以通过点击它与程序交互，并执行设定的功能。

制作用户菜单界面的程序流程如图4-2所示：

-33-

**北京工业大学工程硕士专业学位论文**



图**4-2**制作用户菜单界面流程图 **Figure4-2 Production flow chart of the user menu interface**

实现用户菜单界面代码如下：

//设定菜单按钮区域的位置和大小

GUILayout.BeginArea(new Rect(10, 10, typeWidth + 2 \* buttonWidth + 8, 500));

//水平排列一行按钮组件，绘制"〈"按钮并将值设为false,绘制’>"按钮并将值设

为true,它们中间显示组件名 GUILayout.BeginHorizontal();

if (GUILayout.Button("<", GUILayout.Width(buttonWidth))) ChangeCharacter(false);

GUILayout.Box("Character", GUILayout.Width(typeWidth)); if (GUILayout.Button(">", GUILayout.Width(buttonWidth)))

ChangeCharacter(true);

GUILayout.EndHorizontal();

//为菜单加入换装各组件分类名称 AddCategory("face", "Head", null);

-34-

第4章3D试衣间功能的实现与测试 AddCategory(MeyesM, "Eyes", null);

AddCategory("hairM, "Hair", null);

AddCategory("top", "Body", "item\_shirt");

AddCategory("pants", "Legs", "item\_pants");

AddCategory("shoes", "Feet", "item\_boots");

//绘制保存配置和删除配置按钮 if (GUILayout.Button("Save Configuration"))

PlayerPrefs.SetString(prefName, generator.GetConfig()); if (GUILayout.Button("Delete Configuration"))

PlayerPrefs.DeleteKey(prefName);

4.1.2 3D**试衣间模特初始化的实现**

当用户菜单生成后，用户将看到按照上次存储配置渲染生成的模特走入试衣 间，如果是第一次进入试衣间程序的话，模特的配置会随机生成并渲染，当模特 停止走动则播放模特闲置动画。如图4-3所示：



图**4-3**模特的初始化与入场动画 **Figure4-2 Initialization and admission animation**

实现模特初始化的代码如下：

//当资源数据集合下载完毕

while (!CharacterGenerator.ReadyToUse) yield return 0; //如果有上一次存储配置 if (PlayerPrefs.HasKey(prefName))

-35-

//按上一次存储的配置渲染生成模特

generator = CharacterGenerator.CreateWithConfig(PlayerPrefs.GetString(prefName));

//否则按随机生成配置渲染生成女性模特 else

generator = CharacterGenerator.CreateWithRandomConfig("Female");

//若是刚打开程序，第一次加载模特，则播放走入场的动画 if (!firstCharacter) return; firstCharacter = false;

if (character.animation["walkinM] == null) return;

//当用户菜单生成后，使渲染生成的角色走入试衣间

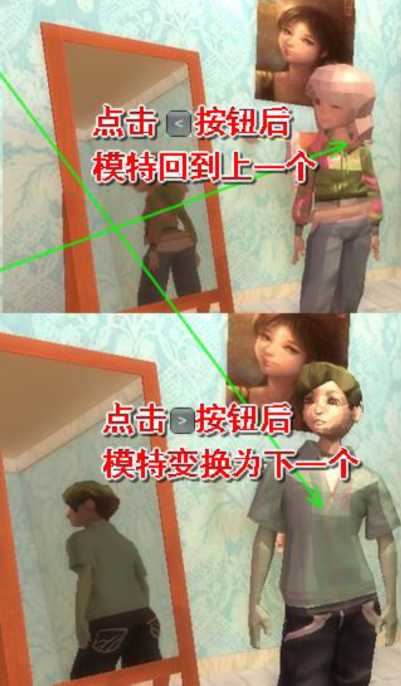
GUI.enabled = usingLatestConfig && !character.animation.IsPlaying(MwalkinM); //模特走入试衣间后，播放闲置动画"idler character = generator.Generate(); character.animation.Play("idle1");

character.animation["idle1"].wrapMode = WrapMode.Loop; newCharacterRequested = false;

4.1.3界面控制组件换装功能的实现

用户菜单上的按钮，通过设置可与程序交互，并执行设定的功能。使用此功 能可实现选择模特性别以及为模特换装，以“选择性别更换模特”为例，如图 44所示：

-36-

第4章3D试衣间功能的实现与测试

eeeseeel

80008891



曰曰曰|〔|曰曰|

**Character**

el@3lel@p\_ I

ooeoeoel

图**4-4**性别选择功能 **Figure4-4 Gender selection function**

**Delete Configuratior**

**Save Configuration**

**Save Configuration**

具体实现代码为：

//水平布局按钮

GUILayout.BeginHorizontal();

//若点击，则选择上一个模特

if (GUILayout.Button(M<M, GUILayout.Width(buttonWidth)))

ChangeCharacter(false);

//更换模特

GUILayout.Box(MCharacterM, GUILayout.Width(typeWidth));

//若点击则选择下一个模特

if (GUILayout.Button('>M, GUILayout.Width(buttonWidth)))

ChangeCharacter(true);

GUILayout.EndHorizontal();

