

# 目 录

摘 要.....	3
Abstract.....	4
前 言.....	5
第一章 绪 论.....	6
1.1 研究背景.....	6
1.2 发展历史回顾.....	6
1.2.1 电影发展史.....	6
1.2.2 电子游戏发展史.....	7
1.2.3 手机游戏发展史.....	8
1.3 主流市场研究.....	8
1.3.1 腾讯互娱和王者荣耀.....	9
1.3.2 网易娱乐和阴阳师.....	9
1.4 开发平台分析.....	10
1.5 开发工具选择.....	11
1.6 本章总结.....	12
第二章 程序开发环境.....	13
2.1 Framework.....	13
2.2 .NET、C#.....	13
2.3 基于控制台的游戏程序.....	14
2.3.1 游戏目标.....	14
2.3.2 编程步骤.....	14
2.3.3 程序测试.....	15
2.4 Random 算法初步探究.....	15
2.5 Random 算法优化.....	17
2.5.1 双重随机数组.....	17
2.5.2 时间种子数组.....	17
2.5.3 GUID 随机数组.....	18
2.5.4 CSP 随机数组.....	19
2.6 本章总结.....	20

<b>第三章 游戏程序设计</b>	21
3.1 游戏设计文档	21
3.2 技术设计文档	23
<b>第四章 游戏程序实现</b>	24
4.1 开发环境	24
4.2 建立场景	24
4.3 角色添加	24
4.3.1 建立人物模型	24
4.3.2 添加控制脚本	25
4.3.3 射击功能实现	25
4.3.4 实现摧毁目标	26
4.3.5 人物跳跃功能	26
4.4 其他物体	26
4.5 算法应用	27
4.6 游戏逻辑	28
4.7 UI 搭建	30
4.8 添加音效	30
<b>第五章 Android 系统平台移植</b>	31
5.1 程序测试	31
5.2 手机端移植	31
5.3 拓展功能	32
5.4 项目总结	32
<b>参考文献</b>	
<b>致谢</b>	

## 摘要

人生有两大苦事，一是物质匮乏，二是精神空虚。随着社会生活水平的日益提高，物质匮乏的现象也就越发的少见了起来，取而代之的则是精神上的不满足，也就促使出现更多的娱乐方式，伴随着电子时代的到来，电子游戏作为新的艺术种类出现。经过三四十年的发展，电子游戏产业也渐渐的开枝散叶，个人电子设备的普及更是让电子游戏充斥着在每一个人的周围，尽管社会上仍然对于电子游戏的存在留有争议，但是电子游戏确实成为网络世界中的不可忽视的一股主流力量。在当今流量为王的时代，便携式游戏如何发展成为游戏界的必修课题，本次论文将跟随时代的潮流，重新审视电子游戏，尤其是便携设备上电子游戏的发展，如何正确的引导电子游戏发展，搭建优秀的产业生态圈确保电子游戏、网络文化和人类社会这三者的良性循环是本文的研究重点。最后利用常用的 Unity3D 开发软件基于 Android 平台，尝试开发一款属于自己的游戏。

关键词：游戏设计，Unity3D，C#，随机算法，Android

作 者：陆 洲

指导老师：华中伟

## Abstract

There are two great evils in life. One is the want of material things, the other is the emptiness of spirit. With the improvement of social living standard, the phenomenon of material shortage is becoming rare. Instead, it is the spiritual dissatisfaction, which promotes the emergence of more forms of entertainment. With the appearance of electronics, video games emerge as a new art category. The video game industry had been developing with a steady pace for about 40 years. The popularity of personal electronic devices makes video games can be seen everywhere. Although the existence of video games is still controversial now, it has indeed become a mainstream force that cannot be ignored in the network world. In network, Traffic is all, which way for mobile game become compulsory course in the video game industry, this paper will follow the trend of the times, reconsider the video games in the mobile platform. Try to make sure the electronic games, Internet culture and human society a virtuous cycle of these three. Finally, I will develop my own game program based on Android platform by using Unity3D.

