|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **华南农业大学数学与信息学院**  程序设计与算法综合实习Ⅱ   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **班级** | | 24软工5班 | **学号** | 202425220527 | **姓名** | 徐阳 | | **实**  **验**  **题**  **目** | **基于Ren’Py的视觉小说游戏实现**  ■设计性 ■综合性 | | | | | | | **自**  **我**  **评**  **价** | 本人在项目中负责游戏设计和代码编写、游戏素材收集和总体质量把控，顺便协调组员的开发学习，做了大部分的事情。 | | | | | | | **教**  **师**  **评**  **语** | 能够实现实验要求的功能 □全部 □部分  算法有新意 □有 □一般  程序运行通过 □全部 □部分  算法注释说明 □完善 □仅有功能说明  接口参数说明 □有 □无  按期上交文档资料及源程序 □所有 □部分  独立完成实验 □能 □不能  体现团队合作精神。 □能够 □不能 | | | | | | | **成绩** |  | | | | | | |

摘 要

本项目基于 Ren’Py 引擎开发了一款视觉小说游戏，旨在通过文字剧情、角色立绘与交互式选项为玩家提供沉浸式体验。与传统视觉小说不同，本项目特别强调“数据结构与算法在游戏编程中的应用”，在剧情进程中嵌入了一个 Pong 小游戏模块。该模块使用 Python 语言实现，通过维护位置变量与状态变量，完成了玩家球拍控制、球体运动与 AI 对手行为的交互逻辑，体现了数据结构在实际游戏开发中的价值。

在项目实施过程中，首先进行了需求分析与工具选择，明确了使用 Ren’Py 作为开发平台的优势与局限；随后设计了整体架构，包括剧情脚本、界面元素和小游戏对象类的封装；最终实现了视觉小说与小游戏的整合，并在不同设备上进行了多维度测试。结果表明，本项目能够稳定运行，视觉小说部分叙事流畅，小游戏部分交互自然，达到了预期目标。

本研究不仅锻炼了小组成员的编程实践能力，也加深了对数据结构应用场景的理解。该项目展示了视觉小说与互动小游戏结合的可行性，证明了在有限开发周期与成本下，学生团队亦能完成较为完整的跨学科项目实践。

关键词：视觉小说；Ren’Py；小游戏；数据结构

目 录

**未找到目录项。**

1. 绪论

1.1 课题的研究目的及意义

* + 1. 课题研究目的

本课题研究的目的在于小组四人合作、动用身边一切可以用到的资源，开发出一款视觉小说游戏，可以在电脑和手机中运行，带给人乐趣、享受和对真善美的体验。

* + 1. 课题研究意义

本课题研究的意义在于，软件工程大一新生应当勇敢闯荡各行各业，在各种领域取得自己的体验，尤其是在做一个项目时。能完整的做完一个项目就已经超越了不少人，在此之中，人不得不培养自己独立解决问题的能力，过关斩将地走下去，这种项目开发经验在将来的人生路上非常有用。因此，做一个有头有尾、同时小组成员投入的游戏项目意义非凡。

[1.2 国内外研究现状](#_Toc1124)

[1.2.1 国外研究现状](#_Toc7362)

视觉小说（Visual Novel）起源于日本ACG文化，最早作为介于小说与游戏之间的互动媒介出现。其特点在于以文字剧情为核心，辅以立绘、CG和音乐渲染氛围，玩家通过阅读文本和有限选项来推动故事进展。随着上世纪九十年代日本PC平台的发展，视觉小说逐渐形成了完整的产业链：从企划、剧本、美术到发行都有成熟分工，作品题材也从单纯的恋爱物语扩展到科幻、推理、悬疑等多个方向。

在欧美地区，视觉小说虽然起步较晚，但依托Steam等数字发行平台，以及Unity、Ren’Py等通用引擎的普及，逐渐建立了小规模创作社群。与日本产业化模式不同，欧美视觉小说更偏向独立开发，多数由个人或小团队完成，常见形式为英文版Steam游戏或网页互动小说。总体而言，日本在产量与成熟度上依然处于引领地位，而欧美则强调创意表达与题材多元化。