成对使用 GUILayout.BeginHorizontal()和 GUILayout.EndHorizontal()，对包含 在其中的GUI组件进行水平布局，这样保证GUILayout.Button和GUILayout.Box

组件都会按照代码中的出现次序水平排列成一行。其中，“ <”和“>”按钮用于

-37-

更换上一个或下一个模特。

“更换模特”之后的六个菜单项的实现方法相同，并且与“更换模特”只有 细微的差别，限于篇幅，这里只列出“上衣”菜单项的实现加以说明，在“上衣” 菜单选项中，点击“> ”按钮更换模特上衣，点击“<”返回上一件上衣，并播放 更换上衣对应的换装动画“上衣”更换功能如图4-5所示：

**Save Configuration**

模特

图**4-5**上衣更换的功能

**Figure4-5 The function of change the tops**

//水平排列组件 GUILayout.BeginHorizontal〇;

if (GUILayout.Button("<", GUILayout.Width(buttonWidth)))

//选择上一件上衣 ChangeElement("body", false, anim);

GUILayout.Box("上衣"，GUILayout.Width(typeWidth)); if (GUILayout.Button(">", GUILayout.Width(buttonWidth))) //选择下一件上衣 ChangeElement("body", true, anim);

-38-

GUILayout.EndHorizontal();

GUI组件仍然是水平排列，但差别在于这里“<”和“>”按钮的功能应该设 定为更换模特的上半身的模型数据。因此，需要调用ChangeElement()，传入三 个参数，分别指定模型数据属于模特的哪个身体部位、选择上一套还是下一套模 型数据，以及模特渲染完成后需要播放的动画效果。

以此类推，模特“面部”、“眼睛”、“头发”、“裤子”和“鞋子”这其 余的五个菜单项的实现方法是类似的，在此不再赘述，只列出这五个菜单项功能 实现的截图，进行功能测试。

在“面部”菜单选项中，点击“>”按钮更换模特脸型，点击“<”返回上一 个脸型，“面部”更换功能如图4-6所示：



图**4-6**面部更换的功能 **Figure4-6 The function of change the head**

-39-

在“眼睛”菜单选项中，点击“> ”按钮更换模特眼球，点击返回上- 个眼球，“眼睛”更换功能如图4-7所示：

@e\_oy@@@

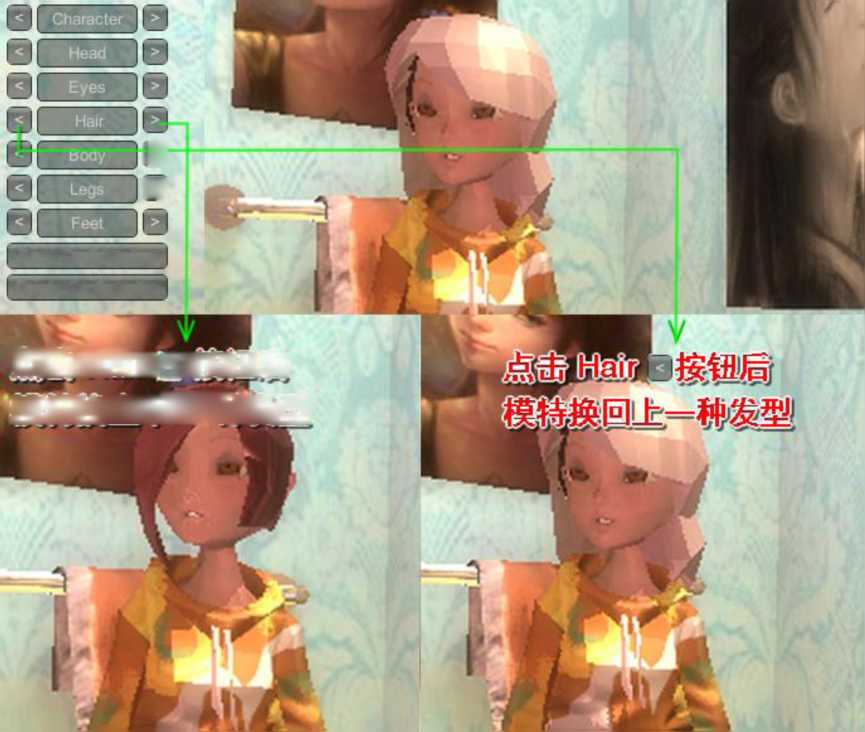
篇||\_|\_

ee@e@@®)

图**4-7**眼球更换的功能  
**Figure4-7 The function of change the eyes**

在“头发”菜单选项中，点击“>”按钮更换模特发型，点击“<”返回上- 种发型，“头发”更换功能如图4-8所示：

-40-

丨] r?

