学号： 201410311072

上海海事大学
本科生毕业设计(论文)

基于 Cocos 引擎的跨平台飞行射击游戏的设计与开发

学　　院：信息工程学院

专　　业：计算机科学与技术

班　　级：计算机 141

姓　　名：杨睿

指导教师：栾翠菊

完成日期：2018 年 5 月 13 日

承 诺 书

本人郑重承诺：所呈交的毕业论文《基于 Cocos 引擎的跨平台飞行射击游戏的设计与开发》是在导师的指导下，严格按照学校和学院的有关规定由本人独立完成。文中所引用的观点和参考资料均已标注并加以注释。论文研究过程中不存在抄袭他人研究成果和伪造相关数据等行为。如若出现任何侵犯他人知识产权等问题，本人愿意承担相关法律责任。

承诺人（签名）：_____

日期：　　年　　月　　日

摘　要

曾经被视为“洪水猛兽”的游戏，如今也在市场经济中担任了不容小觑的角色。游戏并非玩物丧志，适度游戏能够唤起积极的情感，成为人类生活的有效调味剂。

随着前沿技术与标准的一代代更迭，游戏开发门槛不断降低，游戏的实现方式也多种多样，而 HTML5 与 Cocos 引擎正是能够展现其魅力的途径之一。而且随着游戏的发展，游戏平台的主流不断向着便携移动趋势发展，走向全民娱乐时代。与此同时，微信推出的小游戏，或许将成为未来 HTML5 与 Cocos 引擎展现自己实力的一大平台。

微信小游戏以及 Cocos Play 等平台现今所推出的游戏，基本皆为休闲类型的小游戏。而因为目前浏览器与设备性能等限制，微信小游戏与 Cocos 引擎暂时并不适合开发重度游戏，当然也因为其即开即用、轻便快捷、迭代迅速、社交化、跨平台的特色，而成为了休闲类小游戏最适合的开发环境。

本毕业设计旨在开发一款基于 Cocos 引擎，运行于 Web 端，以飞行射击为主题并可发布至 PC 、手机、微信小游戏等的跨平台休闲类游戏。本游戏上手简易，界面交互友好，并且生态齐全，搭建有运营维护的网站的前后台。同时，由于其基础逻辑样式原理等使用 JavaScript 来实现，在编写样式组件的过程中，还开源实现了一套统一风格的前端样式主题。

本设计的创新点在于基于 Cocos-2d 引擎，使用 JavaScript 进行编写，通过 Html5 的 Webgl/Canvas 进行渲染，使得游戏能够跨平台运行于任何支持浏览器环境的设备，同时还对不同大小的设备进行了适配。游戏玩法新颖，有着完备的物理引擎与碰撞检测，通过子弹的反弹等进行战斗，还设置了道具、环境重力等各种有趣因素，为游戏策划大赛获奖之作。用户中心通过接入微信小游戏 api 实现，可以方便地进行登录，信息存储，与社交分享、排行。前台网站基于 Hexo 框架构建，接入 GitHub Issue 作为评论系统，可以实现无后端发布文章与评论。游戏与同一风格的前端样式组件均采用模块化编写，易于拓展与复用。后台则采用 Lumen ，使用 RESTFUL 风格搭建 API ，实现前后端分离。

关键词：Cocos； Html5；微信小游戏； Vue； Hexo

Abstract

The game that was once regarded as a "great beast" has now played a role in the market economy. The game is not bad, moderate games can evoke positive emotions and become an effective flavoring for human life.

As the cutting-edge technologies and standards have changed from generation to generation, game development barriers have been continuously reduced, and the realizing ways of game are also varied. The HTML5 and Cocos engines are just one of the ways to show their charm. And with the development of the game, the mainstream of the game platform continues to develop toward the trend of portable mobile, to the era of universal entertainment. At the same time, the small game launched by WeChat may become a platform for future HTML5 and Cocos engines to show their strength.

The games launched on platforms such as the WeChat game and Cocos Play are basically casual type games. Due to the limitations of current browsers and device performance, WeChat games and Cocos engines are not suitable for heavy game development at the moment. Of course, they also become instant, lightweight, fast, iterative, social, and cross-platform. The most suitable development environment for casual games.

The purpose of this graduation project is to develop a cross-platform casual game that is based on the Cocos engine and runs on the Web side, with the theme of flying shooting and can be released to PCs, mobile phones, and Wechat games. The game is simple to use, friendly to interface, and ecologically complete. At the same time, due to the use of JavaScript to implement the basic logic style principles, in the process of writing style components, a set of uniform style front-end style themes was also realized in open source.

The innovation of this design is based on the Cocos-2d engine, written using JavaScript, and rendered through Webgl/Canvas of Html5. This allows the game to run across any platform that supports the browser environment, is adaptive as well as for devices of different sizes. The gameplay is novelty, with a complete physics engine and collision detection, fighting through bullet rebounds, and various interesting

factors such as props, environmental gravity, etc. The user center is accessed through the WeChat mini game api, which allows easy login, information storage, social sharing, and ranking. The front-end website is based on the Hexo framework and uses the GitHub Issue as a commenting system, enabling post-free publishing of articles and reviews. Both the game and the same style of front-end style components are modularized and easy to develop and reuse. Lumen is used in the background and APIs are built using the RESTFUL style to achieve separation between front and back ends.

Keywords: Cocos; Html5; Wechat Mini Game; Vue; Hexo

目 录

插图索引

表格索引

主要符号对照表

VS Code	Visual Studio Code
RESTful	Representational State Transfer
SPA	单页应用程序 (Single Page web Application)
IDE	集成开发环境 (Integrated Development Environment)
ES6	ECMAScript 6 (JavaScript 第六代语法标准)
NPC	Non-Player Character (非玩家角色)
AI	Artificial Intelligence (人工智能)
PC	Personal Computer (个人电脑)
NPM	Node Package Manager (Node 包管理工具)
LFS	Large File Storage (大文件存储)
Prefab	预制体
Vec	向量
foe	敌人
hp	health point (生命值)

第1章　引言

游戏也许能改变世界。

Reality Is Broken: Why Games Make Us Better and How They Can Change the World?[1]

游戏是通往未来的线索，著名未来学家 Jane McGonigal 在她的书中如是写到。

游戏是未来人类不可或缺的娱乐途径，同时游戏产业也是市场经济的重要组成部分。如今的游戏主流正向着轻巧、便携发展，使用 HTML5 与 JavaScript[2] 进行底层编写的轻量游戏，因其跨平台、轻便快捷、迭代迅速等特性，而拥有着广阔的发展前景。与此同时，老牌完备的 Cocos 游戏引擎也进军 Web 端，为游戏的一站式开发提供了更加便捷的渠道。

1.1　背景

2017年12月19日，由国家新闻出版广电总局主管，中国音像与数字出版协会等承办的 2017 年度中国游戏产业年会在海口成功举办。会上中国音数协游戏工委（GPC）、伽马数据（CNG）、国际数据公司（IDC）联合发布了《2017年中国游戏产业报告》[3]，总结了中国游戏产业的整体状况。数据显示，2017年中国游戏市场实际销售收入达到2036.1亿元，同比增长23.0%。

毫无疑问，游戏产业已然成为了如今我国市场上的一大热点，其蕴藏着的潜力也不断吸引着众多国产游戏的诞生。同年，国家新闻出版广电总局批准出版游戏约9800款，其中国产游戏约9310款，进口游戏约490款，自研比例高达95%。而最为重头的移动游戏，也有95%属于自主研发。2017年自主研发网络游戏海外市场实际销售收入达到81.6亿美元，相比2008年的0.7亿美元，十年暴增百倍以上。

而在2008年，中国游戏产业收入仅为185.6亿元，十年之后的2017年，中国游戏产业收入激增到2036.1亿元，十年增长了11倍，从2012年开始，游戏产业收入开始爆发式的增长，而这正是移动游戏开始兴起的年代，到了2017年游戏产业收入已经高达2036.1亿元，移动游戏收入更是突破千亿大关。目前全球活跃游戏玩家总数约为22亿人，而中国游戏玩家规模已有约5.83亿人，而其中移

动游戏用户规模更是达到5.54亿人。移动游戏市场收入1161.2亿元，成为游戏产业的新龙头，且份额持续增加，占57.0%。

广泛的群众基础，与丰厚的市场利润，以及不受限制的时间地点，使得移动游戏逐渐成为游戏产业主流，开启了全民娱乐时代。

2017年11月15日下午，中国目前市值最高的腾讯公司公布了截至2017年9月30日的第三季度综合业绩。财报显示，其第三季度收入人民币652.1亿元，比去年同期增长61%，而其中网络游戏收入增长48%至人民币268.44亿元，占比更是达到41.2%。2017年12月28日，腾讯旗下微信团队发布了最新版本6.6.1，微信小程序蓄势已久的「微信小游戏」终于到来。微信官方自身首发了包括风靡朋友圈的“跳一跳”在内的16款小游戏，其中半数采用了 Cocos 引擎制作完成。Cocos Creator 1.8 也同步发布微信小游戏支持。