Keywords: game design; Unity3D; C#; Random(); Android

Written by: Lu Zhou

Supervised by: Hua Zhong Wei

## 前 言

本课题为基于 **Android** 系统的游戏设计与开发，以 **Android** 平台上的便携式手游为研究对象，将在这些游戏中寻求相似的游戏功能模块和可值得改进之处。

本文共分为五个章节。第一章研究游戏发展历史和手游端市场行情，确立研究目标。第二章着重研究程序核心内容，脚本及函数方法。第三章将对所需编写程序的各部分概念和细节提出设计和规范。第四章为整体游戏的开发，从零开始编写出一个可运行的游戏程序。第五章为游戏程序的 **bug** 测试，移动端的移植，以及整个项目的总结。

# 第一章 绪论

## 1.1 研究背景

娱乐，一直以来是一项极其重要的人类活动方式，随着人类数千年的发展，娱乐的方式也在不断演变，在我国古代，琴棋书画并不是生存所需的必要物质活动，而在西方，戏剧也正是早在古希腊时候就已经出现，娱乐方式的出现很直观的体现一个时代的特征。经过数千年的沉淀和发展，其中大多数的娱乐方式已经有着属于自己成熟的发展体系，也就形成了所谓的“艺术”。

科技是第一生产力，21 世纪正处于以原子能、计算机出现为标志的第三次科技革命的浪潮之上，数字化、信息化、网络化是当前时代的代名词，也就不可避免的促使了娱乐方式向信息化方向的发展，电子游戏就是在这样一个合适的时机悄然到来。

## 1.2 发展历史回顾

历史总是惊人的相似，但又不是简单的重复。电子游戏作为九大艺术中的末座，无疑是所有娱乐形式中最为年轻，最不成熟的艺术形式。在探求电子游戏的发展前，我们不妨先回顾一下同为数字化娱乐产物的前辈，“电影”的前世今生。

### 1.2.1 电影发展历史

电影的出现的标志是在 1895 年 12 月 28 日，由法国的卢米埃尔兄弟在巴黎的一家咖啡厅里投射了 12 部纪实短片。在这之后电影分为了以卢米埃尔为代表的纪实派和梅里爱为代表的浪漫主义风格。一直到 1908 年大卫·格里菲斯的出现，创立了著名的蒙太奇镜头手法，在这个时期无声电影的发展初具规模，社会也将电影从戏剧中分离出来作为一门独立的艺术。再到 1927 年有声电影的出现使得电影不再只是画面的组合变换，后期的配音更是让电影立体化，电影技术真正成型并走向飞速发展时期。在 1946 年前后，在全世界笼罩在美苏两国的冷战氛围下，电影却丝毫没有收到约束与影响，出现了大批与政治、战争、宗教有关的优秀电影，同时好莱坞作为发展规模最大的电影工厂正逐步掌控电影市场，直至至今好莱坞成为了全球最著名的电影中心。

在电影史回顾中，不难发现电影艺术是由技术成型到艺术成熟两个发展进程。

### 1.2.2 电子游戏发展史

电子游戏本质上就是计算机软件，而软件的发展必然离不开硬件条件的成熟，在早期还没有电子游戏概念时，仅仅是电子工程师在工作之余进行消遣用的实验，而最早公开的游戏是用于在展会上娱乐游客的一款类似于网球玩法，在示波器上控制光点的视频游戏，这之后随着电子工业的发展更多的工程师效仿这种行文，有了电子游戏的雏形。在这之后，美国的电子工程师布什纳尔配合自己开发的电子游戏《电脑空间（Computer Space）》研发出第一款商用电子游戏机，并创立了最早的游戏公司——雅达利。

可惜由于公司的监管不力导致游戏市场被大量低质量、盗版游戏所侵略，在著名的 E. T. 事件后雅达利建立的游戏帝国一夜之间消失殆尽。与之同时，在北美的游戏帝国消散之际，日本的任天堂公司冉冉升起，在任天堂的经典红白机进入北美后，游戏市场仿佛注入了一剂强心剂，旗下的超级马里奥更是成为游戏界永远的传奇，直至今日这项 IP 也依旧活跃在超一线的游戏市场，在任天堂的带领下日本作为游戏大国先后涌现了 NEC、世嘉、索尼等等游戏大厂。