[1.2.2 国内研究现状](#_Toc23372)

在国内，视觉小说的发展路径有所不同。一方面，腾讯、米哈游等大型厂商推动了乙女向与剧情向手游的发展，虽然在商业模式上引入了抽卡、养成等系统，但依然延续了视觉小说“以剧情为核心”的特点；这类产品因强社交属性与高投入美术而获得了较高人气。另一方面，独立团队与个人开发者也尝试使用Ren’Py、Unity等工具制作小型视觉小说，并通过爱发电、Steam等平台发行。但受限于市场体量、政策审查和资金支持，国产视觉小说整体数量有限，且高完成度作品较少。

总体来看，国外视觉小说产业化程度高、题材成熟，而国内市场处于探索与成长阶段，商业化产品与独立创作并存。本项目正是借鉴了国外视觉小说“叙事+交互”的模式，同时结合课程要求，将数据结构应用到其中，体现了在教学与实践中探索国产视觉小说开发路径的一种尝试。

* 1. [本论文主要内容与结构安排](#_Toc29100)

本论文主要从小组中我个人涉及的制作角度出发编写，描述游戏制作的步骤并解释。由于代码编写的角色主要由我本人担任，本论文将会尽我所能简洁围绕其代码功能实现，按照一定的结构讲述。

[第二章 功能分析与工具选择](#_Toc23654)

2.1 需求分析

首先，我期望这个游戏能有一般定价一美元的小游戏水准，在此前提下，应当借助成熟的开发工具，寻找或写作合适的脚本剧情，使用AI生成悦目的美术资料，并使用PS工具进行立绘和差分制作，同时在游戏中嵌入小游戏或其他具有可论说的数据结构内容，以便完成作为数据结构课设的基本要求。但由于多方问题，游戏无法配音，且因制作周期和成本限制无法在美术上做到完全原创，因此游戏实际上可能难以上架，仅作为课设和学习而做。

[2.2](#_Toc14243) 开发工具选择

在刚开始策划这个问题的时候，我们小组采取了投票的方式决定开发工具和题目的选择，这是因为题目和引擎是紧密相关的，工具引擎的局限性也是制作出的游戏的局限性。

因此我们小组成立了微信群便于沟通，我在群中发布了这样一条信息，简述可以实现的选题以及该选题应当使用的开发工具：

1.视觉小说，用剧情和对白撑起游戏进程，使用少量选项分支剧情，在物品清单或关卡进程中使用数据结构；

优点是特别简单，使用接近Python的语言进行开发，使用Ren'Py引擎

2. RPG游戏，角色可以上下左右移动，有地图可交互，使用rpg maker进行开发，需要一定时间学习成本；

优点是不用敲代码，可以纯用图形开发界面实现数据结构，且游戏性较高

3.独立c++游戏，调用一些专门用于游戏开发的库，自己开发图形界面，设计游戏；

优点是自由度很高，无论做什么都可以，数据结构也非常好嵌入，使用codeblocks或者vscode就能开发；缺点是没有现成的美工图库，需要自己找素材或作画。很多好玩游戏的学长的毕业设计作品也是做的这个。

以上都属于说是正统的游戏开发，几乎可以上架的那种类型（用到引擎的需要正版版权）；以下的选题纯属作为一门课来对待。

4.飞机大战、大战外星人、坦克大战等写在编程实践入门教程里的经典项目。优点是参考书随便做。缺点是没有创意。

5.使用一切力量复现现在市面上有的一些成熟的游戏，比如做个小王者，做个小奇迹暖暖，做一个小抽卡游戏。优点是可以很好了解现在的游戏现状，熟悉大众游戏的逻辑；缺点是纯属抄袭且实现难度很高。

因此，我们在表决后，决定使用Ren'Py引擎制作视觉小说。

[2.3 游戏可行性分析](#_Toc14243)

[2.3.1 技术可行性](#_Toc534)

