

广州大学华软软件学院

# 本科毕业设计任务书

设计题目 VR 教育类游戏“BOOM”中分子

结构与液体仿真的设计与实现

系 别 游戏系

专 业 数字媒体技术

班 级 16 级数字媒体技术 1 班

学 号 1640622121

学生姓名 陈梓浩

指导教师 秦胜伟

下发时间： 2019 年 10 月 20 日

# 毕业设计须知

- 1、认真学习和执行广州大学华软软件学院学生毕业论文(设计)工作管理规程;
- 2、努力学习、勤于实践、勇于创新,保质保量地完成任务书规定的任务;
- 3、遵守纪律,保证出勤,因事、因病离岗,应事先向指导教师请假,否则作为缺席处理。凡随机抽查三次不到,总分降低 10 分。累计缺席时间达到全过程 1 / 4 者,取消答辩资格,成绩按不及格处理;
- 4、独立完成规定的工作任务,不弄虚作假,不抄袭和拷贝别人的工作内容。否则毕业设计成绩按不及格处理;
- 5、毕业设计必须符合《广州大学华软软件学院普通本科生毕业论文(设计)规范化要求》,否则不能取得参加答辩的资格;
- 6、实验时,爱护仪器设备,节约材料,严格遵守操作规程及实验室有关制度。
- 7、妥善保存《广州大学华软软件学院本科毕业设计任务书》。
- 8、定期打扫卫生,保持良好的学习和工作环境。
- 9、毕业设计成果、资料按规定要求装订好后交指导教师。凡涉及到国家机密、知识产权、技术专利、商业利益的成果,学生不得擅自带离学校。如需发表,必须在保守国家秘密的前提下,经指导教师推荐和院领导批准。

课题名称	VR 教育类游戏“BOOM”中分子结构与液体仿真的设计与实现
完成日期： 2020 年 3 月 30 日	
<p>一、题目来源及原始数据资料：</p> <p>题目来源：自拟</p> <p>原始数据资料：</p> <p>(1) 参考相关资料及文献</p> <p>(2) 这是一款 VR 教育类游戏，其中主要功能是深度学习在 VR 虚拟世界中的应用，以及实验瓶中的液体仿真效果。氢分子、氧分子等各种分子的组成识别采用卷积神经网络完成，液体仿真技术主要考虑化学液体的状态模拟。</p> <p>市面上关于卷积神经网络技术的应用一般在产品中广泛存在，例如面部识别。但直接将卷积神经网络应用到游戏中的却尚未出现。液体仿真算法大多是基于 SPH 流体算法或粒子模拟来实现对液体的仿真。为了减少 GPU 的计算负担，本文液体仿真采用 Shader 实现。</p>	

## 二、毕业设计要求：

1. 开发工具：Unity3D
2. 程序有良好的设计风格和代码规范
3. 了解与游戏程序相关的软件技术
4. 程序能正确地运行
5. 进行必要的调研和资料搜集、文献阅读；
6. 要有完整的开发文档
7. 有游戏玩法详细说明文档
8. 每周有开发进度报告

(以下为你的游戏项目的具体要实现的目标)

- 1、收集分子模型的结构
- 2、数据集采集
- 3、CNN 网络训练
- 4、Unity 中的 CNN 网络识别框架
- 5、摄像机跟随处理拍照
- 6、实现用户提取组成分子结构的模型识别
- 7、完成关卡系统和排行榜系统
- 8、UI 界面
- 9、仿真液体效果的实现

## 三、进度安排、应完成的工作量：

1. 2019 年 10 月 20 日前，下达任务书。
2. 2019 年 10 月 26 日前，上交任务书，完成毕业论文开题，开始游戏项目设计。
3. 2019 年 11 月 1 日前，开始游戏项目第一版，开始撰写毕业论文。
4. 2020 年 1 月 20 日前，完成游戏项目第一版，提交毕业论文的初稿。
5. 2020 年 3 月 15 日前，提交游戏最终版。
6. 2020 年 3 月 20 日前，提交毕业论文的第二版。

7. 2020 年 4 月 6 日前，提交毕业论文的完成稿。
8. 2020 年 4 月 16 日前，提交毕业论文的打印稿。
9. 2020 年 4 月 18-4 月 20 日，组织毕业论文（设计）答辩及成绩评定。

（以下对自己的工作进度进行详细说明）

1. 2019 年 11 月 1 日-2019 年 11 月 3 日，收集分子模型的结构。
2. 2019 年 11 月 4 日-2019 年 11 月 6 日，数据集采集。
3. 2019 年 11 月 7 日-2019 年 11 月 8 日，CNN 网络训练。
4. 2019 年 11 月 9 日-2019 年 11 月 15 日，Unity 中的 CNN 网络识别框架。
5. 2019 年 11 月 16 日-2019 年 11 月 18 日，摄像机跟随处理拍照。
6. 2019 年 11 月 19 日-2019 年 11 月 22 日，实现用户提取组成分子结构的模型识别。
7. 2019 年 11 月 23 日-2019 年 11 月 29 日，完成关卡系统和排行榜系统。
8. 2019 年 11 月 30 日-2019 年 12 月 3 日，UI 界面。
9. 2019 年 12 月 4 日-2020 年 1 月 12 号，仿真液体效果的实现
10. 2020 年 1 月 20 日，完成第一版项目，完成毕业论文初稿。
11. 2020 年 3 月 15 日，完成游戏最终版。
12. 2020 年 3 月 20 日，完成毕业论文第二版。
13. 2020 年 4 月 5 日，完成并提交毕业论文的完成稿。

应完成的工作量：

- 1、完成代码行数：5000 行以上
- 2、毕业论文字数：8000 字以上

#### 四、主要参考文献

- [1] 喻俨，莫瑜.深度学习原理与 TensorFlow 实践[M]. 北京：电子工业出版社，2017. 71-121
- [2] 曼普里特·辛格·古特.TensorFlow 神经网络编程[M]. 北京：机械工业出版社，2018. 65-102

- [3] 胡良云.HTC Vive VR 游戏开发实战[M]. 北京：清华大学出版社，2017. 230-280
- [4] 刘刚，孙文涛.unity 官方案例精讲[M]. 北京：中国铁道出版社，2015. 40-50
- [5] 谭小辉, 万旺根, 黄炳, 崔滨. 基于 SPH 的三维流体模拟[J]. 计算机应用与软件, 2009, 26(12):222-224+258.
- [6] 欧阳劲夫, 龚捷. 基于 Unity3D 的液体面波浪模拟[J]. 电脑知识与技术, 2020, 16(05):219-220.
- [7] 雷国伟. 图像特征的 CNN 提取方法及其应用[J]. 计算机工程与应用, 2004, (14), pp. 204-206+216CNKI
- [8] 洪睿, 康晓东, 郭军, 李博, 王亚鸽, 张秀芳. 基于复杂网络描述的图像深度卷积分类方法[J]. 计算机应用, 2018, 38(12):3399-3402
- [9] 张家欢. 基于 GPU 元球造型的液体滴模拟[D]. 武汉科技大学, 2011.

指导教师（签名）：

系（教研室）主任（签名）：

**摘 要** 近年来，图像识别技术作为计算机视觉领域的一个重要分支，已被广泛应用于各个领域，但在教育类游戏领域上的应用却较为少见。在仿真教学上，大多数液体仿真算法，性能上消耗较大。针对以上两种问题，本文介绍了 VR 化学实验教育类游戏“BOOM”分子结构学习和液体仿真两大模块的设计思路，主要探讨了虚拟现实技术（Virtual Reality，简称 VR）与深度学习技术（Deep Learning，简称 DL）在教育游戏开发中的实现过程。其中，本文重点阐述了卷积神经网络技术（Convolutional Neural Networks，简称 CNN）在 Unity3D 中的构建和识别以及液体仿真效果的实现。

**关键词** 虚拟现实技术；液体仿真；卷积神经网络；

**ABSTRACT** In recent years, as an important branch of computer vision, image recognition technology has been widely used in various fields, but its application in the field of educational games is relatively rare. However, in the simulation teaching, most of the liquid simulation algorithms cost a lot in performance. According to the two problems, this paper introduces the design ideas of "BOOM" molecular structure learning and liquid simulation modules in VR chemistry experimental education game, and mainly discusses the realization process of Virtual Reality (VR) and Deep Learning (DL) in educational game development. Among them, this paper focuses on the construction of Convolutional Neural Networks (CNN) in Unity3D and the realization of recognition and liquid simulation effects.