早在2014年10月29日，万维网联盟便正式宣布，HTML5 标准规范制定完成。HTML5 的出现使得诸多开发者看到了 Web 端的潜力，其中所具有的设备兼容、多媒体、性能与集成等特性，和诸如 WebGL、Canvas 能够呈现出的表现与交互效果更是令人叹为观止。但百废待兴，新兴的 HTML5 虽然功能强大、效果出色，但仍旧缺少各式各样的类库与生态，且此时的 JavaScript 新标准也仍旧处于草拟之中。因此想要实现复杂的效果与功能，仍旧需要极大的开发成本。

2015年6月17日，ECMAScript（即 JavaScript 正式名称） 6 发布正式版本，即 ECMAScript 2015[4] 。并决定每年以年份命名，发布一个新版本，补充特性。且 JavaScript 的超集 TypeScript 也与之不断迭代更新，使得 JavaScript 摆脱了弱类型动态语言的一些劣势。2016年3月31日，Cocos Creator 1.0 正式版发布。Cocos 这一老牌原生游戏引擎，也终于正式进军 HTML5，试图在 Web 端尽可能还原原生体验。Cocos Creator 正是其推出的一个完整的游戏开发解决方案，包括了 cocos2d-x 引擎的 JavaScript 实现，以及更快速开发游戏所需要的各种图形界面工具。并可发布游戏到 Web、Android 和 iOS，以及点开即玩原生性能的 Cocos Play 手机页游平台乃至微信小程序，实现一次开发，全平台运行，大大降低了个人独立游戏开发的门槛。

1.2 目的

是游戏，让我们在无事可做时有事可做。所以，我们才把游戏当做“消

遣”，视为填补生活空隙的调剂，但它们远比这些重要得多。它们，是通往未来的线索。它们此刻辛勤培育的东西，或许正是我们日后唯一的救赎。

—— 哲学家 伯纳德·苏茨

曾经被视为“洪水猛兽”的游戏，如今也在市场经济中担任了不容小觑的角色。游戏并非玩物丧志，适度游戏能够唤起积极的情感，成为人类生活的有效调味剂。

随着前沿技术与标准的一代代更迭，游戏开发门槛不断降低，游戏的实现方式也多种多样，而 HTML5 与 Cocos 引擎正是能够展现其魅力的途径之一。而且随着游戏的发展，游戏平台的主流不断向着便携移动趋势发展，走向全民娱乐时代。与此同时，微信推出的小游戏，或许将成为未来 HTML5 与 Cocos 引擎展现自己实力的一大平台。

微信小游戏以及 Cocos Play 等平台现今所推出的游戏，基本皆为休闲类型的小游戏。而因为目前浏览器与设备性能等限制，微信小游戏与 Cocos 引擎暂时并不适合开发重度游戏，当然也因为其即开即用、轻便快捷、迭代迅速、社交化、跨平台的特色，而成为了休闲类小游戏最适合的开发环境。

本毕业设计旨在开发一款基于 Cocos 引擎，运行于 Web 端，以飞行射击为主题并可发布至 PC 、手机、微信小游戏等的跨平台休闲类游戏。本游戏上手简易，界面交互友好，并且生态齐全，搭建有运营维护的网站的前后台。同时，由于其基础逻辑样式原理等使用 JavaScript 来实现，在编写样式组件的过程中，还开源实现了一套统一风格的前端样式主题。

本设计的创新点在于基于 Cocos-2d 引擎，使用 JavaScript 进行编写，通过 Html5 的 Webgl/Canvas 进行渲染，使得游戏能够跨平台运行于任何支持浏览器环境的设备，同时还对不同大小的设备进行了适配。游戏玩法新颖，有着完备的物理引擎与碰撞检测，通过子弹的反弹等进行战斗，还设置了道具、环境重力等各种有趣因素，为游戏策划大赛获奖之作。用户中心通过接入微信小游戏 api 实现，可以方便地进行登录，信息存储，与社交分享、排行。前台网站基于 Hexo 框架构建，接入 GitHub Issue 作为评论系统，可以实现无后端发布文章与评论。游戏与同一风格的前端样式组件均采用模块化编写，易于拓展与复用。后台则采用 Lumen ，使用 RESTFUL 风格搭建 API ，实现前后端分离。

玩一个小游戏才是正经事儿。 —— 张小龙

一款可以随时开始冒险的休闲游戏可以为生活添加很多趣味。

1.3 研究内容

本毕业设计旨在开发一个以飞行射击为主题的跨平台休闲类游戏，并主要运行于 Web 端。根据拟定的策划与想要实现的功能来看，内容和技术关键点在于浏览器端对 PC / 手机等设备权限的调用，场景对象等的抽象实现，地图相机跟随，碰撞检测，物理系统模拟，游戏角色 AI 的实现，对象池管理，官网与后台，微信接入，以及前端组件的模块化封装。

因此，本设计的游戏开发内容主要包括以下几个方面：

1. 输入控制，不同设备使用不同的输入模式，如PC使用键盘，手机移动端使用触屏虚拟按键或重力感应。
2. 界面关卡设计，对界面进行组件化编写，与资源绘制。
3. 游戏角色设计，完成游戏角色形象设计与动画绑定，编写NPC 的简单 AI 逻辑。
4. 对象设计，完成道具技能样式的设计与逻辑编写。对地图与游戏场景中的共用对象进行抽象编写。
5. 物理系统与碰撞检测，对参与碰撞的物体进行分类，实现碰撞计算，模拟重力与加速度等物理环境。
6. 全局状态监管，对游戏中各角色技能等情况进行监管，并在游戏结束后根据游戏情况进行奖惩结算。
7. 游戏官网，编写与游戏相同界面风格的样式，来搭建对游戏内容进行介绍，以及相关资讯的网站，并提供玩家意见反馈功能。
8. 前端框架组件，在搭建官网的过程中，将可复用的组件样式进行提取封装，构建一套基础的前端组件库。
9. 游戏后台，实现可对游戏的数值与用户进行管理，以及接收玩家的反馈意见的后台系统。

第2章　技术简介

本毕业设计所使用的技术主要基于 Cocos-2d 引擎，及 GitHub 上开源维护的框架，使用同样开源的 Cocos Creator 与 VS Code 编辑器。实现技术中使用了大量开源项目，所以本项目同样进行了开源。

2.1　引擎简介

Cocos 目前在全球拥有 110 万的注册开发者，30 万的月活跃开发者，遍布全球超过 200 个国家和地区。采用 Cocos 引擎开发的游戏覆盖市面全品类，在移动游戏全球市场份额占比 30%，中国市场份额占比 45%，是全球第一开源移动游戏引擎。

Cocos 引擎特点之一便是开源免费，使得个人开发者更加容易上手，开发成本低。其次功能专注于 2D，相比竞争对手 Unity3D 在 3D 领域的光彩，Cocos 在 2D 市场攻城略地，相对个人开发者来说制作小型休闲游戏更加适合。 2016 年末，Cocos 引擎官方宣布 Cocos Creator 编辑器正式发布，战略转移至属于 Html5 / Js 天下的 Web 端，并继续保留其跨平台优势。

Cocos Creator 是一个完整的游戏开发解决方案，包括了 cocos2d-x 引擎的 JavaScript 实现，以及具有更快速开发游戏所需要的各种图形界面工具。支持发布游戏到 Web、Android 和 iOS，以及点开即玩原生性能的 Cocos Play 手机页游平台与微信小游戏，实现一次开发，全平台运行。其以内容创作为核心的游戏开发工具，在 Cocos2d-x 基础上实现了彻底脚本化、组件化和数据驱动等特点。[5]

2.2　开发环境

Visual Studio 是美国微软公司的开发工具包系列产品。VS 是一个基本完整的开发工具集，它包括了整个软件生命周期中所需要的大部分工具，如 UML 工具、代码管控工具、集成开发环境等等。而其封闭、笨重与无法跨平台等种种缺点，则常常为人诟病。

2014 年，萨提亚　纳德拉出任微软首席执行官，对开源文化给予了更多的

关注。而全球最大开源社区 GitHub 年度统计上，微软员工的贡献代码量也跃居榜首。

开源社区是极为多样化的世界， Visual Studio Code 正是封闭的微软逐渐拥抱开源的一个代表作品。编辑器轻量跨平台，可以安装众多插件，只要稍加配置，便可以打造出自己专享舒适的 IDE 。由微软提供技术支持，社区共同维护，更新迅速，反馈和问题都能得到及时解决。而其轻量特性，对于以 JavaScript 为主要编写语言的 Cocos Creator 与开发周边网站来说，更是不二选择。