由于Ren'Py从入门到发行对新手非常友好，在“能否实现”问题上不会有太大的门槛；关键是它基于Python语言，而我们是否可以根据这个特性拓展出很多复杂的实现。即便复杂的东西最后实现不了，也能做出最基础的、能安装到手机.apk文件电脑的.exe文件，因此在技术上完全可行。

[2.3.2 经济可行性](#_Toc16858)

由于美术方面借助生成、取消配音，软件免费获取，排除人员耗费的时间精力，经济成本为0。

[2.4 系统非功能性需求](#_Toc27201)

[2.4.1](#_Toc534) 界面与图片设计

Ren'Py引擎制作的视觉小说游戏自带最基础的UI设计，且UI设计并非我们项目的重点。图片主要由另一位小组成员制作。

2.5 开发能力学习

[2.5.1](#_Toc534) Ren'Py引擎学习

Ren'Py有详细的文档可以线上阅读，且自带教程丰富；尽管语言冷门，但在B站这类视频平台上仍有不少视频教学可以观看学习。由于本人曾独立完成过一个最简单的Ren'Py项目，因此前期组员学习主要由我来引导，为其自学开一个好头。

[2.5.2](#_Toc534) Git使用学习

在多人开发中，团队代码管理举足轻重，应当学会git的使用。由小组中有经验的成员初步带领组员注册GitHub代码托管平台并加入项目，给予git的学习资源并确认组员是否掌握提交代码并发布拉取请求的能力，建立起团队代码开发的便捷环境。

[第三章 游戏设计](#_Toc4374)

[3.1 技术路线与软件架构](#_Toc25169)

[3.1.1](#_Toc18002) 开发架构

一个由Ren'Py开发的视觉小说游戏由代码和素材两部分构成。

代码分为四个部分，一是script，即脚本，是整个游戏制作的核心，所有内容以及内嵌的python语句都归属于脚本之中。二是options，即配置，记录游戏配置、版本号等功能性参数。三是gui，即界面，负责游戏的界面绘制，并非我们开发的重点。四是screens，即屏幕，用于搭建整个游戏的框架，是难以撼动的地基。

素材分为六个概念，一是base，指整个项目仓库。二是game，包含base中除了git相关说明、游戏日志等外关于游戏本身的东西。三是images，储存游戏中的图片文件，新项目默认为空。四是audio，储存游戏中的全部音频，包括配乐、配音、音效。五是gui，储存诸如消息提示框、主界面UI背景、游戏图标等与核心内容关系较远的界面性图片。最后一个是lib，用来囊括附加的库文件，比如live2D拓展等。

以上为开发架构，我们后续的工作就是打开script.rpy编辑代码（由于体量不大，无需分文件），以及向images和audio文件夹中加素材。

3.2.2 内容架构

内容架构较为难以详述，姑且认为由三个部分构成，一是演出，意为小说阅读过程中配合文本进行的视觉化效果展示，如场景CG的展示与过场动画、人物立绘的切换移动、特效等。二是选项逻辑，或者说线路，意为视觉小说由于可存在的选项选择不同而有不一样的剧本组成方式，在代码中主要由label代码块和类似于C语言中goto的jump（Ren'Py语言自定义）来实现小说阅读顺序的处理和满足玩家的选择意愿。三是小游戏，是本次数据结构实现上的亮点，在Ren'Py脚本中可以随时随地通过init python:语句来插入一段原生的python代码，例如定义一个class对象。由此可以实现内嵌小游戏的制作，并让小游戏在游玩过程中达到一定场景后触发。

[3.2 用例实现](#_Toc20402)

[3.3.1某某某功能](#_Toc18002)

这里搞几个有代表性的renpy语句及其效果上去

[3.3 内嵌小游戏设计](#_Toc4138)

为增强视觉小说的互动性与课程实验的创新性，本项目在Ren’Py脚本中内嵌了一个经典Pong小游戏。该设计既能展示Python与Ren’Py的良好结合，也能体现数据结构在游戏交互中的具体应用。见script.rpy line 928。

[3.3.1 对象设计](#_Toc10494)