**KEY WORDS** VR; Liquid Simulation; CNN



# 目录

摘要 .....	I
ABSTRACT .....	II
第 1 章 绪论 .....	1
第 2 章 游戏需求分析 .....	2
2.1 市场分析 .....	2
2.2 需求分析 .....	2
第 3 章 游戏概要分析 .....	4
3.1 总体方案设计 .....	4
3.2 模块设计 .....	5
3.3 模块划分 .....	6
第 4 章 详细分析 .....	8
4.1 化学分子结构学习设计 .....	8
4.1.1 数据集设计 .....	8
4.1.2 卷积神经网络设计 .....	9
4.1.3 化学分子结构模拟设计 .....	11
4.2 液体仿真设计 .....	14
4.1.1 液体 Shader 设计 .....	14
4.1.2 液体控制效果设计 .....	16
第 5 章 游戏测试 .....	19
5.1 测试目标 .....	19
5.2 测试过程 .....	19
参考文献 .....	25
致谢 .....	26

# 第 1 章 绪论

近几年来，随着科技的发展，VR 虚拟现实技术和图像识别技术不断成熟，并在医学、教育、游戏、娱乐、建筑、工业仿真等多个领域有着广泛的应用。所以，在游戏设计中引入 VR 技术已是市场趋势所向，使用 VR 技术能够使人更加沉浸在游戏中，体验游戏的乐趣。在教学应用上，通过 VR 技术开发出来的虚拟现实仿真应用也让游戏的趣味性渗入到教学中去，从而使学生能在教学实验中受益匪浅。但传统的教学评估实验中只是通过内置固定的模型去编写代码，从而对教学实验进行评分。相对于传统方式而言，图像识别技术能够更加智能地对教学实验进行评分。所以，VR 教育类游戏“BOOM”的分子结构游戏中采用图像识别的技术，对游戏中拼装的模型进行识别，让图像识别、VR 技术与液体仿真设计相结合，使用户在游戏中有更加新颖的体验。

本论文主要针对 VR 教育类游戏“BOOM”的分子结构识别、游戏系统构造以及液体仿真设计进行详细的说明，包括了游戏的市场需求分析、概要分析、详细分析、游戏功能测试四个部分。其中，各部分要探讨的内容如下：

市场需求分析：主要对 VR 游戏市场进行分析。

概要设计分析：主要对游戏核心的设计、各模块关键类、流程的设计进行分析。

详细分析：主要从代码实现角度和模块设计思路分析游戏两大核心模块的实现。

游戏功能测试：本部分是从游戏设计过程中的一些测试方法、测试用例和结果的讨论。

## 第 2 章 游戏需求分析

### 2.1 市场分析

自从 2016 年以来，VR 的焦点逐渐从硬件转到软件上，VR 线下体验主要分为两种形式，一种是以 VR 设备为基础的轻量化的线下体验店，另一种则是拥有一定规模主题公园。VR 应用在扩大规模上仍然有一定的阻碍，但随着 VR 技术逐步成熟，VR 技术逐渐受到了资本青睐和广大用户的喜爱。其中，比较有代表性的就是教育行业，因为教育行业是一个国家顶力支持的行业。大多数教师现在仍然采用的是单一化的传统教育的形式，而 VR 技术上的优势可大大提升传统教育上的趣味性，增强学生对知识的理解和记忆。近年来，VR 硬件的价格随着 VR 技术的推动骤降，大部分高校完全能支撑起这部分费用，而在 2019 上半年 VR 软件市场规模首次超越硬件市场规模，在市场产业链上已经形成了小规模雏形。可见，VR 技术与教育行业的结合是教育行业发展的一种新趋势，而本文的“BOOM”是基于 VR 技术与图像识别技术相结合而设计的 VR 教育类严肃游戏，是对 VR 教育类产品的一种创新。

### 2.2 需求分析

目前市场上的 VR 游戏层出不穷，但技术实现方法却比较传统化。在技术上没有创新点，所以本文的 VR 游戏“BOOM”是基于 VR 技术与图像识别技术所开发的，主要针对需要初级阶段化学实操练习的学生。本文所实现的功能主要分为两个模块：分子结构学习模拟和液体仿真。分子结构学习模拟构建了玩家拼装原子模型的场景，玩家可以对拼装的模型进行识别，并在此加入了排行榜系统和关卡系统，使整个游戏过程更具趣味性。液体仿真则是从液体模拟的角度出发，模拟瓶内液体的效果。所以本文的主要需求功能具体如下：

1. 发布到 PC 端及其相关设置
2. VR 中原子模型的生成、组成及识别开发

3. VR 中液体的仿真效果
4. 游戏数据分数排行显示
5. 游戏关卡的解锁和模型组装的评分

## 第 3 章 游戏概要分析

### 3.1 总体方案设计

本游戏主要应用在化学实验教学方面，玩家可以在有限的时间里，通过组装原子模型来进行游戏。在游戏的过程中，不断地加深玩家对化学分子知识的印象，同时，激发玩家对相关化学知识的兴趣。

本文主要实现的是分子结构学习和液体仿真设计两个模块，分子结构学习场景中主要对玩家组装的原子模型进行图像识别。图像识别算法有 SIFT、CNN、Eigenface 等识别技术，但在众多技术中，CNN 的计算速度远远超过其他图像识别算法，并且 CNN 的应用领域十分的广阔。本文设计的游戏中“分子结构学习”场景采用 CNN 进行图像识别。

液体仿真模块主要模仿现实中液体流动和倾倒的效果。在大多数液体设计中，主要采取的是粒子模拟、3D 元球等技术技术，然后进行三维网格重建，这样实现的效果较为真实，但对于 CPU 的计算仍然十分费力，很容易造成游戏卡顿。所以液体仿真采用 Shader 设计实现的，因为 Shader 的计算主要在渲染管线进行，减小了游戏加载的负担，提升了游戏的流畅度。

玩家通过佩戴 VR 设备进行游戏场景选择，玩家可以通过碰触选择方块进入“分子结构学习”场景。其游戏操作流程是以玩家触碰方块为基准，玩家需要在有限的时间完成原子模型的组装并碰触识别方块，系统会根据玩家剩余的时间以及 CNN 评判的结果来计算最终的分数，然后录入排行榜系统中。如果玩家组装的模型识别分数超过 60 分，则显示游戏胜利，解锁下一个关卡模型。评分标准为 CNN 识别模型的准确度和剩余时间除以总的通关时间所得到的分数，两者各占 50%。游戏流程如图：