本设计为开发基于 Web 端的跨平台游戏程序，为减少兼容问题，使用目前市场份额最大的谷歌 Chrome 浏览器，与基于 Chrome 内核的 QQ 浏览器进行调试。同时，由于可发布至微信小游戏，而微信 WebView 使用 QQ 浏览器 X5 内核，使用 QQ 浏览器进行开发调试更加有代表性。

微信小游戏通过 Cocos Creator 打包，使用微信官方提供的微信开发者工具进行调试。微信小游戏实质为微信小程序的一种品类，使用其中小程序调试工具，开发者可以完成小程序的 API 和页面的开发调试、代码查看和编辑、小程序预览和发布等功能。[6]

开发工作均在 Windows 平台进行，最终运行于 Web 端，采用 ES6 语法进行编写。由于万维网联盟的存在，现今市场上存在的大部分浏览器都有着统一的实现标准。因此无需过多担心，不同平台的兼容情况。

2.3 工具平台

除了主要使用 Cocos 引擎与 VS Code 、微信开发者工具进行开发调试外，还使用了其他许多开源工具。开源即开放源代码，是一种软件发布模式，其他开发者可以根据不同的开源协议，使用与编辑源代码。随着开源文化的流行，越来越多的企业与个人开发者拥抱开源。

Git 是一个开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。GitHub 是目前全球最大的开源社区。 Git 最初是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。本项目也采用 Git 进行了版本管理，托管于 GitHub ，使得程序更加规整，便于维护迭代。

Hexo 是一个基于 Node.js ，快速、简洁且高效的开源博客框架。Hexo 使用 Markdown（或其他渲染引擎）解析文章，在几秒内，即可利用靓丽的主题生成静态网页。[7] 游戏官网采用 Hexo 搭建，可使用 Markdown 发布编译公告文章，

通过 Gitalk 插件接入了 GitHub Issue 作为评论系统。

Vue 是一套用于构建用户界面的开源渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。[8]其生命周期及许多思想特性与 Cocos Creator 相吻合，可以使用 Vue 开发编写 Cocos Creator 面板与网站前端。

Element 是一套为开发者、设计师准备的基于 Vue 的开源桌面端组件库。[9]是本设计开发统一风格前端时使用到的组件库，为此项目贡献了代码，并在此之上开发了 Ink 主题。后台管理网站使用了这一组件库与主题。

Laravel 是 GitHub 上 Star 最多的 Php 全栈框架，而 Lumen 是精简的 Laravel 框架。后台使用 Lumen 搭建 RESTful 规范的 API，采取前后端分离方式进行构建。

npm 为我们打开了连接整个 JavaScript 天才世界的一扇大门。它是世界上最大的软件注册表，每星期大约有 30 亿次的下载量，包含超过 600000 个包（即，代码模块）。来自各大洲的开源软件开发者使用 npm 互相分享和借鉴。包的结构使您能够轻松跟踪依赖项和版本。[10] 项目皆使用了 npm 进行了管理，前端组件主题还打包进行了发布。

第3章　需求分析

本系统总共可分为四大模块，游戏官网，游戏程序，游戏后台，前端框架。

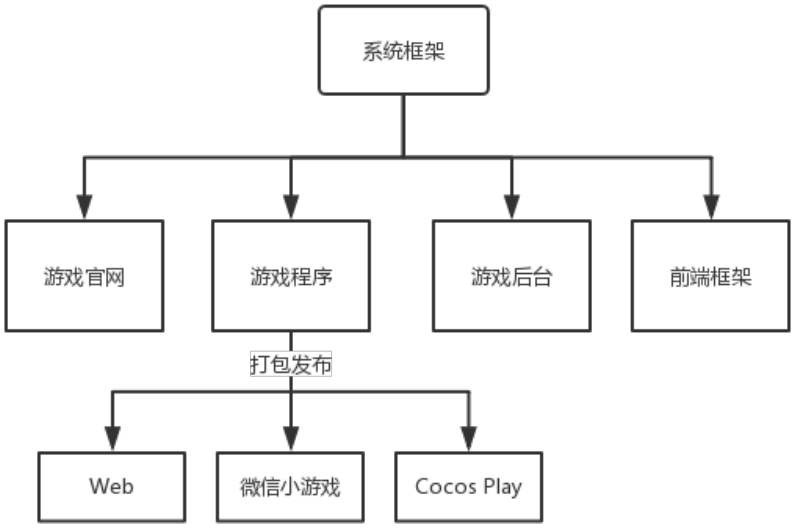


图 3.1　系统框架

本游戏系统采用模块组件化开发，尽可能实现高内聚低耦合。

3.1　游戏主程序

游戏主程序是本设计的中心，其基于 Cocos 2d 引擎制作，采用 Cocos Creator 编辑场景等节点，使用标准 JavaScript[11] 实现游戏逻辑。本游戏系统为独立开发的游戏，游戏内容为自主设计。故主要内容还包括游戏策划，游戏机制与游戏可行性等研究。游戏程序的实现部分则包括输入控制，界面与动画交互，游戏角色设计，NPC AI，对象池管理，地图相机跟随，碰撞检测，物理系统模拟，道具场景对象制作，全局状态监管，微信小游戏接入。用户中心通过接入微信 api 实现，可以方便地进行登录，信息存储，与社交分享、排行。

3.1.1 游戏内容

边沿被折起，收敛起双翼。折好的纸飞机被掷出后，却很少有人再去关心它的下落。漂流在外的纸飞机，在升空消失后，来到了属于它们的独一无二的纸宇宙。在这里，它们继续飞翔，继续那属于它们的冒险。

本游戏为飞行射击类休闲游戏，因此不强调竞技性和操作难度。游戏玩法新颖，有着完备的物理引擎与碰撞检测，主要通过子弹的反弹等进行战斗，还设置了道具、环境重力等各种有趣因素。

为展示其跨平台特性，对不同屏幕的设备进行了适配。同时设定了背景故事，纸飞机的星际冒险。①

3.1.2 输入控制

游戏的操作离不开输入设备的支持，不同设备使用不同的输入模式，跨平台游戏需要对不同的输入事件进行监听，如 PC 使用键盘，手机移动端使用触屏虚拟按键或重力感应。

PC 端通过浏览器对键盘事件进行监听，移动端则通过浏览器对触屏事件与位置信息等进行监听。制作方便操作的跟随形式虚拟摇杆。此外为了增加趣味性，展现 Web 端可能性，还增加了重力感应模式。

在浏览器端实现重力感应操纵，放在以往几乎是天方夜谭。但 HTML5 的诞生与浏览器的支持，使得浏览器端能够如同原生应用程序对手机的一些设备进行调用。 HTML5 有一个重要特性：DeviceOrientation[12]，它将底层的方向和运动传感器进行了高级封装，它使我们能够很容易的通过 JavaScript 实现重力感应、指南针等有趣的功能。

重力感应虽然有趣，但是有时对于玩家操作反倒并不方便，所以重力感应在游戏中作为一种游戏模式选择，而非主游戏强制使用。

3.1.3 界面与动画交互

作为一款休闲型的小游戏，界面呈现简洁统一的黑白线条风格。游戏中使用了 Iconfont 中的一些图标。当玩家进行操作时，给予其他动画细节反馈，如按钮放大变色等。当玩家出现、击杀敌人、死亡与复活，或者子弹碰撞消亡，敌人死亡时展现对应的反馈动画，提高用户体验。

① https://github.com/PaperStar/story

3.1.4 游戏角色设计

游戏主角为纸飞机，与场景统一风格，使用简约的黑白线条作为样式，同时与输入监听进行绑定，实现玩家对角色移动、射击等动作的控制功能。

纸飞机为不规则三角形，实现对玩家的边缘碰撞检测。能够随机生成于地图的不同位置，并随机赋予不同的颜色。

玩家作为一个类，拥有多种属性。同时继承 Cocos 中 Node[13] 节点，宽高透明度等基本属性。

3.1.5 NPC AI

游戏 AI 与现在市场上的智能助手实现的功能并不相同，游戏 AI 需要对游戏的不同情况进行判断，与玩家在游戏中产生交互，而无须其他额外复杂的功能。目前小型游戏的AI往往有着固定的行动模式，主要采用有限状态机进行实现。而大型游戏的 AI 算法过于庞杂，实现难度高但实用意义小。