此外，在计算机的高速发展下，个人电脑走进了千家万户，虽然在游戏性能上不足商用机，但是网络游戏的发展促使了互联网的出现，PC 端的网络游戏似乎比起独立的游戏平台更为大众所接受。MUD 的出现更是奠定了网络游戏发展的基础，这种技术支持多人在同一个频道进行游戏，尽管当时还只是简单的文字类游戏，但是有无数玩家愿意支付每小时 12 美元的高昂费用。越来越多的游戏开发商都想在这块未开垦的处女地上分一杯羹，网络游戏产业初露锋芒，而 MMO 即大型网络游戏的概念也在当时提出，包月制成为当时游戏商收入的主要来源，也正是在这一时期大批优秀的网络游戏也呈现井喷式的爆发，这其中就有着由暴雪公司出品的《魔兽世界》，这款划时代的网游自发售后全面席卷线上游戏市场，以至于到今天所有的网络游戏都或多或少的借鉴了该作。

而随着互联网的进一步发展，开始兴起不需要单独下载客户端的网页游戏，但大多千篇一律，离不开与魔兽世界这类 MMORPG 类型的通俗套路，而逐渐成为开发周期短、内容空洞、缺少可玩性、刻意诱导消费这类没有灵魂的游戏，虽然网络游戏没有形成可持续发展的良性生态圈，但是其短周期、高回报的特点也吸引了更多的投资者的目光，为了之后游戏轻量化、便捷化的发展做出了不可忽视的贡献。

### 1.2.3 手机游戏发展史

可能在以前，没有现金出门可以说是寸步难行，但是现在的中国被称为无现金社会，出门在外没有谁的身上是不带着手机，随处可见支付用的二维码。手机不在是一项可有可无的奢侈品，而是一件实实在在的生活必需品，作为电子技术的结晶，手机的出现也给了电子游戏的发展提供了新的思路。

世界上手机游戏最早出现在了 1994 年的 Hagenuk 在其手机上安装的俄罗斯方块，而随后在 1997 年，当时的诺基亚在其产品中安装了控制光电移动的经典手机游戏——贪吃蛇。到了 2001 年 java 技术的问世和手机端彩屏出现，手机在软件和硬件都能更好的支持手机端游戏的运行与游玩，甚至诺基亚还专门推出游戏专用的手机，一度能与任天堂这样的专业游戏大厂共同竞争游戏市场。2007 年是手机历史上最为灿烂辉煌的一天，时任苹果公司 CEO 的史蒂夫·乔布斯推出了划时代的产品 iPhone，极具未来感的设计和简单傻瓜式的操作操作系统，使得购买的它的人接踵而至，由 iPhone 引起的消费狂潮，和其独特的 SDK 机制，使得手机的软件市场空前火爆，而之后的 Android 系统填补了中低端消费市场，原本混乱而无序的手机游戏市场走上了正规，市场上的手机游戏层出不穷，其中不乏很多优秀的游戏作品如《捕鱼达人》、《我叫 MT》、《Temple Run》以及一些其他 IP 改编或其他平台移植《梦幻西游》、《倩女幽魂》。

而在 2016 年手机游戏再创辉煌，由中国腾讯出品的 mmorpg 现象级手游《王者荣耀》引起了社会各界的关注，在其带领下，2016 年，手游以 57.2% 的市场占比超过端游市场，并且该数字还在持续的增长，在上游游戏的推动下，整个游戏市场收入剧增，游戏产品、电子竞技、游戏周边迅速完善。在 2017 年游戏市场规模突破两千亿元，其中移动端占比 63.8%，而游戏用户规模高达惊人的 6 亿人，24 岁以下玩家仅占 33%，也就是说游戏已经不再是孩童式的电子玩具，而是脱离了低级趣味，成熟、合理的娱乐方式。手游的盛行并不仅仅是在中国发生，《列王的纷争》就是由国内游戏公司开发主要面向欧美市场的优秀作品，并且在未来会有更多的手游公司开拓海外市场，产出国际性的对抗赛事，将手机游戏作为一项重要的文化输出。