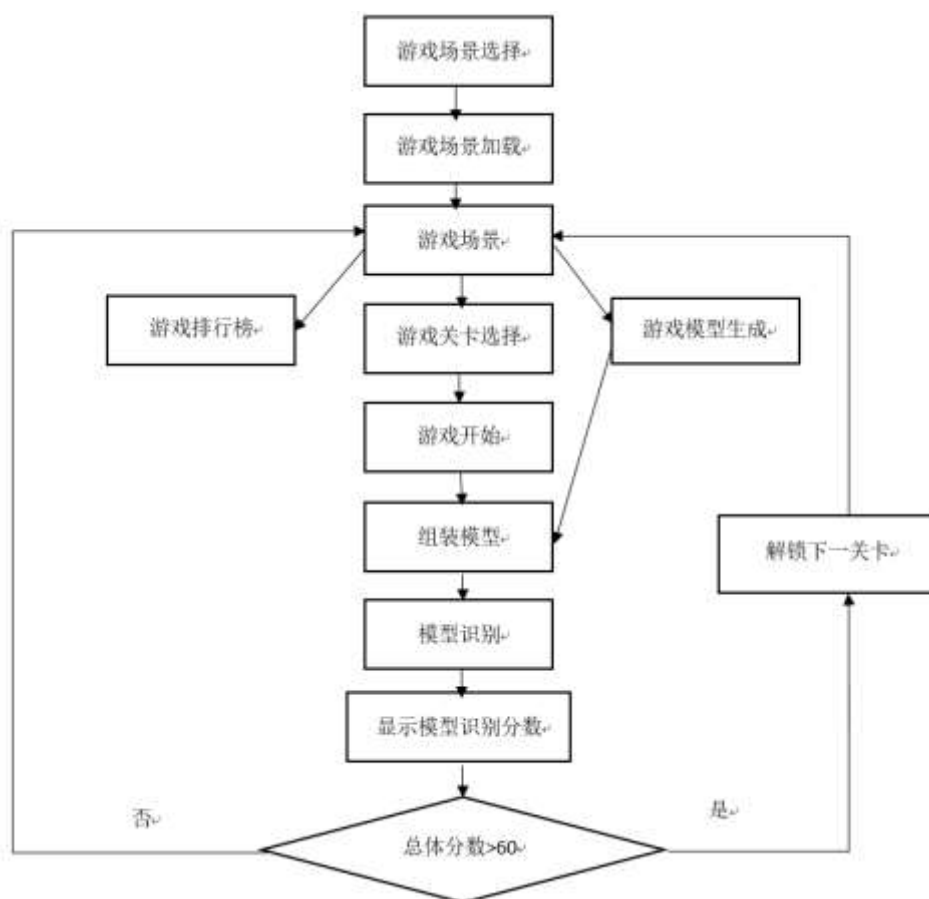


图 3.1 游戏流程图

### 3.2 模块设计

分子结构学习游戏主要以分子结构模拟为核心，并基于卷积网络训练，交互，数据存储几大模块设计的。卷积网络训练由于自身的独立性，不和其他模块产生联系。在众多编程语言中，Python 相对于 Unity3D 引擎对数学函数的倾向性高，并且集成了神经网络训练框架 Tensorflow，所以主要采用 Python 并导入 Tensorflow 的框架进行 CNN 的训练开发。分子结构学习模块是基于用户操作产生交互的。在设计之初，考虑到代码的重用性以及各大模块之间的独立性，采用开闭原则和接口隔离原则让模块间的接口尽可能单一化，并通过单例模式

GameManager 来完成模块与模块之间的对接，从而降低系统的耦合度。以下为分子结构学习游戏的模块设计图：

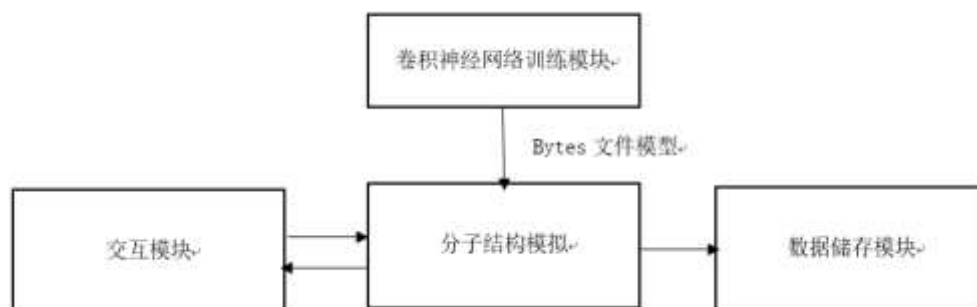


图 3.2 模块设计

Unity3D 的组件化避免了影响自身内部系统的流程，强迫开发者开发高内聚低耦合的组件。液体仿真在化学实验开发符合 Unity3D 组件化开发思想，所以液体仿真部分以组件化的形式开发，极大程度地提高代码的重用性，提升了开发效率。液体仿真需要比较逼真地模仿液体的效果，所以要计算液体的流动速度，液体的反射，液体面的倾倒高度，故采用 Shader 和 C#实现的。Shader 主要处理的是材质在渲染管线中顶点着色器和片元着色器的相关计算，C#主要计算液体倾倒的高度和抖动的幅度，从而达到液体仿真的效果。

### 3.3 模块划分

化学分子结构学习游戏的主要模块是分子结构模拟和数据存储模块。分子结构模拟的核心部分主要是图像识别模块，其大致可分为摄像拍摄、图片缩放、保存图片数据和图片识别这几部分功能，每一部分都有对应的脚本组件。

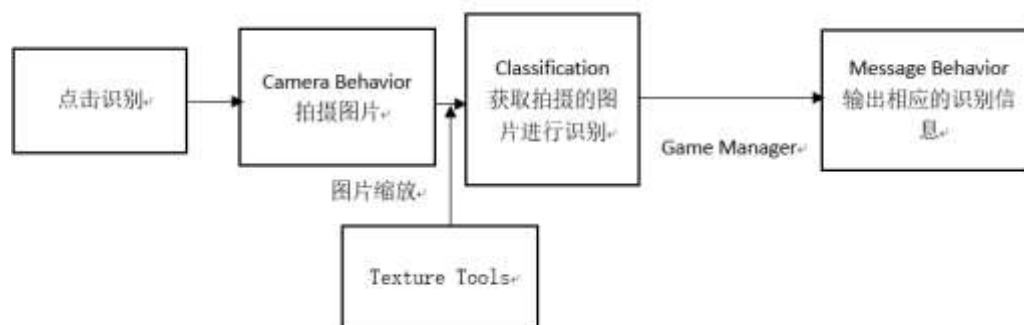


图 3.3 图像识别模块流程图

图片识别模块最重要的是图片数据如何处理和分类，Unity3D 提供了 Textrue2D 类，使得开发者可以通过坐标来设置图片的像素，本文通过 Color32[] 数组的变量来保存物体从世界空间转换到剪裁空间所形成的二维图片，缩放图片的大小与训练网络输入一样，然后在图片分类组件中进行预测，并传递预测模型的结果。具体的流程如图 3.3。

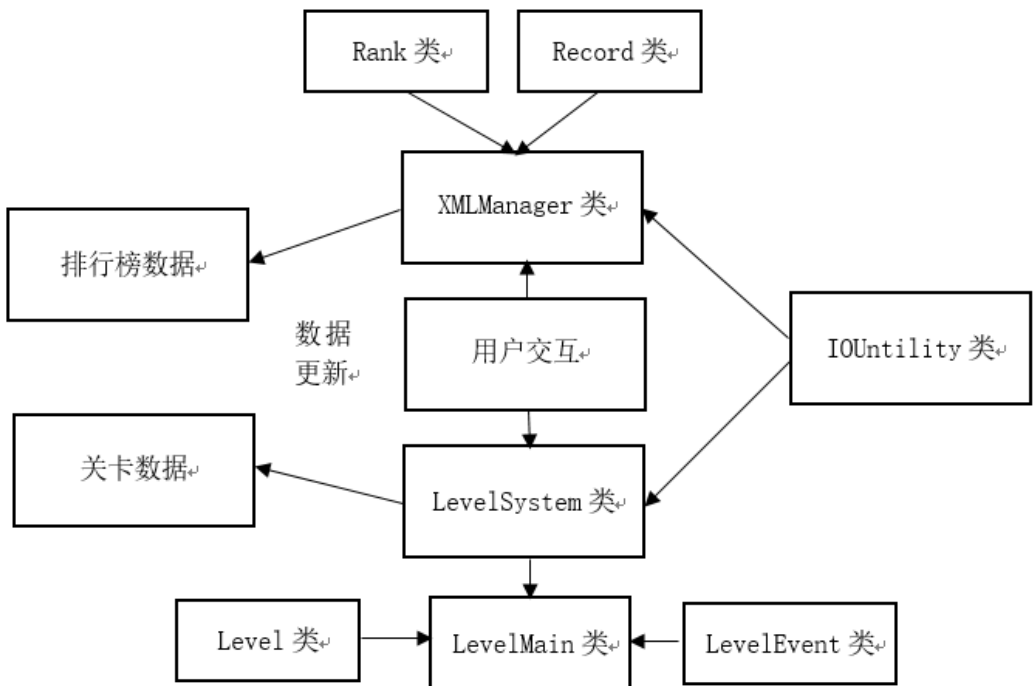


图 3.4 数据储存模块

数据储存模块主要是基于交互系统实现对排行榜系统和关卡系统数据的更新和存储。在这一模块中，XML 文件的节点数据、排行榜数据和关卡数据能在文件读写形成良好的对接，所以采取的是 XML 文件存储的方式。