本游戏系统编写了简单的 AI 逻辑，敌人分为两种行为模式，远程攻击与自杀式袭击。敌人在地图中随机生成，不同类型的敌人，使用不同的对象池。为保证运行速度和游戏体验，敌人同时存在的数量存在上限。

3.1.6 对象池管理

游戏画面默认通过 WebGL 进行绘制，如不支持 WebGL 加速，则使用 Canvas 进行绘制。同时，游戏中存在的对象均为一个节点，如果玩家频繁发射大量子弹，或场景中存在大量敌人，不断创建销毁节点，会使得游戏运行缓慢，降低 Canvas 刷新频率，影响游戏体验。

因此需要对存在的节点进行分类，使用对象池进行管理与复用。当子弹生存时间结束，将其状态置为不激活，放入子弹对象池。当玩家再次发射子弹时，若对象池中存在子弹节点，则直接激活子弹节点，不再重新创建节点。不同类型敌人死亡与生成同理。

3.1.6.1 地图相机跟随

游戏初始为 8000*8000 px 超大地图，使用相机组件跟随玩家移动，玩家始终处于地图中心。同时相机组件上编写动画效果，如镜头晃动，以方便游戏事件触发时展现交互效果。地图背景为白色，为防止缺乏移动标识，使用绘制的网格纹理进行动态拼贴，自动生成地图大小的背景。

3.1.7 碰撞检测与物理系统

游戏特色为模拟物理系统，玩家通过子弹反弹、碰撞或利用场景引力进行战斗。因此需要对，玩家、子弹、敌人、地图、场景物体等进行边缘检测。对需要进行碰撞检测的物体进行分类，并使用 Cocos Creator 中编辑器编辑各物体多边形轮廓。

游戏中玩家前进停止时，会产生纸飞机的滑翔效果。以及大幅度滑动虚拟摇杆时，会产生加速效果。游戏地图中，随机存在一些旋转的星球，当玩家接近一定距离后，会被引力所俘获。

子弹碰撞到玩家、敌人或地图边界，若碰撞次数有剩余，可以进行反弹，否则消亡不可见。敌人之间也可以被玩家引导相互进行碰撞，造成伤害。可见的物体，均可互相进行碰撞，伤害根据不同分组进行判定。

3.1.8 道具场景对象

地图中在不同位置随机生成不同样式的星球，与星球动画，以及道具。随机生成的位置坐标若已存在星球，则重新随机生成一个位置，直至有空位置容纳星球。星球为通过组件方式进行编写的预制体，类似玩家与敌人，具有属性，带有吸引周围物体的重力环，密度大，可移动，可被击碎。

地图中随机展现绘制的星星帧动画。

3.1.9 全局状态监管

游戏需要对全局的状态进行监管，其中包括各对象、玩家、敌人子弹等。编写游戏主逻辑，除 Logo 与故事板采用单独场景加载，其余存在于游戏场景，通过游戏主逻辑文件进行初始化，以传递得分、时间等不同数值。同时对游戏中各角色状态、道具等情况进行监管，并在游戏结束后根据游戏情况进行奖惩结算。

3.1.10 微信小游戏接入

游戏发布至微信小游戏，使用微信平台提供的 API 进行接入。以实现用户的登录，信息读取，头像加载。通过 LocalStorage 存储微信 API 返回的基础用户信息。通过微信开放数据域，实现排行榜功能。

3.2 游戏官网

一个完整的游戏往往有着一套完整的生态，包括游戏的官网介绍与后台管理等。一般游戏的官网主要放置游戏的内容介绍，获取渠道及相关资讯，基本由静态内容组成。纯静态内容网站，有着更良好的访问速度和更低的维护成本。现今很多开源项目的作者都会将自己的开源项目网页或文档托管于 GitHub Pages 上。而维护静态网站，编写不同资讯内容的文件仍旧十分繁琐。所以现在一些静态网站可以采用 Hexo 框架，通过编写 Markdown 文件在本地动态生成后，再通过 GitHub 进行部署，快捷方便。而这种方法实现自定义网站的缺点则是需要自己事先花费大量精力对主题进行自定义开发。

3.3 游戏后台

游戏后台则提供基本的游戏数值与玩家用户管理功能。开发后台需要良好的框架模式，所以一般使用已有的框架进行开发。采用前后端分离的开发方式，使用 Lumen 编写 RESTFul API 提供前后台交互。

3.4 前端框架

在搭建官网的过程中，将可复用的组件样式进行提取封装，构建开源一套基础的前端组件库。并可应用于后台样式。

第4章　系统设计

本章节主要介绍游戏系统及其相关模块的的设计方案、项目结构、分组管理、对象属性等。

4.1　游戏架构设计

游戏设计流程，为先对休闲类的游戏玩法模式进行了确定，拟定了游戏策划。随后将游戏功能拆分为输入控制、界面、动画、游戏角色、场景对象、道具等多个模块，并逐一实现。最终对各模块功能进行组合，进行全局监管、测试，并发布游戏。

游戏设计详细流程如图4.1所示。

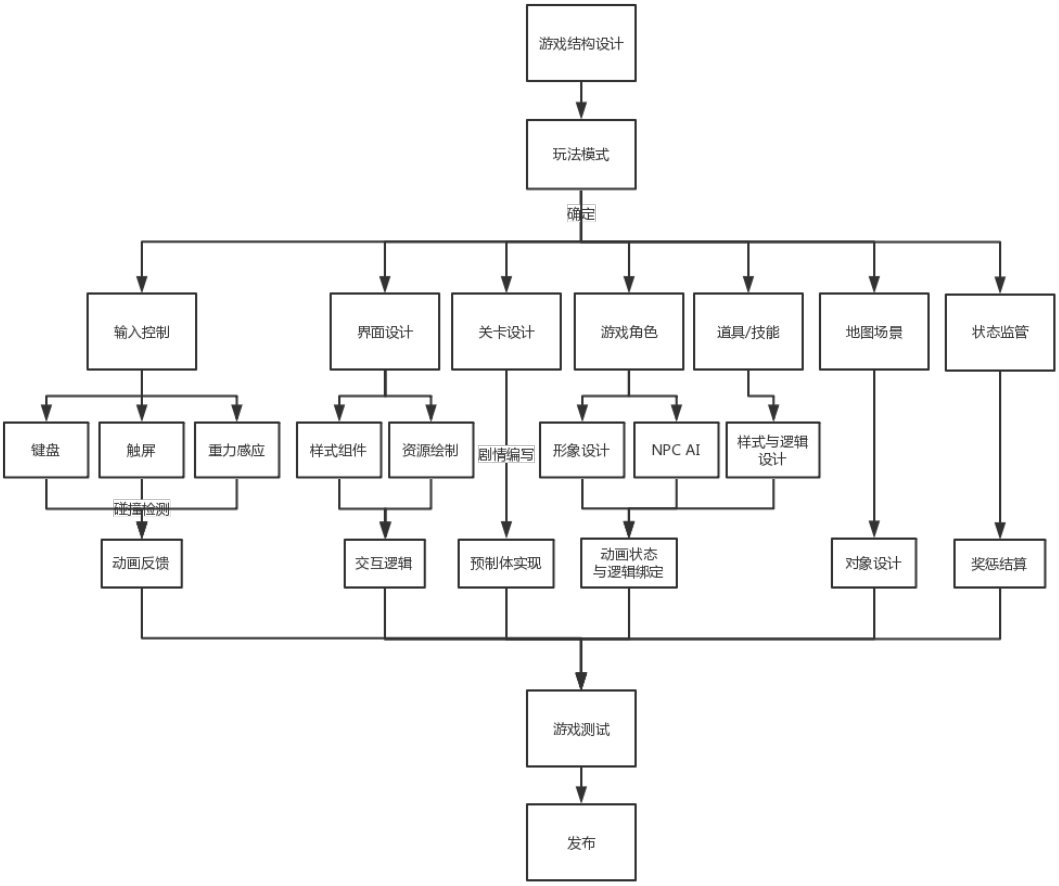


图 4.1　游戏结构设计

4.1.1　项目结构

项目基于 Cocos 引擎开发，使用了标准的 Cocos Creator 项目结构。并使用了 Git 进行了管理，通过目录结构的分析，即可清晰掌握游戏程序的模块划分。

游戏程序目录结构划分如表4.1所示。

4.1.2　游戏性设计

游戏为 2D 模式，玩家点击开始游戏，操纵纸飞机，随机出生在地图-纸宇宙中的一个位置，在平面上飞行，开始随机探索地图。地图中中设立道具、场景物品等各式触发事件，增加探索的乐趣。