### 1.3 主流市场研究

在上述的简介中也已经例举了部分数据，那么成为市场引领者的是哪些作品呢？这些作品有着怎样样的魅力呢？在这里选取了一些成功的手游案例来对游戏市场进行分析。



### 1.3.1 腾讯互娱和王者荣耀

说到王者荣耀就离不开其发行商，腾讯。游戏一直以来都是腾讯集团的重点，腾讯互娱共有五大泛娱乐板块：腾讯影业、腾讯文学、腾讯动漫、腾讯游戏、腾讯电竞，我们可以看到游戏有关的内容就占了腾讯总体战略的五分之二，腾讯在游戏领域的发展确并非一帆风顺，由于较晚的入市，早期中国游戏市场一直被网易、盛大、九游、等公司所霸占，而腾讯另辟蹊径战略性放弃了大型网络游戏，通过短期的搭建 QQ 游戏平台在游戏领域站稳了脚跟，在公司缺乏核心技术性人才时，腾讯通过代理的方式连连创造传奇，CF、DNF、LOL 这三款高娱乐、易上手、低配置但又爆火的游戏不仅仅给腾讯带来了创建帝国的资本以及将腾讯推向游戏界龙头老大的地位，更是帮助腾讯在游戏领域确立了未来的发展方针。

社交，可以说是腾讯的立身之本，无论在其发家的社交应用上还是令其辉煌的游戏领域中，庞大的用户量使得腾讯在手机平台上的争夺如虎添翼，曾经席卷朋友圈的天天游戏系列，就是最好的证明，可能那些 2D 纸片小人在平台上奔跑显得十分简单幼稚，但是腾讯巨大的用户量证明了腾讯无论是自研还是代发游戏都会有令人满意的玩家数量。而直到《王者荣耀》的出现，社交游戏才真正得发挥出他的实力。

娱乐、简单、互动，这三个特点无一不在王者荣耀这款游戏上体现得淋漓精致，很多玩家都称为掌上 LOL，LOL 与手机这两种不同领域的简单结合，产生了爆炸般的化学反应。据统计在 2016 年，QQ 和微信月活跃用户分别为 8.68 和 8.69 亿，而王者荣耀作为一款合格的即时对战类型的手游拥有 2 亿月活跃玩家（MAU）的用户数量也就不足为奇。王者荣耀的成功，不只是自身的优秀素养，更多的是手机的便捷、互动和社交这样的平台优势。

### 1.3.2 网易娱乐和阴阳师

如果说时势造就了腾讯，那么网易就是就是模范游戏公司。早期凭借梦幻西游积累了大量的游戏用户与游戏经验，优秀的策划与运营维持着梦幻西游整个游戏的良好生态圈，再通过代理暴雪的热门网游，将网易的品牌形象深深植入在了每一个玩家的心中，及时的战略调整，也使得网易在手机端的角逐中始终没有跌落热销榜单。在 2016 年网易推出的阴阳师也是成为了位居第二的火爆游戏大作。

阴阳师展现的魅力与王者荣耀截然不同，如果说王者荣耀做到了最火，那么阴阳师就应该是手游中最好的作品。阴阳师是一款 3D 日式和风回合制 RPG 游戏，更多的来说是 3D

卡牌收集或者说角色养成游戏，王者荣耀填充了大量手机用户的碎片时间，而阴阳师则相反，大量玩家投入时间和金钱，似乎在这点上阴阳师更具备一些硬核游戏的特征。而阴阳师的闪光之处，已经超出了手游的范围，完全值得用审视 3A 级游戏大作的眼光来看待。

新颖的妖怪题材：阴阳师是以日本作家梦枕貊的同名小说《阴阳师》为背景，讲述了在日本历史上最为著名的阴阳师安倍晴明，其实起初网易对于游戏的定位是在日本发行，却没想到在国内先大红大紫了起来，独特的题材就让玩家在游玩之前充满了好奇。

唯美的和式画面：阴阳师给人的第一印象便是唯美，细腻画风、精致的建模、搭配上一些妩媚的动作设计以及后期绚丽的人物服装，也是日后大批游戏公司所纷纷相仿的特色之一。

强大的声优阵容：阴阳师创新性的引入了国内外优秀的配音演员，这一措施也间接地引入大量原本属于动漫、漫画等其他二次元的用户，通过解锁自己熟悉的声优台词而成为该游戏的忠实粉丝和免费宣传者。