关卡数据元素设计为：关卡的 ID、关卡的名字、关卡是否解锁和通关时间，这些数据都被抽象成关卡类的属性，然后通过解析 XML 文件从而进行数据的本地存储。在设计关卡系统时，LevelSystem 被设置成静态类，只进行对关卡读取和解锁两种功能的实现，LevelMain 负责对关卡数据的更新，LevelEvent 负责响应关卡消息，从而进行回调进行关卡选择。排行榜系统也采用类似的设计，但在基础数据方面有些区别，排行榜的数据设计为识别的模型和模型的分数，主要为玩家识别的分子模型以及分数，然后根据通过选择排序对玩家的游戏数据进行排行，再对排行榜进行更新。



## 第 4 章 详细分析

### 4.1 化学分子结构模拟设计

化学分子结构主要是对玩家完成的模型进行结构识别，然后根据剩余的时间和 CNN 的结构识别准确度对玩家本次操作进行评分。所以如何检测到玩家拼装的模型结构是否准确是化学分子结构模拟考虑的核心问题。本文在设计之初设置了关卡系统，为玩家拼装分子结构时起到导向的作用，但当玩家选择一个关卡去拼装相对应的原子模型时，CNN 系统会根据拍摄的图片数据，对玩家拼装的分子模型进行结构检测，主要是通过比对该模型结构在图片中呈现的数据进行结构识别，例如氢气模型在图片的中间，所占的像素数据肯定和氧气模型在图片中所占的像素数据不一样。本文的化学分子结构模拟正是通过 CNN 对所拍摄的图片数据进行结构识别，在根据用户游戏过程中剩余的时间对玩家本次操作进行评分。但当选择不对应的模型时，分子结构模拟系统则会根据玩家的模型评判出最相似的模型及分数。所以本文的化学分子结构的主要步骤可分为三个部分：如何构建数据集、如何设计卷积神经网络以及如何在 Unity 中进行分子结构模拟。



图 4.1 分子模型图

#### 4.1.1 数据集的设计

数据集的设计是卷积神经网络训练的基础。数据集制作不当的话，识别的准确率会大大降低，所以如何制作数据集是 CNN 训练模型关键的一部分。目前，大多数研究采用的是已有的数据集，但针对化学分子模型识别，就需要制作对应的数

数据集。CNN 通常对二维图片进行识别，但本文识别的模型是三维模型，采取传统二维图像的数据集制作明显不合适，所以需要重新设计制作数据集的方法。

在 CNN 输入接口中，输入多张图片数据形成张量，可以理解成多个图片构成的图片矩阵。本文的张量构成  $100 \times 100 \times 3 \times 64$ ，其中  $100 \times 100 \times 3$  是图片数据维度，图片数据处理成  $100 \times 100$  像素，主要是通过缩小图片的像素，减少 GPU 计算的负担，64 则是图片张数。CNN 会对输入张量进行卷积池化处理，但在处理过程中仍然是每一张图片进行特征压缩和降维，如经过一层卷积后张量会变成  $50 \times 50 \times 3 \times 64$ 。

本文的数据集在网络训练的时候仍然要转化为图片矩阵，即决定了本文数据集的制作仍是以图片的形式来进行网络训练，所以如何对三维模型进行图片制作时本文考虑的一个重点。考虑到数据集的制作要尽可能的保存三维模型特征以及适应模型在图片的所占位置不同的问题，所以本文采取的是多方向拍摄模型角度的数据集制作方法，具体思路如下：

1. 相机围绕着数据集数据进行 X, Y, Z 进行旋转
2. 模型围绕本身进行 X, Y, Z 轴自转
3. 每一帧对相机的角度和模型的角度进行更新，相机捕获图片。

上述数据集制作方法主要为了使相同角度下不同图片位置的模型和相同图片位置下不同角度的模型，尽可能地保留模型的特征。制作的数据集如下图：

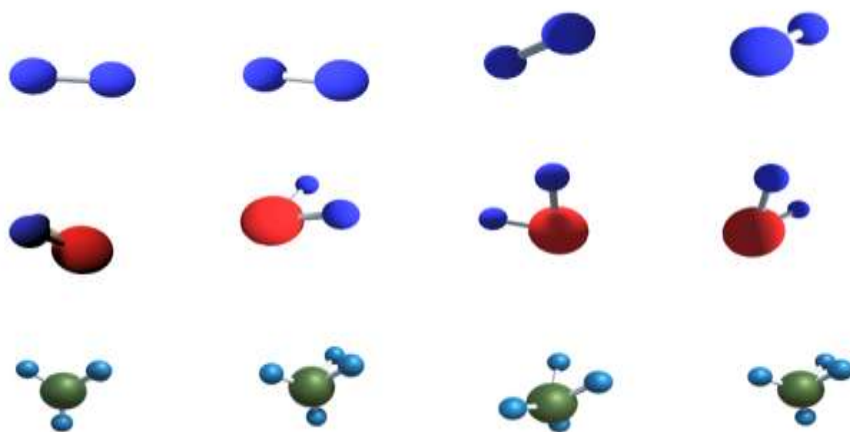


图 4.2 不同模型不同角度的二维图像

### 4.1.2 卷积神经网络设计

CNN 训练模块另一个核心部分就是神经网络层面上的设计。网络层数、卷积核大小、池化核大小和激活函数等参数的设计决定了 CNN 识别的性能与准确度，所以如何构建卷积神经网络也是 CNN 训练模块应考虑的一个核心部分。

在卷积神经网络中，网络层数越多意味着权重越多，过多的权重约束着网络识别准确度，很容易造成过拟合的现象。由于模型图片特征较为简单，所以构建网络的层数也相应减少。卷积神经网络的网络层主要分为卷积层，池化层和全连接层。

卷积层就是在原始的输入下，通过卷积核一个小区域一个小区域地进行特征提取。例如一张  $32 \times 32 \times 3$  的图片数据经过  $5 \times 5 \times 3$  的卷积核会被提取成  $28 \times 28 \times 1$  的图片数据，池化层则是对特征图进行压缩降维，主要有最大值池化和均值池化两种方式，这两种方式的差别是在池化核覆盖的区域上选择该区域的平均值或者最大值来代表该区域。全连接层则是连接所有上一层网络的所有特征并输送给下一层，最终输送给分类器。