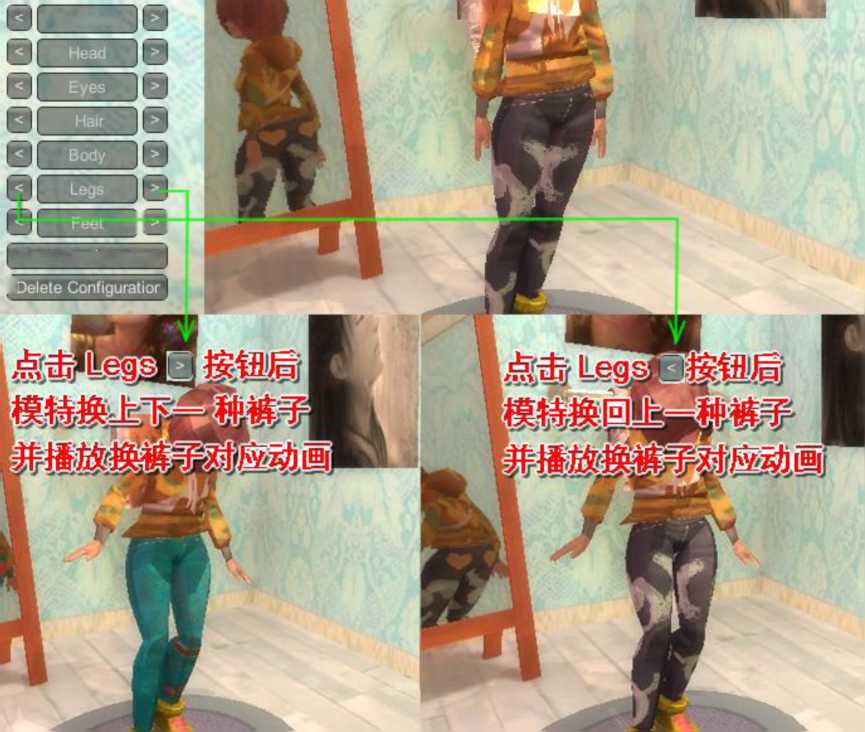
Save Configuration

Delete Configuratior 驗1\_：1丨观钮丨扁

图**4-8**发型更换的功能 **Figure4-8 The function of change the hair**

在“裤子”菜单选项中，点击“>”按钮更换模特裤子，点击返回上 个裤子，模特换装后会播放换裤子的动画，“裤子”更换功能如图4-9所示：

-41-

Character

11 I

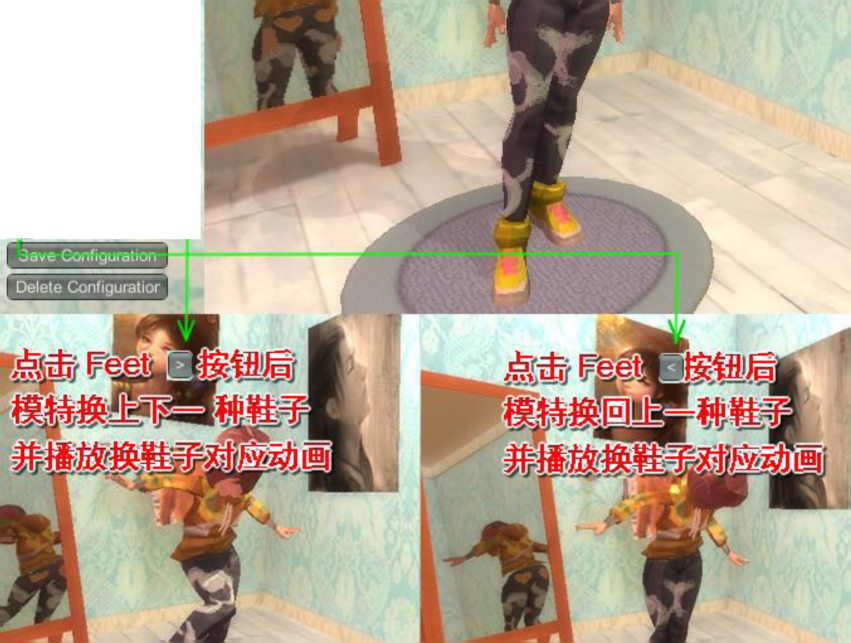
Save Configuration

fi

图**4-9**裤子更换的功能  
**Figure4-9 The function of change the Legs**

在“鞋子”菜单选项中，点击“>”按钮更换模特鞋子，点击返回上 种鞋子，模特换装后会播放换鞋子的动画，“鞋子”更换功能如图4-10所示:

-42-

****4.1.4**配置存储与删除功能的实现**

eeeaBolsl

imB

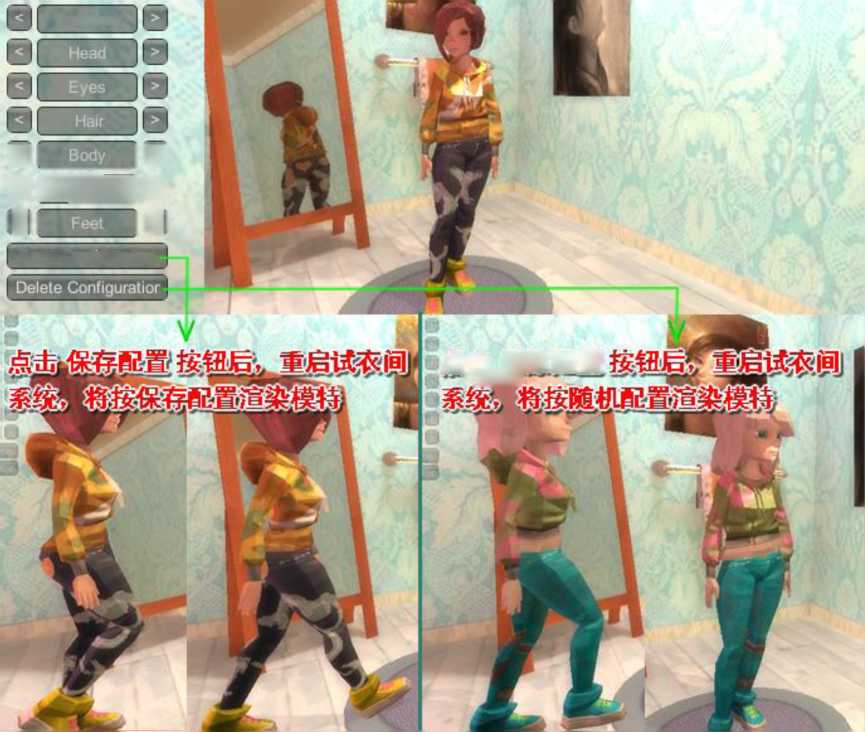
@@@@@@@1

Character

图**4-10**鞋子更换的功能 **Figure4-10 The function of change the Feet**

菜单用户界面的最后是两个选项，是“保存设置”和“删除设置”按钮，是 用来存储和删除用户保存的换装配置的。当用户点击“保存设置”按钮时，下一 次开启试衣间系统，进入试衣间的模特将按照保存设置进行渲染生成。当用户点 击“删除设置”按钮时，将删除保存的配置，下次重新开启试衣间系统，将随机 配置一套着装方案，按随机配置渲染生成模特。如图4-11所示：

-43-

Character

Ul IU

***mmEmm***

i：h [i

Save Configuration

点細除配豐

图4-11保存与删除配置功能

Figure4-11 Save and delete Configuratior

实现保存和删除配置功能的代码如下：

//保存设置

if (GUILayout.Button(MSave Configuration"))

PlayerPrefs.SetString(prefName, generator.GetConfig());

//删除设置

if (GUILayout.Button("Delete Configuration"))

PlayerPrefs.DeleteKey(prefName);

遵循GUILayout.Button组件的一般实现形式，将其作为if语句的判断条件， 当用户点击了按钮，则该条件为true,执行保存或删除的操作；否则该条件为

false,不执行任何操作。

以上所有代码都写在OnGUI()函数重载中，该函数继承自Unity系统类 MonoBehaviour，Unity引擎可以保证在程序运行后渲染每一帧时，OnGUI()函数

都会被调用，从而绘制出整个菜单。本系统的用户界面就是通过这样的机制显示 出来的。

-44-**4.2**核心模块的实现

4.2.1数据文件管理器模块的实现

在真实的用于商业目的的系统中，应该会有丰富的模型数据，这种情况下可 以考虑使用关系型数据库或者普通的文件系统来管理和存储数据。由于本系统是 一个示例程序，模特可选的模型数据比较有限，基于简化的目的，模型文件采用 文件管理的形式存储数据。

根据“3. 2. 1. 1”节的分析，本系统把模特的模型数据分为两部分，即共 享模型数据和非共享模型数据，两部分数据分开存储。

共享模型数据包括模特的模型数据中可以被共享使用的部分，即模特的整套 骨骼和动画效果。由于可相互替换的模特上下身模型都是基于同一套骨骼制作， 所以骨骼在模特换装过程中是保持不变的，而动画效果与模特的骨骼又是绑定的 关系，所以只要不更换模特，无论怎样搭配服装，使用的都是同一套骨骼和动画， 因此可以将这两部分数据提取出来，单独存储。数据文件管理器程序流程图如图 4-12所示：

-45-

图4-12文件管理器程序流程图

具体实现代码如下：



Figure4-12 The flow chart of create assetbundles

//在Unity引擎中实例化模特

GameObject characterClone = (GameObject)Object.Instantiate(characterFBX);

//设置动画播放的规则，即使模特在屏幕外也照常播放 foreach (Animation anim in

characterClone.GetComponentsInChildren<Animation>Q) anim.animateOnlyIfVisible = false;

//丢弃不需要存储的数据 foreach (SkinnedMeshRenderer smr in

characterClone.GetComponentsInChildren<SkinnedMeshRenderer>())

Object.DestroyImmediate(smr.gameObject);

//将数据存储为预制件prefab，然后输出成文件

-46-characterClone.AddComponent<SkinnedMeshRenderer>();

Object characterBasePrefab = GetPrefab(characterClone, "characterbase"); string path = AssetbundlePath + name + M\_characterbase.assetbundleM; BuildPipeline.BuildAssetBundle(characterBasePrefab, null, path, BuildAssetBundleOptions.CollectDependencies);

AssetDatabase.DeleteAsset(AssetDatabase.GetAssetPath(characterBasePrefab));