良好的游戏运营：网易作为一家拥有足够资历的老牌游戏公司，一直在提高玩家的保有率，通过发放定期福利、增加游戏内容来确保玩家不会流失过快。并且在游戏内添加更多的细节、剧情，让玩家更具有沉浸感和认同感，做到将游戏文化与社会文化相结合。

独特的抽卡机制：作为网游，能够给发行商带来理想的收益是必然的，不同于其他的道具收费、垄断收费、时长收费，阴阳师引入了概率收费也就是所谓的抽卡机制，这项机制不是阴阳师的首创，但绝对是阴阳师能作为商业游戏典型案例的画龙点睛之笔。抽卡就是有一定的概率能获得游戏内道具的游戏活动，越稀有的道具获得的概率也就越低，所以这也是大量玩家愿意充值人民币消费或者消耗大量游戏时间来获得抽卡所需的珍贵资源，此外还有分享功能使其成为一种玩家之间交流、竞争的资本，大大加强了游戏的社交性、粘性和扩散性。

## 1.4 开发平台分析

既然市场的趋势是游戏向手机端方向发展，那么说到手机必然离不开一个选择，手机系统的选择，目前智能手机软件系统只有两种选择 iOS 和 Android。Android 系统是由 Google 领头的多家手机商联合开发而成，并在研发后的不久开源放出，凭借其优秀的兼容性几乎可以应用在所有的电子设备上，这也得意于开发 Android 的底部 Java 语言的功能强大和简单易用的特点，或许在语言结构上不如由 Object-C 编写的 iOS 严谨，但是就目前看来 Java 和 Android 的发展前景可以掩盖先天的不足。Android 系统将电子设备转化为一台虚拟机，将底部数据传送给虚拟机，再通过虚拟机来接受和发送用户指令。这样用户

甚至可以直接对每一个数据进行修改，而手机的运行效率将取决于每一家手机厂商对 Android 的修改和优化，这也就造成了整个 Android 市场的不可控性，单论数量而言，Android 平台的应用是远远大于 iOS 平台的，可惜的是这其中真正能正常运行、投放使用的应用十不存一。Android 更多的是作为 iOS 平台的试水和补充，但随着时间的发展，Android 阵营出现了诸如三星的 Touchwiz、小米的 MIUI 等更多更成熟的定制 Android 系统。

## 1.5 开发工具选择

工欲善其事必先利其器，既然想做出一款合格的游戏产品，那么合适的开发工具必不可少，随着游戏的发展，游戏专用的开发引擎也在逐渐增多，虽然大多数引擎都是内部的商业机密，但是由于手机端游戏市场出现时间较短，专注于轻量化的引擎也就非常的有限。

Unity3D 是少见的多平台综合型游戏开发工具，无论是 2D 或是 3D，都可以做到不俗的效果，也正是它的诸多优点成为了大多数独立开发者的首选引擎。

可视化操作：用户界面的设计就体现出其较于其他引擎更易上手的地方，所有的场景资源和文件资源都是以可视化的形式陈列在资源列表中，对于资源的使用就像我们日常对电脑文件夹的使用，点击、拖拽、右键就可以完成大多数的基本操作。

模块化功能：在 Unity 中每一个细小的功能模块都称之为组件，你可以很快的找到所需要用到的功能，外观、材质、动画片段、脚本控制、物理效果、粒子特效等等，引擎会自动拆解成每一小份，且相互之间完全不受影响，就算需要物体之间的交互，也会在可视化窗口高亮来提示你物体之间的联系。

专用化脚本：虽然 Unity3D 底部是由 C/C++ 编写而成，但是其运行环境则是 .Net 下的 Mono 平台，它支持两种脚本语言 C# 和 JavaScript，而这两种语言都不需要担心 C++ 的内存和效率问题，Unity 内部通过 DLL 将用户的脚本语言传输至 C++ 语言进行操作。