小游戏采用面向对象的方式进行封装。核心对象为PongDisplayable，继承自renpy.Displayable类。通过该类，定义并维护游戏的主要元素：

1) 球拍对象：左右各一，分别由玩家与电脑控制。

2) 球体对象：在场地内不断运动，并与球拍、上下边界发生碰撞。

3) 场地边界：确定运动范围，用于检测球体是否出界。

此外，还维护若干状态变量，包括球拍坐标、球体速度与方向、电脑速度等。

[3.3.2 数据结构设计](#_Toc18555)

小游戏中涉及的关键数据结构设计如下：

1. 位置变量：

playery：玩家球拍的纵坐标，随用户输入实时更新。

computery：电脑球拍的纵坐标，基于算法逐步靠近球体位置。

ballx, bally：球体坐标，每一帧根据速度增量进行更新。

1. 状态变量：

stuck：布尔值，表示球是否处于发球等待状态。

ballvx, ballvy：球体在x、y方向上的速度，用于控制运动方向与速度大小。

1. 图形对象引用：

paddle与ball分别由Solid图元生成，充当球拍与球体的图像。

这些数据结构共同维护了小游戏的运行状态，并驱动画面刷新与逻辑判断。

[3.3.3 类的描述](#_Toc17802)

class PongDisplayable(renpy.Displayable)

1. 初始化方法\_\_init\_\_

定义游戏参数与图形元素，包括球拍、球体大小、场地上下边界等；同时初始化状态变量。

1. 渲染方法render
2. 每一帧调用一次，用于绘制游戏元素并更新逻辑：
3. 根据键盘输入更新playery。
4. 根据球体坐标与边界条件更新ballx, bally。
5. 检测球与球拍的碰撞，若发生则反转水平速度。
6. 调整computery使其逐渐靠近球体纵坐标，实现简单的电脑控制。
7. 将球拍与球体绘制到画布上并返回。
8. 事件处理方法even

接收用户输入（上下方向键），并据此调整playery。

通过上述类的定义，游戏逻辑与画面渲染实现了高度耦合，确保在Ren’Py引擎中能以稳定帧率运行。

[3.3.4](#_Toc17802) 设计意义

小游戏的设计不仅是单纯的娱乐模块，更承担了展示“数据结构与算法在游戏中的应用”的课程任务。其实现体现了以下几点：

1. 状态维护：通过多个变量构建起完整的游戏状态机，展示了数据结构在程序运行中的核心作用。
2. 算法应用：电脑球拍采用“逐渐逼近球体纵坐标”的简易跟随算法，虽原理简单，却体现了基于状态变量的实时决策过程。
3. 扩展性：类的封装方式使得后续可轻易添加得分系统、难度调整或音效反馈，为未来改进预留了空间。

综上，本项目的Pong小游戏设计在技术上可行，在展示效果上具备互动性和趣味性，成功支撑了实验课程“数据结构在游戏编程中的应用”这一目标。

[第四章 游戏实现](#_Toc5238)

[4.1 整合](#_Toc23942)

[4.2 导出](#_Toc7821)

[第五章 游戏测试](#_Toc3528)

[（1）基础功能测试](#_Toc21588)

[（2）内容游玩测试](#_Toc616)

[（3）内嵌小游戏测试](#_Toc30639)

本项目在视觉小说的中后期嵌入了一个经典的Pong小游戏，用于展示Ren’Py引擎中init python:脚本与自定义Displayable类的结合能力，也是本次课程设计的数据结构与算法应用重点。为了确保该模块在游戏整体框架下的可用性与稳定性，本节对小游戏进行了较为系统的测试。

**测试目的**

1. 验证小游戏能否在视觉小说剧情进程中被正常调用、启动和退出。
2. 检查玩家操作与游戏反馈之间的响应速度是否满足基本可玩性。
3. 确认游戏状态的变量维护、碰撞判定逻辑、电脑控制逻辑等是否符合预期，实现预设的交互效果。
4. 评估小游戏与主游戏内容在体验上的衔接是否自然。