代码首先在Unity引擎中实例化模特对象，该对象包含人物的骨骼、动画和 模型材质数据。由于我们只需要骨骼和动画数据，所以在将其输出为文件之前， 先将材质数据丢弃，并设置好动画的播放参数，然后将其存储为预制件prefab。 由于骨骼和动画数据在程序中是共享使用，而且用户可能会多次加载同一个模 特，那么程序中就可能多次实例化同一个对象，为了不过多占用内存空间，使用 预制件prefab可以保证该资源在程序运行过程中只被实例化一次，只占用一个对 象的内存空间，这样可以提高程序运行效率。最后将生成的预制件prefab输出为 文件，如果以模特“female”为例，则本程序将该共享数据的文件名定义为 female\_characterbase.assetbundle，以供程序的其他模块使用。

非共享模型数据包括可相互替换的模型部件数据，每个模特都有多套身体各 个部位的非共享模型部件数据，以供用户进行选择和搭配。每个部件的一套数据 存储在单个文件中，多套数据分别存储在多个文件中，这样保证文件管理器的可 扩展性，如果需要添加一套新的模型数据到文件管理器中，只需要生成相关文件， 并将文件放在指定的路径下，而无需更新整个文件管理器。具体实现代码如下：

//收集模型数据使用到的所有材质 List<Material> materials =

EditorHelpers.CollectAll<Material>(GenerateMaterials.MaterialsPath(characterFBX)

);

//把模特身体各部分的模型数据分别输出 foreach (SkinnedMeshRenderer smr in

characterFBX.GetComponentsInChildren<SkinnedMeshRenderer>(true))

{

List<Object> toinclude = new List<Object>();

//将该部位的渲染器存储为预制件prefab，存入toinclude对象 GameObject rendererClone =

(GameObject)EditorUtility.InstantiatePrefab(smr.gameObject);

GameObject rendererParent = rendererClone.transform.parent.gameObject; rendererClone.transform.parent = null;

Object.DestroyImmediate(rendererParent);

-47-

Object rendererPrefab = GetPrefab(rendererClone, "rendererobject"); toinclude.Add(rendererPrefab);

//筛选出与当前渲染器相关的材质，存入toinclude对象 foreach (Material m in materials)

if (m.name.Contains(smr.name.ToLower())) toinclude.Add(m);

//存储与当前渲染器相关的骨骼的名字，存入toinclude对象 List<string> boneNames = new List<string>(); foreach (Transform t in smr.bones) boneNames.Add(t.name);

string stringholderpath = "Assets/bonenames.asset";

StringHolder holder = ScriptableObject.CreateInstance<StringHolder> (); holder.content = boneNames.ToArray();

AssetDatabase.CreateAsset(holder, stringholderpath); oinclude.Add(AssetDatabase.LoadAssetAtPath(stringholderpath, typeof (StringHolder)));

//将toinclude对象输出文件

string bundleName = name + "\_" + smr.name.ToLower(); path = AssetbundlePath + bundleName + ".assetbundle"; BuildPipeline.BuildAssetBundle(null, toinclude.ToArray(), path, BuildAssetBundleOptions.CollectDependencies);

}

代码中按照自定义的模特身体划分规则，即“面部”、“眼睛”、“头发”、 “上衣”、“裤子”和“鞋子”这六部分分别提取相关的模型数据进行存储。需 要存储的模型数据包括对应部位的模型网格、材质和骨骼名字。需要特别说明的 是，在Unity引擎中，模型网格和相应的骨骼是绑定的，如果将模型网格和骨骼 数据分别单独保存，则必须先将它们之间的绑定解除，当把分离后的数据再分别 从文件载入程序后，模型网格就无法得知对应的骨骼数据信息了。解决这个问题 的办法是，将与模型网格对应的骨骼名字一一保存起来，并把这部分信息也输出 到文件中，当程序将模型网格和骨骼数据加载后，可以通过骨骼名字检索到相应 的骨骼数据，从而实现分离数据的再组合。以输出模特“female”的“面部”为 例，输出文件的文件名是female\_face-1.assetbundle，统一文件命名规则是为了让 程序便于读取。

模型数据生成完毕后，就可以构造最终的文件管理器文件，该文件作为文件 管理器的入口，通过它程序可以找到每一项模型数据的位置。具体实现代码如下：

//模型数据集合

-48-

List<CharacterElement> characterElements = new List<CharacterElement>();

//为每一项模型数据匹配其入口

string[] assetbundles = Directory.GetFiles(CreateAssetbundles.AssetbundlePath); string[] materials = Directory.GetFiles("Assets/characters", "\*.mat", SearchOption.AllDirectories); foreach (string material in materials)

{

foreach (string bundle in assetbundles)

{

FileInfo bundleFI = new FileInfo(bundle);

FileInfo materialFI = new FileInfo(material); string bundleName = bundleFI.Name.Replace(".assetbundle", ""); if (!materialFI.Name.StartsWith(bundleName)) continue; if (!material.Contains("Per Texture Materials")) continue;

characterElements.Add(new CharacterElement(materialFI.Name.Replace(".mat", ""),

bundleFI.Name));

break;