CNN 的搭建是在 Tensorflow 的框架上搭建的。本文的 CNN 采用四层卷积层、四层池化层再加三层全连接层进行网络构建。在卷积层中，卷积核通常选取奇数卷积核尺寸，因为奇数卷积核尺寸有对称中心点，中心点左边和右边是对称的，上面和下面是对称的，而偶数卷积核则没有这样的中心点。这样方便进行特征选取。另外，卷积核过大容易导致特征丢失，所以卷积核的大小一般为  $5 \times 5 \times 3$ 。由于池化层需要进行降维操作，所以池化核的大小一般为  $2 \times 2 \times 3$ 。本文 CNN 的设计选取 ReLu 激活函数，因为 ReLu 函数能够保持梯度不衰减，从而缓解梯度消失问题，还能够很好地收敛模型，并提供给神经网络稀疏表达的能力。CNN 训练过程中，将预先保存好的数据集输入网络，经卷积层提取特征，经池化层进行降维，最后经全连接层进行反向传播算法，不断迭代，更新每一个神经元的权重参数，最后拟合网络模型。本文 CNN 构成如下：

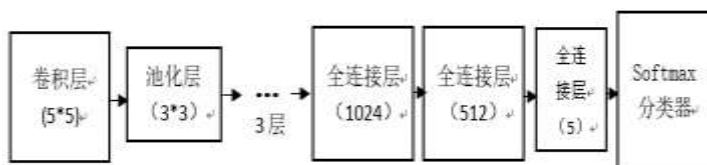


图 4.3 CNN 结构



图 4.4 CNN 可视化界面图

训练 CNN 时，得先对图片进行处理，通过 `ReadImg()` 读入图片转为图片数据，再把处理后的图片数据放入 CNN 训练。图片数据经过网络模型不断进行迭代计算，获得拟合好的模型。为了测试模型的拟合程度，选取部分没训练的图片对训练的模型进行预测，以评估模型的好坏。训练好的模型要经过冻结图删去多余的节点，保存核心节点为 Pb 模型文件再转化为 Bytes 模型文件。另外，相对于命令行输入命令切换路径、运行 Py 文件等繁琐操作，本文通过 Python 的 wxPython 库以可视化界面的形式集成了 CNN 训练和测试两个模块。可视化界面采用简洁的操作就能完成 CNN 训练过程，对扩展其他原子模型的训练提供了便利，如图 4.4。

#### 4.1.3 化学分子结构模拟设计

Unity 的脚本如何调用 CNN 模型进行结构识别是化学分子结构模拟应解决的核心问题。这涉及到编程语言底层不兼容的问题，但在 C# 中引入 TensorSharp 库能很好地解决这个问题，因为 TensorSharp 库提供了 C# 实现的 Tensorflow API 接口。由此可知，化学分子结构模拟应解决的是引入 TensorSharp 库后如何在 Unity 中构建卷积神经网络进行分子结构识别的问题。

为了快速完成对最相似分子结构的识别从而进行评分，把预先构建的数据集经过 CNN 的训练保存成模型，完成对分子结构的编码，并转换成 Bytes 模型。通过原子摄像机捕获用户观察到的模型图像，并对其进行投票识别，从而完成对玩家模型分数的评定。

该模块的构成主要由 Classification 类、CameraBehaviour 类、TextureTool 类以及 MessageBehaviour 类组成。由图 4.4 可知 CameraBehaviour 类内置了拍摄图片的方法，该方法创建 RenderTexture 类，赋值给分子组装摄像机的 TargetTexture 属性就可以观察到剪裁空间的原子组装模型。然后利用 Unity3D 的 Texture2D 类对摄像机所投影的空间进行截图，对拍摄的照片使用 Texture Tool 工具类进行裁剪。需要注意的是，这里把分子摄像机的渲染图层调为只对原子模型可见，确保与训练图片的背景一致，具体代码如下：

```
Color32[] Get Image ()
{
    得到分子摄像机的 Render Texture;
    //根据分子摄像机的 TargetTexture 创建图片
    Texture2D cameraImage = new Texture2D(参数);
    cameraImage.ReadPixels(图片的大小, 0, 0);
    cameraImage.Apply();
    图片裁剪成 100*100;
    return 裁剪的图片数据;
}
```

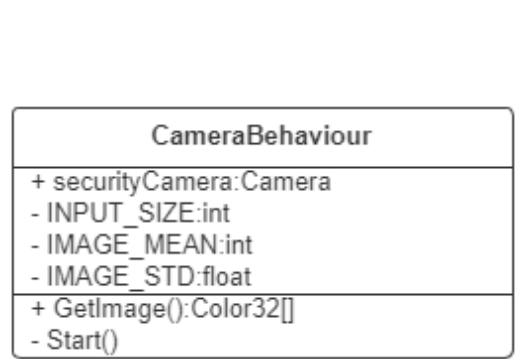


图 4.5 CameraBehaviour 类图

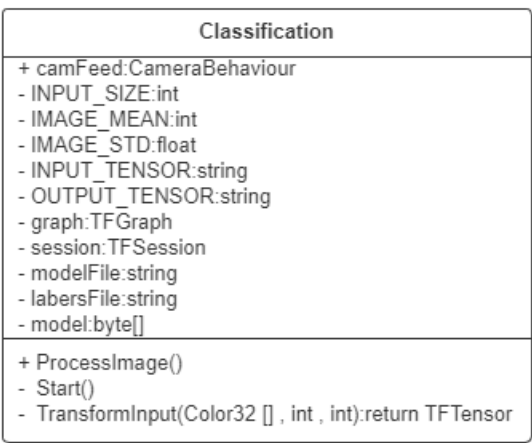


图 4.6 Classification 类图

通过 `GetImage` 函数获取到的多角度图片数据作为参数传入 `Classification` 类的 `TransformInput` 函数中，并把 `Color32[]` 类型的图片数据转换成 `Float[]` 类型的图片数据，再拷贝到 `TensorSharp` 所提供的 `TFTensor` 的类型。

在 `Classification` 类图中，`INPUT_SIZE` 是图片的大小，`IMAGE_MEAN` 是图片的均值，`IMAGE_STD` 是输入数据的维数，`INPUT_TENSOR` 是 CNN 输入节点的名称，`OUTPUT_TENSOR` 是 CNN 输出节点的名称，`graph` 是训练 CNN 所保存的模型，`model` 是加载 Bytes 文件所保存的数据。`modelFile` 和 `labelsFile` 是加载 Bytes 文件和 Txt 文件的路径。

玩家通过触碰交互方块回调 `ProcessImage()` 函数进行识别。在 `ProcessImage()` 函数中构建内部识别图，调用 `TensorSharp` 所提供的神经网络构建接口开启 session，把图像数据赋值给输入节点，进行图像识别，然后再对组装模型进行分数评定，核心代码为：

```
public void ProcessImage() {  
    var labels = File.ReadAllLines(labelsFile);  
    var tensor = TransformInput(图片数据, INPUT_SIZE, INPUT_SIZE);  
    var runner = session.GetRunner();  
    runner.AddInput(graph[INPUT_TENSOR][0], tensor).Fetch(graph[OUTPUT_TENSOR][0]);  
    var output = runner.Run();  
    //根据输出结果计算图片的分数  
    ...  
}
```

Unity3D 化学分子结构模拟设计还有另一个模块：分子结构组装模块。如何检测原子模型碰撞以及全方位固定角度是分子组装模块应解决的问题。Unity3D 的 `FixedJoint` 组件能像一根杆子那样连接固定两个物体，所以分子组装模块是基于 `FixedJoint` 组件实现物体与物体之间的链接。每一个棍棒模型都会挂载着 `ConnectObj` 组件，当物体靠近棍棒模型表面时，如果物体的标签是原子所属的标签，则回调碰撞检测函数，进行原子模型与棍棒模型之间的连接，然后设置原子模型的父节点。如果连接的模型超过 2 个，则不再进行连接。

核心代码如下：

```
public void addFixedJoint(GameObject connectedObj )
{
    if (不是连接过的原子模型&& 连接链表的长度 <= 2)
    {
        connectobj.Add(connectedObj);
        jointComponent = gameObject.AddComponent<FixedJoint>();
        connectedObj.gameObject.transform.SetParent(this.transform);
        FixedJoint hjoint = (FixedJoint)jointComponent;
        ...
        hjoint.connectedBody =connectedObj.GetComponent<Rigidbody>();
    }
}
```