纸飞机在地图中可以与其他物体进行碰撞，并会受到地图中行星的引力影响。玩家大幅度滑动虚拟摇杆时，可以进行加速。停止操控时，玩家角色并不会立即停止，而是模拟物理情况，在地图中滑翔一段距离。发射出的子弹同样

表 4.1　游戏主程序目录

文件/文件夹名	描述
.git	git 版本管理目录
.vscode	VS Code 编辑器工作区配置文件目录
assets	资源分类目录，放置游戏中所有本地资源、脚本和第三方库文件
build	打包构建目录，存放目标平台的构建工程
docs	游戏策划目录
library	library 是将 assets 中的资源导入后生成的，在这里文件的结构和资源的格式将被处理成最终游戏发布时需要的形式[5]
local	包含该项目的本地设置，包括编辑器面板布局，窗口大小，位置等信息
node_modules	npm 扩展包存储目录
settings	保存项目相关的设置
temp	临时文件存储目录
.gitattributes	游戏中使用到了许多图片资源，有些资源采用 Photoshop 自行绘制，故会产生 psd 文件，所占位置较大，采用了 Git LFS 进行管理
.gitignore	Git 版本控制过滤规则文件
creator.d.ts	Cocos Creator 语法提示文件
package.json	npm 包信息文件
project.json	记载当前使用的引擎类型和插件存储位置
README.md	本游戏程序说明文件

可以进行反弹。地图中随机产生敌人，玩家击败不同的敌人将获得不同的奖励。游戏特色，玩法新颖，玩家有时难以操控纸飞机直接击杀敌人，但可以利用行星的阻碍引导敌人撞击，或通过反弹的子弹来击败敌人。

单局游戏时间短，可以随时开始的特性，使得玩家更好的在闲暇时刻进行游戏。

4.1.3　预制体设计

玩家与敌人、子弹、以及场景中存在的对象，均采用了预制体进行抽象化。预制体存放于为项目 assets/Prefab 目录。

预制体目录结构如表4.2所示。

表 4.2　预制体目录

文件/文件夹名	描述
Object/bullet/Line	线型子弹预制体，直线型子弹，可调节宽高
Object/bullet/Line	链状子弹预制体，可以如双截棍般以中心点进行旋转
Role/Boss/Boss1	Boss1 浮空母舰，可以不断释放自杀式袭击的纸飞机
Role/Boss/Boss2	Boss2 黑色引力轰炸机，自身带有引力环
Role/Foe/Foe0	Foe0 移动迅速，普通自杀式袭击的纸飞机
Role/Foe/Foe1	Foe1 移动较为迅速，伤害低，远程攻击型飞机
Role/Foe/Foe1	Foe1 移动较为缓慢，伤害高，远程重装型飞机，生命值高，质量大
Scene/Planet/planet	地图上生成的星球预制体模板，内部编写了不同样式的随机函数
Scene/Planet/gravityradial	重力环预制体，为星球预制体所复用
UI/anim/five	自行绘制的动画五角星，可在地图上随机生成
UI/anim/four	自行绘制的动画四角星，可在地图上随机生成
UI/anim/two	自行绘制的动画十字星，可在地图上随机生成
UI/button	按钮预制体
UI/bar	敌人血条预制体

4.1.3.1　玩家属性

玩家作为一个继承了刚体的物理对象节点，还增加了许多自定义属性。玩家属性如表4.3所示。

4.1.3.2　子弹属性

子弹为场景中大量存在的节点，而游戏特色为子弹可以进行反弹旋转等，所以每一颗子弹都有各自不同数值的属性。子弹属性如表4.3所示。

表 4.3　玩家属性

属性	描述
fxTrail	移动时，尾部粒子动画
bulletType	子弹类型
bulletCollisionTime	子弹可碰撞次数
hp	初始生命值
curHp	当前生命值
score	玩家所取得的分数（当存在多个玩家时，以玩家为对象存储分数更易于管理）
life	生命条数
curExp	当前经验值
curLv	当前等级
roleColor	玩家颜色
isStop	是否停止
moveDir	移动方向
moveAngle	移动角度（由移动方向计算而来，以旋转玩家角色形象）
speedUpFlag	是否允许加速
moveSpeed	移动速度
normalSpeed	正常初始速度
accelSpeed	加速度
maxSpeed	最大速度
_delayFlag	子弹发射延迟
_shootFlag	是否允许射击

4.1.3.3　NPC 属性

NPC 主要为游戏中的敌人，敌人存在不同类型的纹理与碰撞边缘，但基本攻击模式与属性相似，均继承自基础的敌人脚本组件。NPC 属性如表4.5所示。

4.1.3.4　星球属性

星球为游戏中存在的场景对象，其属性中数值均可在地图初始化时，在一定范围内进行随机生成。

星球属性如表4.6所示。

表 4.4　子弹属性

属性	描述
bulletType	子弹类型
width	子弹宽度
length	子弹长度
moveSpeed	子弹移动速度
delay	子弹延迟发射时间
lifeTime	存活时间
collisionTime	剩余可碰撞次数
damage	伤害值
canBreak	是否可以被折断
brokenFX	折断效果
isVisible	是否可见

4.1.4　脚本设计

游戏中全程采用脚本组件化开发，通过 JavaScript 脚本对游戏角色、对象进行逻辑控制。脚本均继承 Cocos Component 组件，挂载至相应的节点上。同时进行了抽象化设计，不同敌人可以继承敌人脚本文件，再对各自行为模式与数值进行调整。

脚本均存放于项目 assets/Script 目录。脚本目录结构如表4.7所示。

4.1.4.1　游戏主逻辑

游戏主逻辑目录为游戏运行的主程序所在，结构如表 4.8 所示。

4.1.4.2　公用函数

全局公用函数目录，存放游戏中公用的一些工具函数以及全局信息。公用函数目录结构如表4.9所示。

4.1.5　场景、动画与资源设计

游戏场景分为启动 Logo 与故事板场景，游戏菜单场景，游戏运行主场景，游戏设置场景，成就排行场景。 Logo 与故事板介绍背景故事，故事板仅当玩家第一次进入游戏时出现，后续回看可从设置中选择。游戏菜单场景包括游戏开始界面，游戏模式选择界面，包括进入其他场景的选项按钮。游戏运行主场景，为游戏主逻辑运行场景，在同一场景中预先加载复活与死亡界面，置为隐

表 4.5　NPC 属性

属性	描述
foeType	敌人类型（不同大小、纹理的敌人）
atkType	攻击模式（近战、远程）
bulletType	子弹类型
bulletCollisionTime	子弹可碰撞次数
hp	生命值
curHp	剩余生命值
score	击败后可获取的分数
crashDamage	碰撞时，所造成的伤害
moveSpeed	移动速度
turnSpeed	转头速度
moveDir	移动方向
atkDir	攻击方向（由当前移动方向，不断向攻击方向转头）
atkDuration	攻击持续时间
atkPrepTime	攻击准备时间
fxSmoke	出现时的烟雾效果

藏，当用户事件触发时再进行显示。成就排行场景，通过接入微信开放数据域实现，存储用户的历史最高分数等。游戏设置场景，使用 LocalStorage 长久存储一些配置信息，进入游戏时自动读取，若没有设置则使用默认配置。

动画设计内容如表4.10所示。

资源目录包括游戏中使用到的一切素材，如图片，字体等。资源绘制内容主要为上表4.10 帧动画与玩家敌人地图场景中的纹理提供素材。资源主要以 psd 存储源文件，存储为 png 格式，并为节约空间，对碎图进行打包合成图集。资源一部分为自行绘制，一部分使用了 IconFont[①] 等处开源图标素材等。psd 文件与导出的图片存于一处，psd 后缀名文件使用 Git LFS 进行管理。

因所涉及资源过多，不再一一赘述。资源主要内容如表4.11所示。

4.1.6　碰撞管理与物理系统设计

可见的物体均模拟物理环境下的碰撞，存在不同的碰撞分组。不同分组之前进行碰撞时，根据分组设定，触发不同的回调函数。

分组管理如表4.12所示。

① http://iconfont.cn/

表 4.6　星球属性

属性	类型	描述
hp	Number	生命值
curHp	Number	剩余生命值
gravityRadial	Prefab	重力环（调用预制体）
radius	Number	星球半径
gravityRadius	Number	重力环影响半径
gravityForce	Number	重力
density	Number	密度
friction	Number	摩擦系数
angularVelocity	Number	自转速度

表 4.7　游戏脚本目录

文件/文件夹名	描述
Game	游戏主逻辑目录
Global	全局公用脚本目录
Lib	放置游戏中使用到的第三方库文件
Menu	菜单脚本目录
Player	玩家角色目录
Launch.js	游戏初始加载 Logo 与故事板的启动脚本
NodePool	对象池基本函数封装脚本
PoolMng	游戏中使用到的对象池管理脚本
Types	游戏中对象的分类脚本（如不同敌人，战斗方式，子弹等）