支持多平台：Unity3D 的跨平台能力是其他任何一款游戏引擎所无法超越的，他支持 Windows、Linux、MacOS、Android、iOS、Tizen、Wii 等等所有主流的电子设备平台，这也是因为其独特的 Mono 运行环境，不仅能将 C# 转化为 C++，更是能进一步的编译成 CLI（Common Intermediate Language），当需要运行的时候 CIL 再转化为目标平台的原生代码，尤其在手机端的开发中，你不需要编写任何的 JAVA/X-code。

一体化环境：Unity 提供了 AssetStore 这样的官方平台，不仅提供了用户之间的交流，还提供了大量的第三方插件，让 Unity 的功能更加完善，开发氛围更加稳定，让独立的开发者或者小型的工作室有寻求帮助、打破僵局的途径。

## 1.6 本章总结

在本章节中，通过对研究背景进行初步的分析，来探寻游戏发展的历史过程和未来趋向，作为资源较少的独立开发者，可以从其中认识到进入该市场需要做哪些应尽的准备和专业知识，在了解到市场需求后，可以更为针对地进行课程的研发，确立自身的游戏尽可能地具备以下三个特点：

易上手：无论从开发者还是消费者的角度来说，轻量化、简洁化的游戏流程对于双方都不会成为在游玩过程中的负担，较短的游戏周期也会直接将开发中遇到的错误降到最少、最易化解，来确保整个游戏结构完整、可以正常运行。

高娱乐：独立开发游戏很难在画面的精美度、玩法的丰富度、架构的复杂度、前期的宣发热度上与成熟的游戏厂商相抗衡，那么小游戏必须要有较为新颖的闪光点来吸引玩家，并且游戏应用的用途就是娱乐玩家，娱乐性就是测量一款游戏最好的标杆。

高互动：在手机端的游戏市场中，我们可以看到未来的游戏更加注重促进玩家间的交流，合格的手游不应该是将玩家圈在自己的游戏内，而应该让游戏成为玩家与玩家之间沟通的桥梁，联系的枢纽，愿意将自己的快乐同他人分享。

## 第二章 程序开发环境

### 2.1 Framework

后期的开发工具使用 unity，基本的脚本编译自然选择了与 unity 契合度最高的 c# 语言。计算机语言离不开一个词叫架构。但是“架构”一词并非是在计算机领域独有的专业术语，最早该词汇出自于建筑行业，架构的直译英文为 Architecture，而架构自身的解释是人们对一个结构内的元素及元素间关系的一种主观映射的产物。而计算机软件是基于虚拟的互联网中，并不是直观能看得见摸得着的实物，这也带来了互联网在管理上的不便，于是人们根据真实社会中的经验，将互联网的开发建造也要进行制式化。也就有了软件架构，描述软件整体和组件之间的联系，也就是计算机系统中的蓝图。软件架构关注的是组成软件的每一个组件，具体来说就是类和对象，而架构的目的就是用来规划开发者所要进行的开发活动，提供规范、统一的标准来满足不同的客户需求。

### 2.2 .NET、C#

.NET 是 Microsoft XML Web services 平台，而这个平台相当于一个官方的、提供给技术人员、面对个人化的网络资源库，在这个平台上所有的应用是使用 .NET Framework 类库编写。而整个架构的核心就是对于网络的应用，该架构提供了一个庞大的 CLR（公共语言运行库）来方便在 .Net 可运行的程序之间的数据传输，甚至是脚本转换，而在 .Net 平台中最为常见的就是 C# 语言。

C#(sharp) 语言是由 C/C++ 衍生出的 OOP 语言，与 JAVA 有着很多相似的地方，但是与 JAVA 相比有更高的运行效率和更多的底层操作。微软在软件开发一直处于垄断地位，作为微软的自身产品，在使用 C# 语言时也有着更多的便捷，同时在大多数程序员都有 C++ 的开发经验上，C# 的入门显得尤为简单。也许 C# 不是最好的计算机高级语言，但绝对是最符合开发趋向的语言选择。既然是微软的开发语言也就顺其自然的使用 VS (Microsoft Visual Studio) 来作为开发工具台。