}

}

//将模型数据集合保存在CharacterElementHolder对象中 CharacterElementHolder t =

ScriptableObject.CreateInstance<CharacterElementHolder> (); t.content = characterElements;

//生成文件管理器入口文件

string p = "Assets/CharacterElementDatabase.asset";

AssetDatabase.CreateAsset(t, p);

Object o = AssetDatabase.LoadAssetAtPath(p, typeof(CharacterElementHolder)); BuildPipeline.BuildAssetBundle(o, null, CreateAssetbundles.AssetbundlePath + "CharacterElementDatabase.assetbundle");

代码中characterElements是模型数据集合，需要为每一项非共享的模型数据 匹配对应的文件名，这样模型数据才能被正确加载。因此本程序中，同一模型数 据所使用的不同材质文件的文件名必须以模型数据的名字开头，以模特“female” “眼睛”的不同材质为例，黑眼睛材质文件名为female\_eyes\_black.mat，蓝眼睛 材质文件名则为female\_eyes\_black.mat。符合这样命名规则的材质才能成功生成 相应的模型数据，并添加进文件管理器。最后，输出的文件管理器入口文件为 CharacterElementDatabase.assetbundle。

4.22数据字典模块的实现

-49-

根据3. 2. 1. 2小节的分析，为了能够更加快速地检索文件管理器中的模 型数据，本程序在运行之初就将非共享模型数据载入内存，进行分类和索引，以 提高程序的运行效率。载入内存的只是模型数据的名称和文件路径等字符信息， 其实并不包含真实的渲染器和材质等模型数据，所以对内存空间的占用非常小， 同时载入过程的性能开销也很小。在程序与用户进行交互之前，数据字典模块就 己经构造完成，这样就可以在时间和空间开销都很小的前提下，对用户的换装操 作进行最大程度的优化，提高系统的反应速度。其构造过程的实现代码如下：

//初始化文件管理器

database = new WWW(AssetbundleBaseURL + "CharacterElementDatabase.assetbundle");

CharacterElementHolder ceh = (CharacterElementHolder) database.assetBundle.mainAsset;

// 初始化数据字典

sortedElements = new Dictionary<string, Dictionary<string, List<CharacterElement>>>();

//构造数据字典

foreach (CharacterElement element in ceh.content)

{

string[] a = element.bundleName.Split('\_'); string character = a[0];

string category = a[1].Split('-')[0].Replace(M.assetbundleM, ""); if (!availableCharacters.Contains(character)) availableCharacters.Add(character); if (!sortedElements.ContainsKey(character))

sortedElements.Add(character, new Dictionary<string, List<CharacterElement>>()); if (!sortedElements[character].ContainsKey(category)) sortedElements[character].Add(category, new List<CharacterElement>()); sortedElements[character][category].Add(element);

}

程序首先读取文件管理器入口文件“ CharacterElementDatabase.assetbundle ”

获得模型数据名称和文件路径等信息。然后开始解析模型数据名称，按照模特、 部位的划分规则将其分类，在这个过程中，程序根据模型数据的名称以确定该数 据属于某位模特的某个部位的第几套数据，归类后再存储在Dictionary类型中。 需要使用时，以模特“female” “面部”的第一套数据为例，只用写 sortedElements[“female”][“face”][0]这样一句调用就可以了，同时因为Dictionary 类型的特点，其运行效率也非常高。

-50- 4.2.3核心函数的实现

实现了数据文件管理器和数据字典这些程序的基础设施之后，就可以开始实 现系统的核心功能函数。本系统是一个3D试衣间，由此其核心功能就体现在更 换模特和更换服装上。

用户在使用本系统时，一般会首先选择一个与自己性别相同的模特进行换 装。下面就介绍选择和更换模特的具体实现： string character = null;

for (int i = 0; i < availableCharacters.Count; i++)

{

if (availableCharacters[i] != currentCharacter) continue;

//更换下一个模特 if (next)

character = i < availableCharacters.Count - 1 ? availableCharacters[i + 1]: avai l abl e Characters [0 ];

//更换上一个模特 else

character = i > 0 ? availableCharacters[i - 1]: availableCharacters[availableCharacters.Count - 1]; break;

}

//随机搭配服装 PrepareRandomConfig(character);

为了实现更换模特的功能，程序中创建了一个模特数组，以记录所有可选的 模特，当用户点击按钮更换模特时，程序就在数组中选择上一个或下一个模特， 然后通过PrepareRandomConfig函数为其随机搭配一套服装，以模特“female” 的“上衣”为例，函数在列表sortedElements[“female”][“body”]中按列表项的下

标随机选择一项作为模特上身的服装，身体其他部分以此类推即可。

当用户确定了模特之后，就可以开始为模特搭配服装了。下面是更换服装的 具体实现：

//可选服装列表

List<CharacterElement> available = sortedElements[currentCharacter][catagory]; CharacterElement element = null; for (int i = 0; i < available.Count; i++)

-51- {

if (available[i] != currentConfiguration[catagory]) continue;

//选择下一套 if (next)

element = i < available.Count - 1 ? available[i + 1] : available[0];