当分子结构拼装完成时，可以碰触图像识别方块，调用图像识别模块进行识别，再根据剩余时间和 CNN 识别模型的分数进行评分，如下图：

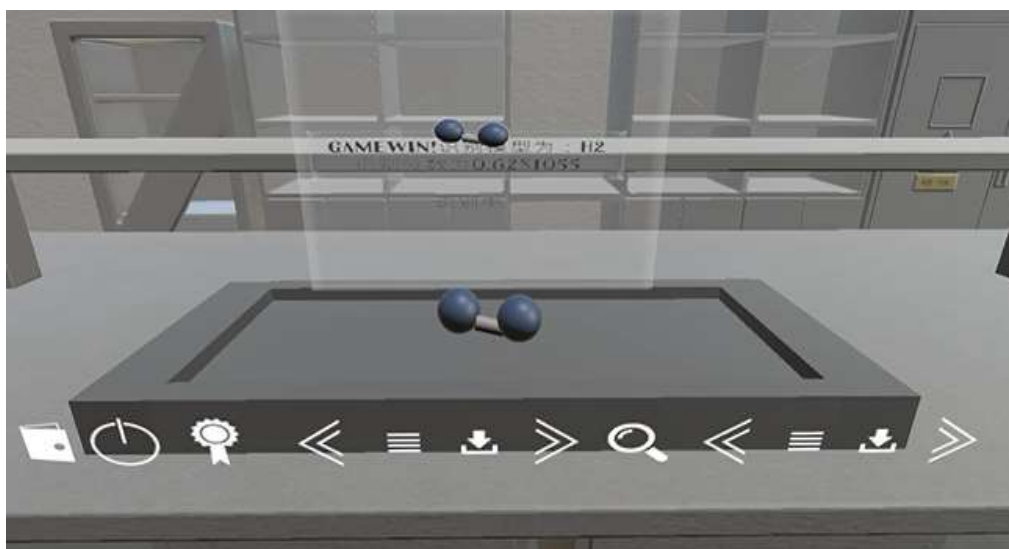


图 4.7 分子结构模拟识别图

## 4.2 液体仿真设计

### 4.2.1 液体 Shader 设计

参考现实中瓶内液体，把液体的仿真效果分成液体的流动速度、液体的液体反射效果、液体的颜色和液体的倾倒效果这四部分。对于四个效果，纯 C#实现的难度不小，但 Unity3D 的 Shader 提供了顶点和片元着色器的编程，让液体仿真的实现减少了不少难度。

液体的倾倒效果主要是在片元着色器下完成的，首先在 C#脚本中执行移动后液体 X 轴正弦波幅度\_MaxHeightInX 和 Z 轴正弦波幅度\_MaxHeightInZ 的计算，然后传入 Shader 中生成顶点着色器各顶点的裁剪高度，如果片元着色器的顶点大于相对应的高度，则将大于的部分舍弃掉。

液体的反射效果是利用 Shader 编程内置的 `reflect()` 函数并传入入射光和顶点法线完成的，但入射光和顶点法线向量是两个坐标空间下的向量，所以需要世界空间转换矩阵来完成向量的转换。液体的流动主要对贴图法线纹理进行采样，再通过更改片元着色器顶点相对应的 UV 坐标来达到流动效果，液体 Shader 的属性设置如表 4-1。

表 4-1 FragClip 属性表

Shader 属性	属性说明
<code>_MainTex</code>	液体纹理
<code>_BumpMap</code>	法线纹理
<code>_WaveSpeed</code>	流动速度
<code>_AdditionColor</code>	液体面偏移比例
<code>_MaxHeightInX</code>	X 轴正弦波系数
<code>_MaxHeightInZ</code>	Y 轴正弦波系数
<code>_AlphaScale</code>	Alpha 值

Shader 计算倾倒的核心代码计算：

```
float getClipYAveValue(float4 pos)
{
    float vexheight;
```



```

    vexheight =(_MaxHeightInX * pos.x + _MaxHeightInZ * pos.z) / 2;
    return vexheight;
}

```

而根据 `getClipYAveValue()` 返回的高度还不是裁剪高度，只是液体液体面晃动的高度，所以，要计算裁剪高度，还必须加上液体本身的高度，如下代码：

```

v2f vert(a2v v)
{
    v2f o;
    o.pos = UnityObjectToClipPos(v.vertex);
    ...
    o.clipY = getClipYAveValue(v.vertex) + _FluidHeight;
    ...
    return o;
}

```

#### 4.2.2 液体控制效果设计

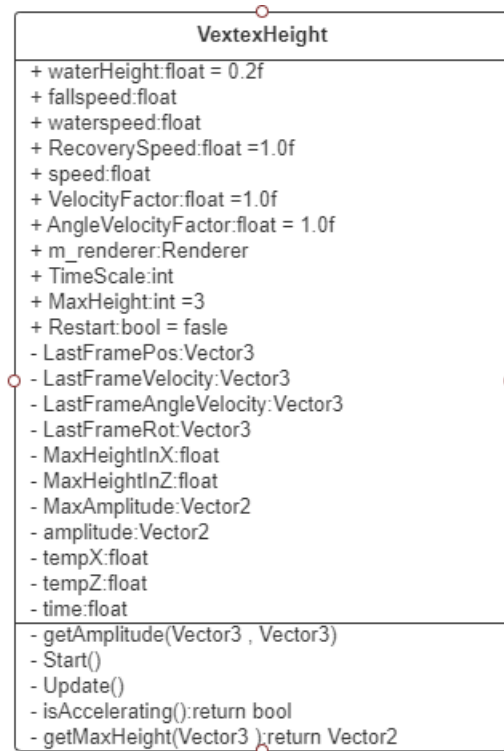


图 4.8 VextexHeight 类图

液体的效果是在 FragClip 上实现的，但液体的控制效果是 VextexHeight 实现的。对于液体抖动的幅度以及倾倒高度的下降，主要通过检测上一帧和这一帧的位置和旋转角度是否移动来计算的。根据位移的速度或者旋转的角速度决定了初始振幅的大小，再判定物体是否处于加速状态。如果是就通过 getAmplirude 方法传入当前运动速度和角速度更新最大振动幅度，并且让液体面始终保持与速度相关的一个倾斜状态。液体的晃动幅度随着时间的增加而减少，晃动幅度的减少主要通过计算正弦波幅度\_MaxHeightInX 和\_MaxHeightInZ 到 0 的插值完成的。如果液体处于静止状态时，则进行正弦波运动的计算，得到当前的振动幅度，如果不是则把振动幅度赋值给正弦波幅度系数。核心代码如下：

```
if(液体单纯移动)
{
    计算液体的速度和角速度;
    如果液体处于加速时期，则对液体晃动的振幅增加;
    amplitude.x=Mathf.Lerp(MaxAmplitude.x, 0, time * RecoverySpeed);
    amplitude.y=Mathf.Lerp(MaxAmplitude.y, 0, time * RecoverySpeed);
    if(液体速度和角速度为零)
    {
        进行正弦波运动计算，并得到 MaxHeightInX 和 MaxHeightInZ 的值;
    }
    else{
        正弦波幅度系数 = 振动幅度;}
    ...
}
```

当液体进行倾倒时，则不对其进行晃动，通过绑定液体的 X 轴或者 Z 轴的旋转角度来增加或减少正弦波幅度 `_MaxHeightInX` 和 `_MaxHeightInZ` 来达到液体向倾倒方向倾倒的一个过程，如下图：