物理系统使用 Cocos 引擎中提供的刚体组件结合碰撞组件实现。另外添加了加速度、旋转速度等属性挂载于角色上。刚体组件可以还原模拟物理世界中的一些受力效果。

刚体属性如表4.13所示。

4.2　生态设计

运营一个游戏需要拥有完整的生态，即包括游戏官网与游戏后台。采用前后端分离的形式构建，前端使用打包压缩的静态文件，后端提供 API 接口。同

表 4.8　游戏主逻辑脚本目录

文件/文件夹名	描述
Game.js	游戏主逻辑文件，初始化其他主场景所需的脚本文件
Enemy	敌人脚本目录
Map	存放地图控制脚本
Object	对象脚本目录，存放诸如子弹、星球等中立对象
Render	渲染脚本目录，存放游戏中使用到的特殊效果
System	存放成就系统与排行榜脚本文件
ui	存放所有 UI 界面相关脚本文件
Wave	游戏中敌人进攻进度管理目录

表 4.9　全局公用函数目录

文件/文件夹名	描述
AnimHelper.js	游戏动画辅助脚本，动画播放前后发射事件触发其他函数
common.js	存储全局信息文件
global.js	暴露在全局下，可进行全局的一些调试设置
Helpers.js	工具函数，导入使用，如随机获取预置的颜色中的一种

时开发过程中将可复用的组件样式进行提取封装，构建了一套前端样式组件主题。

编写与游戏相同界面风格的样式，来搭建对游戏内容进行介绍，以及相关资讯的网站，并提供玩家意见反馈功能。游戏官网采用 Hexo 搭建，可使用 Markdown 发布编译公告文章，生成纯静态单应用页面网站，通过 Gitalk 插件接入 GitHub Issue 作为评论系统，部署于 GitHub Pages ，降低维护成本。接入 DaoVoice ，实现在线交流反馈。

一套基本的前端组件，需要实现不同设备的自适应功能、栅格系统与表单等基本组件样式。样式采用黑白简洁风格与游戏相统一，并降低设计的复杂程度。

其中编写 css 样式，想要保持统一风格，定义变量等功能必不可少，需要使用 less/sass/stylus 语言进行编写。现今 GitHub 上最受广泛关注的前端组件库

表 4.10　动画目录

文件/文件夹名	描述
begin/bgFadeIn.anim	背景渐隐渐显效果
begin/Logo.anim	游戏初始 Logo 动画
begin/paper-line.anim	游戏开始界面，线条纸飞机飞行动画
begin/paper-solid.anim	游戏开始界面，实体纸飞机飞行动画
begin/StoryLabel.anim	游戏故事板动画
bullet/break.anim	子弹破碎动画
bullet/vanish.anim	子弹消亡动画
camera/shake.anim	镜头晃动动画
foe/fadeIn.anim	敌人出现渐显动画
foe/foe0/dead.anim	敌人0死亡动画
foe/foe1/dead.anim	敌人1死亡动画
foe/foe2/dead.anim	敌人2死亡动画
foe/Boss1/dead.anim	Boss1死亡动画
foe/Boss1/prepare.anim	Boss1攻击准备动画
player/dead.anim	玩家死亡动画
player/revive.anim	玩家复活动画
player/start.anim	玩家进场动画
star/five-star.anim	五角星动画
star/four-star.anim	四角星动画
star/two-star.anim	十字星动画
ui/ClosePanel.anim	关闭面板动画
ui/OpenPanel.anim	打开面板动画
ui/combo-pop.anim	连续击中动画
ui/kill-pop.anim	连续击杀动画
ui/wave-pop.anim	进度提示动画

有 Bootstrap、Semantic-UI 等。Bootstrap 是使用广泛的前端组件库，其 v3 版本前采用 less 进行编写，v4 转用sass，其结构清晰，简洁易用，偏向于快速开发。而 Semantic-UI 则偏向于组件样式方面，对组件效果进行了很多美化。 Eleme 团队基于 Vue.js[8] 实现的 Element[9] 组件也十分美观，功能强大。

因此前后台的组件样式主题，使用 Scss[14] (Sass 趋向于 css 的改进语法) 和 Vue 进行编写，基于 ElementUI 开发。样式组件最后的代码使用 webpack[15] 进行打包压缩，并发布于 npm[10] 平台。

表 4.11　资源目录

文件/文件夹名	描述
anim	存放动画设计到的纹理素材
common	公用资源
font	字体资源（微信小游戏官方目前尚不支持自定义字体，但普通浏览器均支持）
icon	使用到的图标资源
particles	粒子动画资源
role	角色动画资源
scene	场景纹理资源
battle	战斗效果资源
efx	特殊效果资源

表 4.12　分组管理

分组名	gravity	scene	bullet	enemy	player	default
default						
player	√	√	√	√	√	
enemy		√	√	√	√	
bullet	√	√	√	√	√	
scene		√	√	√	√	
gravity			√		√	

表 4.13　刚体属性

属性	类型	描述
linearDamping	Number	线性速度衰减系数
angularDamping	Number	角速度衰减系数
linearVelocity	Vec	线性速度
angularVelocity	Number	角速度
density	Number	密度
friction	Number	摩擦系数
restitution	Number	弹性系数

第5章　系统实现

系统中实现细节与使用技术较多，为控制篇幅，此部分仅简要描述一些关键技术点的实现。

5.1　游戏关键技术实现方案

5.1.1　设备输入

设备输入包括键盘输入，鼠标、触屏输入，与传感器输入。皆需要对触发事件进行监听。

5.1.1.1　事件监听

其中键盘输入最为简易，对输入的键盘按下与抬起事件进行监听，判断字符即可。鼠标输入与触屏输入类似。触屏输入对用户四种触摸行为进行监听。(见表 5.1)

表 5.1　触摸事件类型[5]

枚举对象定义	时间名	触发时机
cc.Node.EventType.TOUCH_START	touchstart	当手指触点落在目标节点区域内时
cc.Node.EventType.TOUCH_MOVE	touchmove	当手指在屏幕上目标节点区域内移动时
cc.Node.EventType.TOUCH_END	touchend	当手指在目标节点区域内离开屏幕时
cc.Node.EventType.TOUCH_CANCEL	touchcancel	当手指在目标节点区域外离开屏幕时

Example：

```
node.on(cc.Node.EventType.MOUSE_DOWN, function (event) {
    console.log('Mouse down');
}, this);
```

重力感应的实现，调用 Cocos 封装的重力感应事件，将获得对象形式的陀螺仪信息，传入游戏对象进行相关移动计算即可，监听实现与触摸事件类似。

5.1.1.2　虚拟摇杆

虚拟摇杆基于触屏事件监听实现，制作为跟随形式虚拟摇杆。当玩家点击

屏幕时，将预置的虚拟摇杆节点置于触摸点坐标，离开则重新置为透明。设置内环和外环，当大幅度移动超出外环时，计算半径，内环始终不超出外环，允许玩家进行加速。计算触摸点在屏幕上移动的方向作为玩家移动方向，并旋转相应角度。

虚拟摇杆如图 5.1 所示。可以出现且仅出现在手指触及屏幕的位置，内部圆环可拖动。

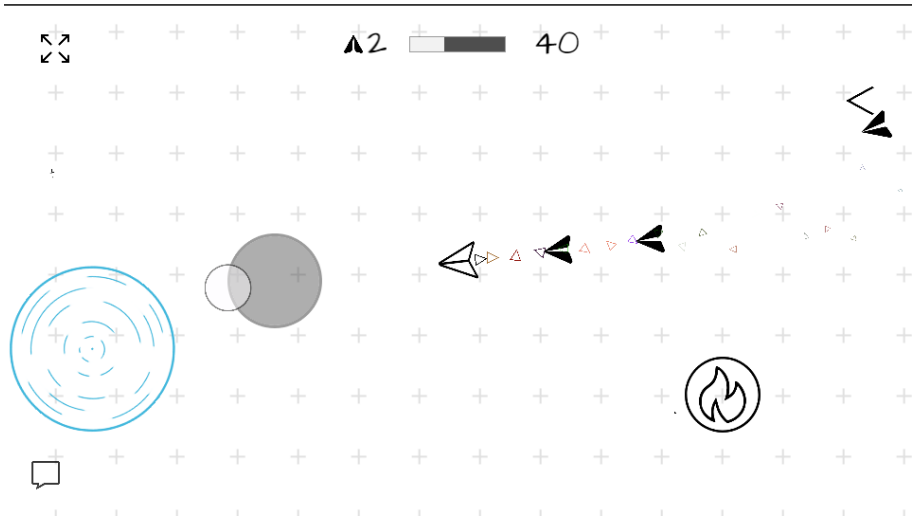


图 5.1　虚拟摇杆

5.1.2　对象池

对象池就是一组可回收的节点对象。[5]