**测试方法**

* **功能测试**：进入触发小游戏的场景，操作角色球拍进行游戏，观察是否能实现上下移动、球体反弹与得分判定。
* **逻辑测试**：通过分析代码中playery、computery、stuck等状态变量的更新情况，验证数据结构的维护是否准确；重点检查球体碰撞与越界检测的正确性。
* **性能测试**：分别在PC（Windows）与Android手机端运行测试，记录帧率与操作延迟。
* **用户体验测试**：邀请 3 名非小组成员进行试玩，记录其对于难度、流畅度及趣味性的主观评价。

**测试过程与结果**

1. **功能方面**：小游戏能在剧情中顺利进入，按下键盘方向键或触摸屏控制球拍上下移动时，反馈及时，球体能够正确实现上下左右的边界反弹和与球拍的碰撞检测。分数在球越界时正确重置，重新开始机制正常。
2. **逻辑方面**：通过在代码中插入调试输出语句验证，playery随用户输入更新正常，computery的 AI 控制逻辑表现为“以固定速度逐渐逼近球的纵坐标”，符合设计思路。边界条件处理合理，未出现数组越界或空引用等错误。
3. **性能方面**：在Windows端运行帧率稳定在60fps，无明显卡顿；在Android端运行流畅，但由于触屏操作灵敏度较高，出现过玩家误触导致球拍移动过快的现象。整体来看性能符合预期。
4. **用户体验方面**：受邀玩家普遍反馈小游戏“简单直观”，但电脑对手难度偏低，缺乏紧张感。玩家提出的改进建议包括：增加球速随时间递增的机制、为击球事件添加音效反馈、在画面边缘加入分数显示等。

**结论与改进**

通过以上测试，可以确认Pong小游戏已能够在视觉小说框架中稳定运行，操作与判定逻辑正确，达到了本次课程设计对于“数据结构在游戏中的应用”的核心目标。其实现展示了在Ren’Py中通过Python类与状态变量维护实现互动逻辑的可行性，为视觉小说这一静态文本形式增添了交互性与趣味性。

然而，在游戏性方面仍有一定提升空间。例如，可以在代码中引入难度调节机制，利用动态调整电脑速度或球速提升来延长可玩时间；同时，在分数统计和失败条件方面增加更多反馈元素，使小游戏更符合现代玩家的习惯。若未来有更长的开发周期，还可结合数据库或外部存档记录玩家成绩，以进一步展示数据结构与算法的应用深度。

[（4）其他设备测试](#_Toc19263)

[（5）小组外玩家测试](#_Toc19263)

[结 论](#_Toc12804)

通过本次基于Ren’Py的视觉小说游戏开发实践，小组不仅成功完成了一款可运行的实验性作品，还在其中展示了数据结构与算法的实际应用。特别是在嵌入Pong小游戏的过程中，我们深刻体会到游戏逻辑中对状态变量的维护、碰撞检测与实时响应算法的重要性。这一模块成为整个课程设计的亮点，也有效体现了课程目标。

整体而言，本项目具有以下成果与意义：功能实现完整，视觉小说主线剧情、分支逻辑与小游戏均能正常运行；数据结构应用突出，小游戏展示了位置变量、状态变量和简单算法的实际作用；团队合作有效，小组成员分工明确，能协同解决开发过程中遇到的技术与资源问题；测试结果理想，在PC与移动端均能流畅运行，玩家体验良好。

当然，本项目也存在不足之处。例如，音效素材依赖外部资源，原创性有限；小游戏难度设置较为单一，缺乏持续挑战性。未来的改进方向包括：引入更复杂的数据结构与算法（如得分统计、难度递增机制）、提升美术与音效质量，以及尝试与数据库结合，扩展玩家数据记录功能。

总的来说，本项目顺利完成了实验要求，不仅实现了“从需求分析到最终发布”的完整开发流程，也为后续更复杂的游戏开发提供了宝贵经验。

[参考文献](#_Toc1309)

[1] Ren'Py. Ren'Py 中文文档[EB/OL]. [2025-08-22]. <https://doc.renpy.cn/zh-CN/>

[2]

致 谢

感谢张义青老师