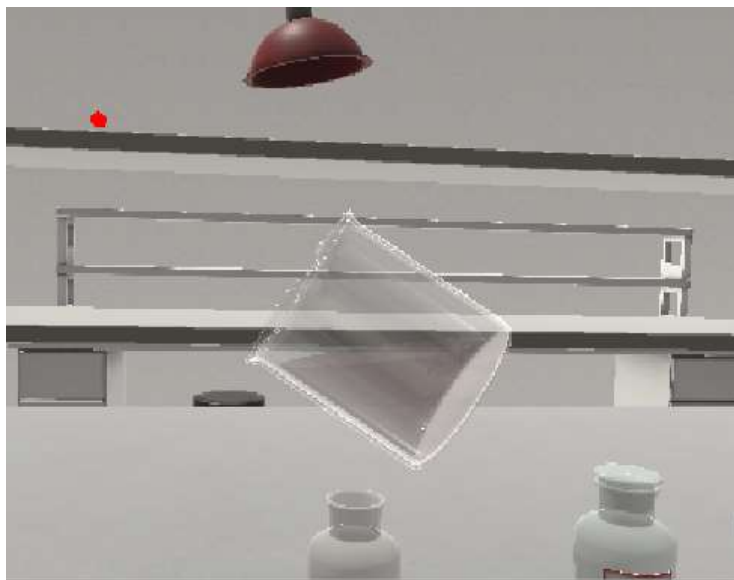


图 4.9 液体倾倒图

在本文中，液体倾倒幅度的计算是通过 `getMaxHeight` 方法传入液体的角度计算的。首先，给 `_MaxHeightInX` 设置 `[-3, 3]` 的区间值来调整液体液体面 X 轴正负方向的晃动，`_MaxHeightInZ` 也是同样的道理。`_MaxHeightInX` 的区间值对应的是液体 X 轴旋转角度的区间值 `[-90, 90]`，`_MaxHeightInZ` 的区间值对应的是液体 Z 轴旋转角度的区间值 `[-90, 90]`，增加或减少其旋转角度就是增加或减少正弦波幅度 `_MaxHeightInX` 和 `_MaxHeightInZ`。而当液体倾倒到 90 度，液体面则倾斜到最大高度，同时液体的高度则会下降。

由于 Unity3d 的旋转区间是以负无穷大到正无穷大表示的，所以对液体的旋转角度和正弦波幅度进行转化计算时，需要对液体的旋转角度进行取余计算，再对得到的角度和正弦波幅度进行转化计算。

## 第5章 游戏测试

### 5.1 测试目标

一款游戏软件在开发的过程中，并不能保证其稳定性和毫无漏洞，为了保证开发的质量和可靠性，应在开发各阶段遵循代码的规范，以及代码的编写原则，但由于个人能力的局限性，在开发过程中，总会不经意下编写出一些错误，这些错误如果没有经过严格的测试，在上线过程会容易产生各种问题从而导致用户的流失。所以在游戏开发阶段，应根据游戏开发文档定下功能测试的标准，写好测试用例，解决潜在的问题，才能保证游戏开发的顺利进行。

在本章节中，主要是对本文所阐述的游戏功能点进行测试：

- 1) 测试原子拼接功能是否正常
- 2) 测试 XML 的文件的读写是否正常
- 3) 测试排行榜的更新是否正常
- 4) 测试关卡系统的选择是否正确
- 5) 测试液体是否能够倾倒
- 6) 测试图像是否能分类

### 5.2 测试过程

本次测试所搭载的平台是 Windows 系统的 PC 端，环境配置是 Intel CoreI7，16G 运存，以下各项表格是本次测试内容及结果。

原子拼接功能的测试，如表 5-1 所示，主要测试的是 ConnectObj 类，测试的主要目标是借助 VR 设备，操纵原子模型和棍棒模型检测模型接触后是否固定在一起，一个棍棒模型是否只能链接两个原子模型，测试结果如图 5.1。

表 5-1 原子拼接功能的测试表

编号	测试项	测试用例	预期结果	实际结果
A01	原子模型和棍棒模型是否能拼装	连接原子模型和棍棒模型	原子模型和棍棒模型能拼装	原子模型和棍棒模型能拼装
A02	棍棒模型是否只能链接两个原子模型	连接多个原子模型和棍棒模型	一个棍棒模型只能链接两个原子模型	一个棍棒模型只能链接两个原子模型

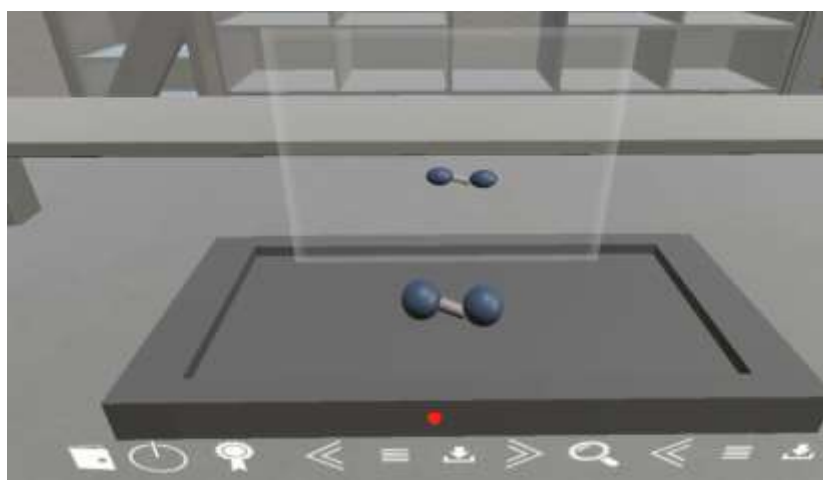


图 5.1 原子模型和棍棒模型拼装测试结果

XML 的文件的读写的测试，如表 5-2 所示，主要测试的类是 LevelSystem 类和 XMLManager 类，测试的主要目标是当进行关卡数据更新和排行榜数据更新时，LevelSystem 类的 LoadLevel ()、SetLevels () 和 XMLManager 类的 UpdateXml () 三个 API 能够正常被调用以及 XML 文件的读写是否正常。

表 5-2 XML 的文件的读写的测试表

编号	测试函数	测试用例	预期结果	实际结果
B01	LoadLevel ( )	点击关卡选择方块并 Debug 输出文件信息	关卡正常加载，关卡的解锁程度正常显示	关卡正常加载，关卡的解锁程度正常显示
B02	SetLevels	文件路径 B	配置文件的内容被重写	配置文件的内容被重写
B03	UpdateXml ( )	文件路径 C	排行榜配置文件内容被重写	排行榜配置文件内容被重写

排行榜更新的测试，如表 5-3 所示，主要测试的是排行榜的模型 UI 和分数是否显示正常，以及当玩家分数超过排行榜显示的分数，排行榜是否会更新分数。测试结果如图 5.2 所示。

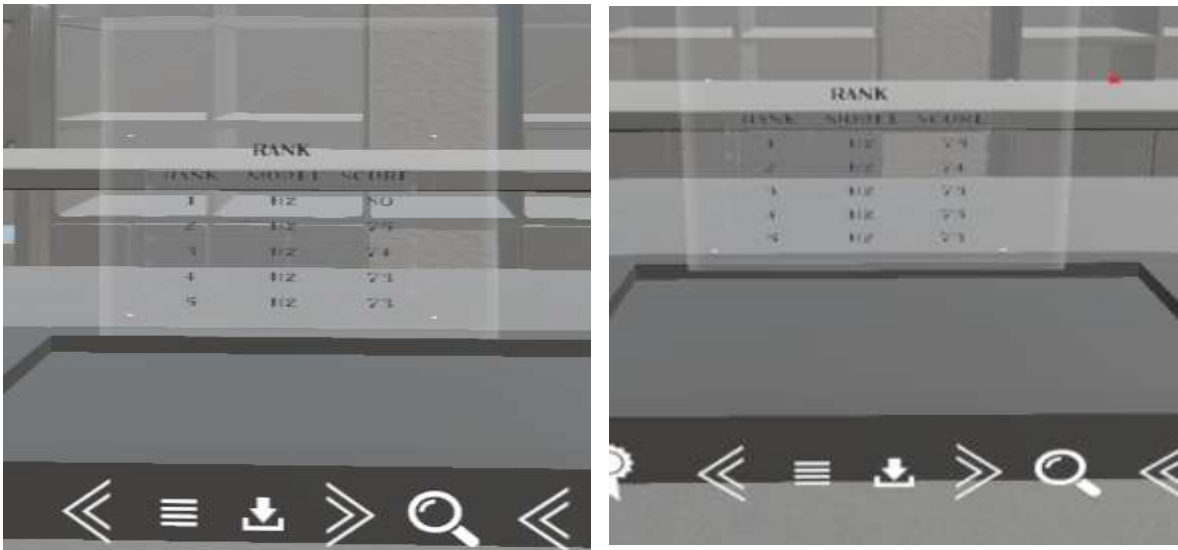


图 5.2 排行榜更新的测试图

表 5-3 排行榜更新的测试表

编号	测试项	测试用例	预期结果	实际结果
C01	模型 UI 和分数的显示	碰触排行榜方块	模型 UI 和分数显示正常	模型 UI 和分数显示正常
C02	排行榜分数的更新	设置固定识别分数	排行榜更新分数	排行榜更新分数