//选择上一套 else

element = i > 0 ? available[i - 1] : available[available.Count - 1]; break;

}

//将选择的服装存入当前服装配置 currentConfiguration[catagory] = element;

//根据当前服装配置生成人物 Generate。；

由于己经将模型数据按照不同身体部位分类存储在数据字典sortedElements 中，所以程序可以简洁而快速地检索出可供选择的服装列表，然后根据用户点选 的按钮来确定更换列表中的上一套还是下一套服装，并将其存入模特当前的服装 配置currentConfiguration中，最后根据当前配置生成并渲染模特，这样就实现了 为模特换装的功能。

模特的生成和渲染是由Generate函数实现，其具体代码如下：

public GameObject Generate(GameObject root)

{

//将模特身体各部分进行组合

List<CombineInstance> combineInstances = new List<CombineInstance>(); List<Material> materials = new List<Material>();

List<Transform> bones = new List<Transform>();

Transform[] transforms = root.GetComponentsInChildren<Transform>(); foreach (CharacterElement element in currentConfiguration.Values)

{

SkinnedMeshRenderer smr = element.GetSkinnedMeshRenderer();

//组合材质数据

materials.AddRange(smr.materials);

//组合网格渲染器

for (int sub = 0; sub < smr.sharedMesh.subMeshCount; sub++)

{

CombineInstance ci = new CombineInstance();

-52-

ci.mesh = smr.sharedMesh; ci.subMeshIndex = sub; combinelnstances.Add(ci);

}

//组合骨骼

foreach (string bone in element.GetBoneNames())

{

foreach (Transform transform in transforms)

{

if (transform.name != bone) continue;

bones.Add(trans form);

break;

}

}

Obj ect.De stroy(smr. game Obj ect);

}

//将组合后的模特对象渲染并返回

SkinnedMeshRenderer r = root.GetComponent<SkinnedMeshRenderer>(); r.sharedMesh = new Mesh();

r.sharedMesh.CombineMeshes(combineInstances.ToArray(), false, false); r.bones = bones.ToArray(); r.materials = materials.ToArray(); return root;

}

Generate函数是本系统的核心函数，它包含了模特生成过程的具体实现。按 照前面章节的阐述，整个模特由“面部”、“眼睛”、“头发”、“上衣”、“裤 子”和“鞋子”六大部分组成，每一部分都有各自的渲染器、材质和骨骼数据， 因此在最终渲染时，Unity引擎会将这六部分独立对待、分别渲染，即一共要渲 染六次，然后将六次渲染结果叠加，虽然最终的叠加结果是正确的，但这样显然 会降低程序的运行效率。模特本来是一个整体，只有一套渲染器、材质和骨骼数 据，但是由于需要实现换装的程序功能，将这个整体划分为六个可替换的部分， 于是一套数据被切分为六套数据，渲染过程变得相对低效了。但是这六套数据实 际上是可以还原为原始的一套数据的，因此在渲染发生之前，程序先将模特的身 体重新组合，将其还原为原始的一套渲染器、材质和骨骼数据，然后再进行渲染， 这种情况下，Unity引擎将组合后的模特视为一个整体，只需要渲染一次就可以 把整个模特渲染出来，这样无疑可以加快渲染速度，提高程序运行效率。