为游戏中每一个可能同时存在多个的 Prefab 创建一个对象池实例。如子弹、不同种类的敌人等。当需要创建节点时，向对象池中申请一个节点，如果对象池里有空闲节点，就把节点以对象形式返回，这时再调用写好的初始化函数，就可以将这个新节点加入到场景节点树中进行循环使用。而不再需要重复创建销毁节点。

当需要销毁节点时，调用对象池实例的 put(node) 方法，传入需要销毁的节点实例，对象池把节点从场景节点树中移除，然后返回给对象池。这样就实现了少数节点的循环利用。比如玩家连续发射了多颗子弹，但同时存在的子弹一般不会超过 100 个，只需要生成 100 个节点大小的对象池，然后循环使用。

5.1.3　设备适配

使用 Widget 组件，能使当前节点对齐到父物体的相对位置。适配方案即采

用 16:9 比例，先获取设备的宽度大小，将画布拉伸与设备同宽，高度多数情况可以溢出。此时再获取设备中显示的画布高度，使用 Widget 组件对游戏中的 UI 元素，进行相对位置的排布，适配不同的分辨率。适配测试了不同分辨率的 PC 和手机多种浏览器，即便 18:9 的小米 MIX2 全面屏手机也成功适配。

5.1.4 场景动画

通过 JavaScript 随机函数，对星球的各类属性进行一定范围内的随机。生成与地图中的任意位置，并通过数组存储所有地图上已存在的星球信息，对已存在星球的位置进行矩形检测，判断是否相交，若相交，则重新生成。

动画使用帧动画编辑，绘制不同帧的动画，获取触发事件后依序播放。如玩家复活画面，使用不同图片旋转、透明度等属性设置关键帧进行实现。（如图 5.2 所示）

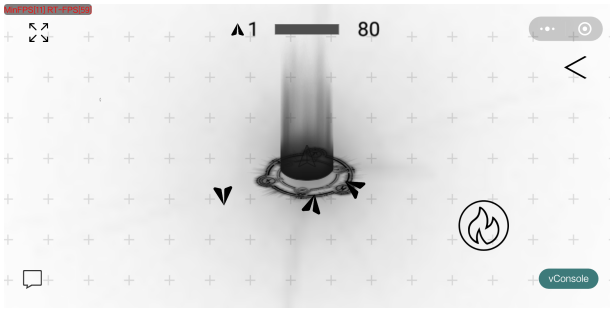


图 5.2　玩家复活画面

开始菜单的星球动画，使用 Graphics 组件绘制，实质与 Html5 的 Canvas 组件相同，通过调用基础 API 进行线条、圆形等绘制，并在每秒执行六十次的渲染函数中不断重绘更新。

游戏开始界面如图 5.3 所示。

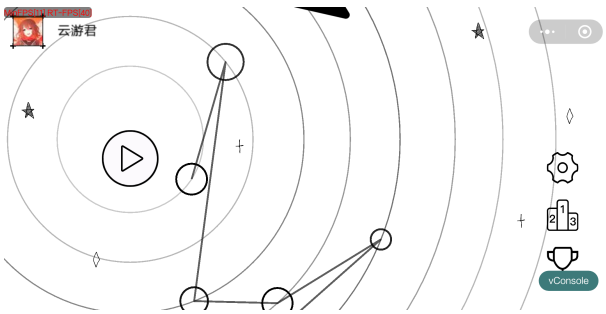


图 5.3　游戏开始界面

玩家死亡界面如图 5.3 所示。（浏览器端可以使用英文自定义字体，而微信小游戏暂不支持自定义字体。）

图 5.4　玩家死亡界面

5.1.5　碰撞检测与物理系统

对碰撞物体进行分组，并使用碰撞组件，绘制不同的碰撞边缘。当碰撞接触时，调用碰撞回调，对碰撞物体的名称进行判断，执行相应的函数。

场景中的对象继承 Cocos 中的刚体组件，赋予各自刚体的物理属性。玩家通过虚拟摇杆为玩家传入移动信息，玩家类中的函数则在 update 函数中不断更新数值，渲染新的坐标和反馈图像。

5.1.6　重力环

当星球预制体生成时，使用重力环预制体生成重力环。并随机配置引力大小与半径等属性。重力环的实现基于物理系统与碰撞组件，重力环实际为一个不可见的具有圆形边缘的传感器。当玩家或子弹与重力环发生碰撞时，对重力环之外的物体以星球为中心，施加一个向量的力，从而改变玩家移动的速度。

重力环为传感器，非实际碰撞物体，故不可见。可开启调试边缘绘制，查看挂载有碰撞组件物体的边缘。边缘检测调试界面如图 5.5 所示。

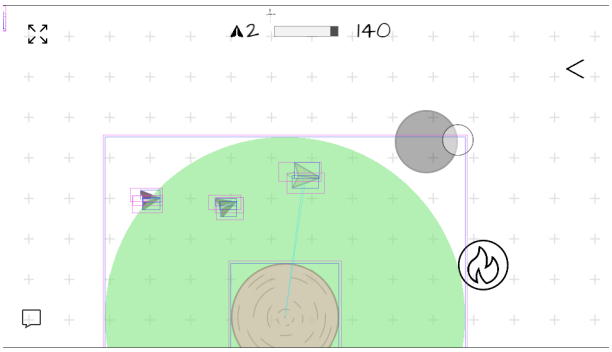


图 5.5　调试界面开启边缘绘制

5.1.7　镜头跟随

在没有摄像机组件的情况下，卷轴游戏都是通过移动场景节点或是游戏根节点来实现的，这样将会导致大量节点的矩阵都需要重新计算，效率自然会有所降低，而使用摄像机是直接将摄像机的矩阵信息在渲染阶段统一处理，将会比移动节点来移动屏幕更加高效。[5]

本游戏玩家可以自由探索地图，因此将玩家角色作为相机跟随的对象。同时地图与场景对象也为摄像机拍摄目标，其他未挂载的物体不会在屏幕中显示。场景中存在两种坐标系，世界坐标系与相对坐标系，获取玩家的世界坐标系，并设置摄像机与玩家的相对坐标系进行跟随。

5.1.8　NPC AI

NPC 在游戏中所做的主要工作，便是寻找玩家并将之消灭。每一个 NPC 都继承了编写的 Foe 脚本，赋予了不同的行动模式与相关移动攻击属性。4.5 NPC 近战逻辑为计算 NPC 与玩家之间的向量，并以各自的转头速度和移动速度，向玩家所处位置移动。 NPC 远攻逻辑为计算 NPC 与玩家之间的向量，与 NPC 的攻击范围进行比较，并以各自的转头速度和移动速度，向玩家周围位置移动，并在抵达玩家一定距离范围内，停止前进。当玩家处于 NPC 攻击范围内，在准备时间度过后，则以当前所处位置与玩家位置的向量差值发射子弹。

5.1.9　数据存储

游戏为单机游戏，游戏中的本地设置通过 Html5 的 LocalStorage[12] API 进行存储。 LocalStorage 可以访问一个远端的本地存储空间的对象，并将数据对象以键值对的形式进行存储。游戏中数据对象使用了 Json 形式，以更加规范地存储历史记录等信息。

游戏结束界面如图 5.6 所示。

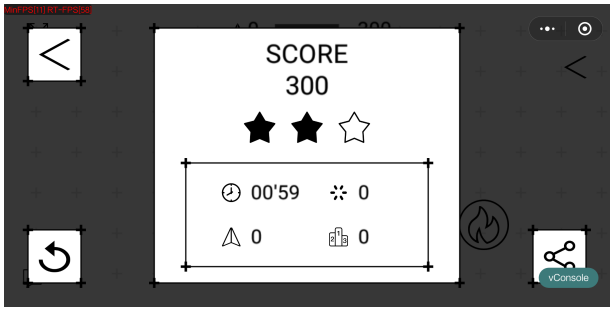


图 5.6　游戏结束界面

Example：

```
localStorage.setItem(app + '-' + 'records', JSON.stringify(
    records))
JSON.parse(localStorage.getItem(app + '-' + 'records'))
```

5.1.10　资源存储

微信小游戏的官方限制的包大小为 4MB 以内。而对于游戏来说，游戏的图片资源往往占据了程序大小的很大一部分比重。为了压缩游戏体积，使用了自动图集，即将图标等碎图拼接为一张大图，以压缩图片资源体积，并减少资源加载的请求次数。