关卡系统选择的测试，如表 5-4 所示，主要测试关卡显示是否正常、关卡选择是否正常，当解锁关卡的条件达到时是否能正常的解锁，测试结果如图 5.3。

表 5-4 关卡系统的选择的测试表

编号	测试项	测试用例	预期结果	实际结果
D01	关卡的显示	点击关卡方块	关卡正常显示	关卡正常显示
D02	关卡的选择	点击关卡选择 方块	关卡正常移动	关卡正常移动
D03	关卡的解锁	固定识别分数	关卡正常解锁	关卡正常解锁

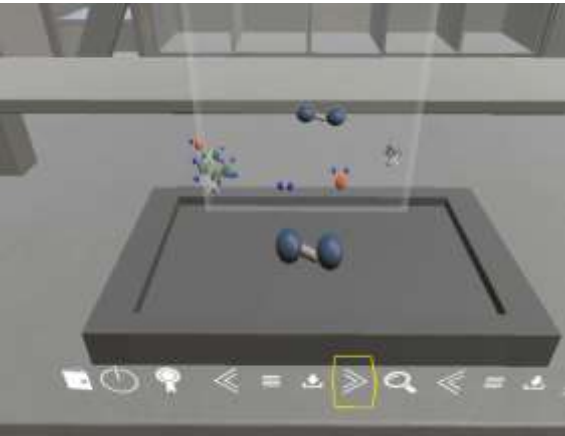


图 5.3 关卡系统选择的测试测试结果



图 5.4 通关示意图

化学分子结构识别测试，如表 5-5 所示，当玩家拼装好分子结构模型时，点击识别，对所拍摄的图片进行结构检测。如果玩家组装失败和玩家超过规定通关时间，则游戏结束，如图 5.5 所示；如果组装成功，识别分数和剩余分数合计分数 0.7360，超过 60 分，玩家通关，如图 5.4 所示。

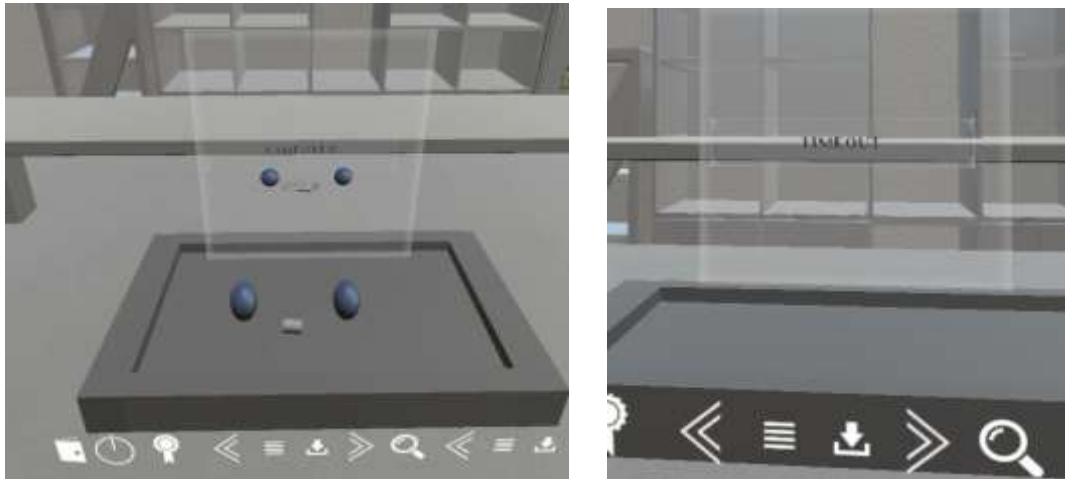


图 5.5 未通关示意图

表 5-5 图片分类结果的测试表

编号	测试项	测试用例	预期结果	实际结果
F01	正常组装分子结构，点击识别，分子结构能否正常检测	点击识别	组装好模型，显示分子结构检测结果和分数	组装好模型，显示分子结构检测结果和分数
F02	不组装分子结构，点击识别，分子结构能否正常检测	点击识别	没组装模型则显示 GameOver	没组装模型则显示 GameOver
F03	玩家超过时间点击识别，分子结构能够正常检测	超过时间，点击识别	超过时间则显示 GameOver	超过时间则显示 GameOver



液体倾倒效果的测试，如表 5-6 所示，主要测试的是借助 VR 手柄抓取容器左右旋转时，液体能否晃动以及倾斜，如图 5.6。

表 5-6 液体倾倒效果的测试表

编号	测试项	测试用例	预期结果	实际结果
E01	液体倾倒	容器左右旋转	液体正常倾斜	液体正常倾斜，液体倾倒效果



5.6 液体倾倒效果测试图

## 参考文献

- [1] 喻偃, 莫瑜. 深度学习原理与 TensorFlow 实践[M]. 北京: 电子工业出版社, 2017. 71-121
- [2] 曼普里特·辛格·古特. TensorFlow 神经网络编程[M]. 北京: 机械工业出版社, 2018. 65-102
- [3] 胡良云. HTC Vive VR 游戏开发实战[M]. 北京: 清华大学出版社, 2017. 230-280
- [4] 刘刚, 孙文涛. unity 官方案例精讲[M]. 北京: 中国铁道出版社, 2015. 40-50
- [5] 谭小辉, 万旺根, 黄炳, 崔滨. 基于 SPH 的三维流体模拟[J]. 计算机应用与软件, 2009, 26(12): 222-224+258.
- [6] 欧阳劲夫, 龚捷. 基于 Unity3D 的液体面波浪模拟[J]. 电脑知识与技术, 2020, 16(05): 219-220.
- [7] 雷国伟. 图像特征的 CNN 提取方法及其应用[J]. 计算机工程与应用, 2004, (14), pp. 204-206+216CNKI
- [8] 洪睿, 康晓东, 郭军, 李博, 王亚鸽, 张秀芳. 基于复杂网络描述的图像深度卷积分类方法[J]. 计算机应用, 2018, 38(12): 3399-3402
- [9] 张家欢. 基于 GPU 元球造型的液体滴模拟[D]. 武汉科技大学, 2011.

## 致谢

本论文及设计在我的指导老师秦胜伟的悉心指导下完成的，回顾制作毕业设计和撰写毕业论文这几个月中，老师给予了我悉心的指导，对毕业设计和论文不足的地方给予了宝贵的意见，让我的毕业设计和论文得以完善，在此对秦老师的辛勤付出表示衷心的感谢！

另外，也感谢学校的所有任课老师的教育与栽培，使我具备了专业知识储备与素养，和萧懿仁同学对我论文提出的建议和帮助。

感谢我的大学期间所认识的所有同学，感谢他们在生活以及学习上给予我的帮助。谢谢！