-53-

4.3 3D试衣间的测试

4.3.1测试原则和计划

3D试衣间系统的软件测试原则和计划如下：

1、 软件测试按照系统需求为基准进行；

2、 系统测试包括功能、可用性等方面的测试；

3、 本测试采用手动黑盒测试，不会采用测试工具，程序测试使用白盒测试；

4、 对发现错误集中的模块必须进行重点测试。

4.3.2测试环境

硬件环境：内存8G，CPUi7-3630QM，主频2.40GHz，硬盘1TG 软件环境：操作系统Windows7 4.3.3测试用例

3D试衣间测试用例如下表4-1所示：

表4-1 3D试衣间测试用例表

Table 4-1 3D fitting room functional test table

测试项目 描述/操作 期望功能结果 真实可用性结果备注

1、 用户菜单试衣间系统开启 开启程序后渲染生成菜单界面 V成功图4-1

界面生成

2、 场景、人物试衣间系统开启 开启程序后渲染生成场景和人物模特V成功图4-3 模特渲染

3、 模特走入动播放模特走入动画 人物渲染成功后播放走入试衣间动画V成功图4-3 画渲染

4、性别选择 点击Character右侧〉按钮更换不同性别人物模型 点击Character左侧<按钮返回上一个人物性别模型

V 成功 图 4-4

V成功图4-5

V成功图4-5

5、 上衣更换 点击巳ody右侧〉按钮 随机更换人物角色上衣

点击Body左侧<按钮 返回上一件上衣

6、 换上衣动画更换上衣结束 播放更换上衣动画 渲染

-54-

北京工业大学工程硕士专业学位论文

4.3 3D试衣间的测试

4.3.1测试原则和计划

3D试衣间系统的软件测试原则和计划如下：

1、 软件测试按照系统需求为基准进行；

2、 系统测试包括功能、可用性等方面的测试；

3、 本测试采用手动黑盒测试，不会采用测试工具，程序测试使用白盒测试；

4、 对发现错误集中的模块必须进行重点测试。

4.3.2测试环境

硬件环境：内存8G，CPUi7-3630QM，主频2.40GHz，硬盘1TG 软件环境：操作系统Windows7 4.3.3测试用例

3D试衣间测试用例如下表4-1所示：

表4-1 3D试衣间测试用例表

Table 4-1 3D fitting room functional test table

测试项目 描述/操作 期望功能结果 真实可用性结果备注

1、 用户菜单试衣间系统开启 开启程序后渲染生成菜单界面 V成功图4-1

界面生成

2、 场景、人物试衣间系统开启 开启程序后渲染生成场景和人物模特V成功图4-3 模特渲染

3、 模特走入动播放模特走入动画 人物渲染成功后播放走入试衣间动画V成功图4-3 画渲染

4、性别选择 点击Character右侧〉按钮更换不同性别人物模型 点击Character左侧<按钮返回上一个人物性别模型

V 成功 图 4-4

V成功图4-5

V成功图4-5

5、 上衣更换 点击巳ody右侧〉按钮 随机更换人物角色上衣

点击Body左侧<按钮 返回上一件上衣

6、 换上衣动画更换上衣结束 播放更换上衣动画 渲染

-54-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7、脸型更换 | 点击Head右侧〉按钮 | 随机更换人物角色脸型 | V成功图4-6 | |
|  | 点击Head左侧<按钮 | 返回上一个脸型 |  |  |
| 8、眼球更换 | 点击Eyes右侧〉按钮 | 随机更换人物角色眼睛颜色 | V成功 | 图4-7 |
|  | 点击Eyes左侧〈按钮 | 返回上一种眼睛颜色 |  |  |
| 9、发型更换 | 点击Hair右侧〉按钮 | 随机更换人物发型以及颜色 | V成功 | 图4-8 |
|  | 点击Hair左侧<按钮 | 返回上一种发型 |  |  |
| 10、裤子更换 | 点击Legs右侧>按钮 | 随机更换人物角色裤子 | V成功 | 图 4-9 |
|  | 点击Legs左侧<按钮 | 返回上一种裤子 |  |  |
| 11、换裤子动  画渲染 | 更换裤子结束 | 播放更换裤子动画 | V成功图4-9 | |
| 12、鞋子更换 | 点击Feet右侧>按钮 | 随机更换人物角色鞋子 | V成功 | 图 4-10 |
|  | 点击Feet左侧<按钮 | 返回上一种鞋子 |  |  |
| 13、换鞋动画 渲染 | 更换鞋子结束 | 播放更换鞋子动画 | V成功图4-10 | |
| 14、保存配置 | 点击Save Configuraton按钮重启系统将按保存配置渲染人物 | | V成功 | 图 4-11 |
| 15、删除配置 | 点击Delete Configuration按钮重启系统将按随机配置渲染人物 | | V成功 | 图 4-11 |

4.3.4测试结果分析

经测试，3D试衣间系统没有明显的功能bug,可以正确执行各项用户操作。

4.4本章小结

本章主要对3D试衣间各个功能模块的实现方法、实现流程以及系统测试做 了说明。通过使用各种类、函数的运用实现了预期的结果。

结论

本文阐述的3D试衣间系统，通过使用Maya技术进行建模、绑定骨骼； iClone3DXchange以及lClone技术进行动画制作；C#技术调用Unity3D的自带类 GUILayout实现GUI界面布局；调用WWW类标识出各种资料地址；使用C# 技术的自带类Dictionary实现数据字典的创建；调用CharacterElement类存储模 特模型构成的基本元素；调用CharacterGenerator类组装和生成模特模型；使用 Unity3D引擎技术进行系统的综合开发，最终实现了 3D试衣间模特换装功能， 解决了用户自由搭配服饰，美化虚拟化身，彰显个性的心里需求。系统经过测试， 界面友好，可由Unity3D引擎发布到几乎所有商用平台，具有良好的实用性以及 商用价值。

本文实现的3D试衣间系统具有的特点为：

(1) 模特使用3D技术建模，通过试衣镜功能使模特360度可视，具有较 强的视觉冲击力。

(2) 换装功能实现时会播放模特换装动画，生动、富于趣味性。

(3) 通过创建数据字典，存储数据集合资源包，提供换装的基本元部件， 实现换装功能。具有采集FBX模型、动画文件的基本数据的功能，易于材质更 新与模特扩展。

(4) 动画制作使用iClone技术与Maya技术相结合的方法，动画制作效率

高。

此次3D试衣间系统的设计中，由于时间以及个人能力因素，整体代码还是 相对有些凌乱，代码量也相对较多，这些问题随着经验的积累，通过优化是可以 解决的，限于精力有限，所做研宄还较浅显，有待继续深入研宄。3D试衣间在 网络销售服装方面如果实行，需要对服装进行精细建模，精心绘制贴图，虽然技 术可实现，但成本较高，只限于具有足够投资的经营项目，若要进行广泛低成本 推广还需对服装建模贴图方法进行继续深入研宄。

-57-