5.1.11　微信小游戏接入

用户方面，则通过接入微信小游戏 API[6]， wx.createUserInfoButton ① 实现用户信息的读取。并读取微信头像链接进行远程加载资源，改变界面中头像。

微信小游戏为线上运行，需要预先加载资源以保证游戏流畅，微信小游戏加载界面如图 5.7 所示。

① 微信 2018 年 4 月 30 日发布公告废弃了历史的小游戏用户信息接口 getUserInfo

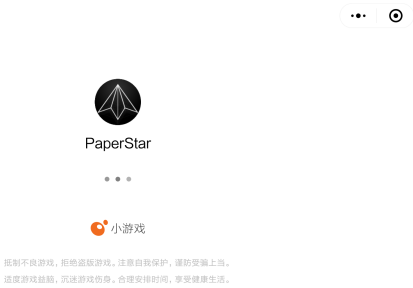


图 5.7　微信小游戏界面

同样接入微信小游戏 wx.showShareMenu 与 wx.shareAppMessage API，实现显示与主动拉起转发分享功能。微信转发一种为提供的右上角默认菜单样式，一种可通过游戏中按钮的函数进行触发。此处可见游戏结束界面（图 5.6）分享按钮。均可拉起微信通讯录界面。

微信转发与分享界面如图 5.8 所示。

(a) 微信主动转发

(b) 微信主动分享

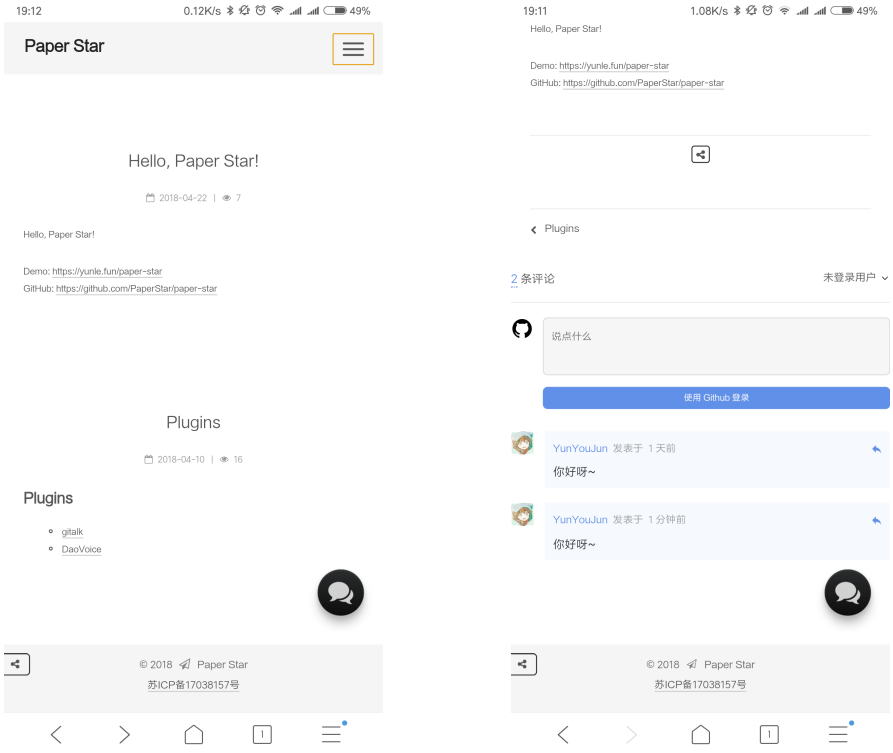
图 5.8　微信转发分享

5.2 游戏官网与前端组件实现效果

5.2.1 游戏官网

游戏官网实现了公告展示编辑，与意见反馈、GitHub 登录评论等功能[①]。网站进行了线上部署，并均采用了 https 加密。

游戏官网界面如图 3.2 所示。



(a) 官网主页　　(b) 文章评论

图 5.9　游戏官网

5.2.2 前端框架主题

完成了基本的样式组件开发和风格确立，以及预览网站[②]。

使用 NPM 发布 0.1.5[③] 版本。

主要模块：

- Basic

① https://yunle.fun
② https://ink.yunyoujun.cn
③ https://www.npmjs.com/package/element-theme-ink

- Form
- Data
- Notice
- Navigation
- Others

前端框架主题 Ink 预览网站如图 5.10 所示。

图 5.10　开源框架主题

第6章　结束语

这个游戏是自己很久以前的游戏策划，但却始终没有将其实现。而借本次毕业设计，我开始进行着手，并由此对整个游戏的开发实现流程有了更加完整的认识，而心里的某些东西似乎也终于向前推动了。

本次毕业设计花费了自己不少的时间心血，但一想到这是自己的选择，便继续坚持了下来。起初预想的内容比已实现的更加繁杂，但在一步步实现的过程中，也发现了许多目前可行性上的问题。比如网络同步、数据的实时存储等，所以许多最后的解决方案都采用了一些折衷的方案。当然，还有些不切实际的工作量。我还为其起了个英文名，PaperStar，并使用了纸飞机作为 logo。就如同一种试验，既因为游戏的主题是飞行射击游戏，又想将其看作一种尝试，就像纸飞机，是一种试验品，绘制于纸上，展示出 Web 端的无限可能性。 Ink 主题的命名也大抵如此。在实现的过程中，我都尝试使用了目前一些较为新颖的技术，所以也常常会遇到一些难以解决的 BUG，而最后不少都是通过开源社区与自己的思考解决了问题。一想到全世界无数人前仆后继，不求利益地为开源世界贡献着自己的代码，便不由为之感动。也衷心希望这个世界能够越来越开源。

我想如果物质极大丰富的那一天真的能够到来的话，那么开源的云端与能够让每个人都享受到乐趣的游戏一定是不可或缺的。

参考文献

[1] McGonigalJane. Reality is broken:why games make us better and how they can change the world[M]. 闾佳. Classic ed. 浙江, 中国: 浙江人民出版社, 2012

[2] Flanagan D. O'reilly: Javascript 权威指南[M]. 6th. 中国北京: 机械工业出版社, 2012

[3] 中国音数协游戏工委. 2017年中国游戏产业报告[R/OL]. 中国游戏行业发展趋势分析与预测报告, 2017. http://www.xinhuanet.com/info/2017-11/29/c_136786870.htm.

[4] 阮一峰. Es6标准入门[M]. 3th. 中国北京: 电子工业出版社, 2017

[5] 触控科技. Cocos creator v1.9.x 用户手册[EB/OL]. [2018-05-10]. http://docs.cocos.com/creator/manual/zh/.

[6] Wecaht. 微信公众平台 - 小游戏开发[EB/OL]. [2018-05-10]. https://developers.weixin.qq.com/minigame/dev/.

[7] Hexo. Hexo - a fast, simple & powerful blog framework[EB/OL]. [2018-05-10]. https://hexo.io/zh-cn/docs/.

[8] 尤雨溪. 渐进式 javascript 框架 - vue.js[EB/OL]. [2018-05-10]. https://cn.vuejs.org/v2/guide/.

[9] ElemeFE, Eleme Inc. Element-网站快速成型工具[EB/OL]. [2018-05-10]. http://element-cn.eleme.io/.

[10] npmjs. npm 中文文档[EB/OL]. [2018-05-04]. https://www.npmjs.com.cn/.

[11] 阮一峰. Javascript 标准参考教程[EB/OL]. [2018-01-04]. http://javascript.ruanyifeng.com/.

[12] Mozilla. Mdn web 接口[EB/OL]. [2018-05-04]. https://developer.mozilla.org/zh-CN/docs/Web/API/.

[13] Creator C. Cocos creator api[EB/OL]. [2018-05-10]. http://docs.cocos.com/creator/api/zh/.

[14] Sass. Sass 中文网[EB/OL]. [2018-05-04]. https://www.sasscss.com/.

[15] webpack. Webpack - module bundler[EB/OL]. [2018-05-04]. https://doc.webpack-china.org/.

致　谢

衷心感谢栾翠菊老师对本毕业设计选题的支持，和对本人的精心指导与关心。

感谢 GitHub 以及在此之上 Git, Cocos, Node.js, Vue.js, Element, Webpack, Hexo, Visual Studio Code 等项目的维护者及其开源精神，它们的存在让我得更加方便、快速地进行开发，而无需花费更多的金钱与时间成本，并使本毕业设计的实现成为可能。

感谢 LaTeX和 ThuThesis，使我更有效地规范了论文格式。

感谢家人一直以来对我的支持，感谢室友们陪我度过的大学时光，感谢所有在我遇到困难时给予我鼓励和安慰的每一个人。