## 2.3 基于控制台的游戏程序

既然是开发游戏，那我们从现实生活中取题，先在 CMD 控制台中模拟出一个简单的游戏程序，作为项目的开发实验，明确游戏运行逻辑。

### 2.3.1 游戏目标

#### 玩法说明

双色球投注区分为红球号码区和蓝球号码区，红球号码范围为 01～33，蓝球号码范围为 01～16。双色球每期从 33 个红球中开出 6 个号码，从 16 个蓝球中开出 1 个号码作为中奖号码，双色球玩法即是竞猜开奖号码的 6 个红球号码和 1 个蓝球号码，顺序不限

图 2-1 玩法说明

### 2.3.2 编程步骤

创建中奖数组→获得随机数字→获得中奖的红球编号→获得中奖的蓝球编号→获得中奖数组→输出

```
namespace CMDgame
{
    class Program
    {
        public class Core
        {
            int[] red = new int[6];
            int ired = 0;
            int blue = 0;

            private int getnum(int max)
            {
                int inum = 1;
                Random random = new Random();
                inum = random.Next(1, max);
                return inum;
            }

            public void getred0()
            {
                for (int i = 0; i < 6; i++)
                {
                    red[i] = getnum(33);
                }
            }

            public void getblue()
            {
                blue = getnum(16);
            }

            public void getarray()
            {
                getred0();
                getblue();
                Console.WriteLine("中奖数组为: "
                    + ToString() + " {" + blue.ToString() + "}");

                //普通数字输出方法
                //for (int i = 0; i < red.Length; i++)
                //{
                //    Console.WriteLine(red[i]);
                //}
                //Console.WriteLine();
                //Console.WriteLine(blue);
            }

            public override string ToString()
            {
                return red[0].ToString() + "--" + red[1].ToString() +
                    "--" + red[2].ToString() + "--" + red[3].ToString() +
                    "--" + red[4].ToString() + "--" + red[5].ToString();
            }

            static void Main(string[] args)
            {
                Core core = new Core();
                core.getarray();
            }
        }
    }
}
```

图 2-2 代码

### 2.3.3 程序测试

结果：失败，没有达到游戏目标。

原因：从图 2-3 中可以看出，红球中奖数字全部相同。

分析：红球中奖数组中的数字没有完全随机，但蓝球与红球为不同随机数字。

图 2-3 测试结果图

### 2.4 Random 算法初步探究

为何在随机算法中没有真正的产生随机数呢？查阅官方文本后发现 RandomClass 是一种伪随机的生成器，只能满足部分随机性统计需求的设备。伪随机算法是从一组有限的数字中根据固定的算法来选择出来。而初始化的随机数组被称为随机种子。上文错误的原因也就是在不同的随机算法中产生了相同的随机种子。

```
int maxNum = 100;

for (int i = 0; i < 5; i++)
{
    Random random = new Random();
    var ranNum = random.Next();

    Console.WriteLine("seed:" + ranNum);

    Console.WriteLine(random.Next(1, maxNum));
}
```

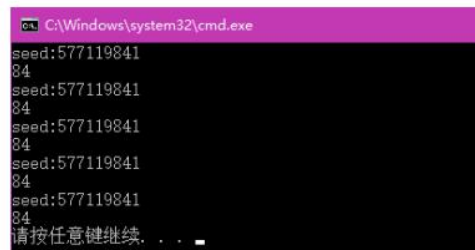


图 2-4 输出查看随机种子

随机算法的随机种子是根据计算机当前的时间作为依据，不同的计算机对于时间处理算法不同，算出不同的时间计数值提供给随机算法，但是在游戏要求不同的随机数算法时，无法将多个随机数分布在不同的计算机上。在不修改 Random 算法的情况下，是否能产生足够数量的随机数呢？

方法 1：多次重复获取随机数操作

```
int maxNum = 100;
Random random02 = new Random();
for (int i = 0; i < 5; i++) // 5 个数字
{ Console.WriteLine(random02.Next(1, maxNum)); }
```

该方法在同一个随机算法内重复获得随机数字，基本具备随机数字原则，但由于随机种子的固定，当获取的随机数字足够多的时候，会出现较多的重复。

```

int maxNum = 100;
Random random02 = new Random();
for (int i = 0; i < 100; i++) //100 个数字
{Console.Write("{0}\t", random02.Next(1, maxNum));}

```

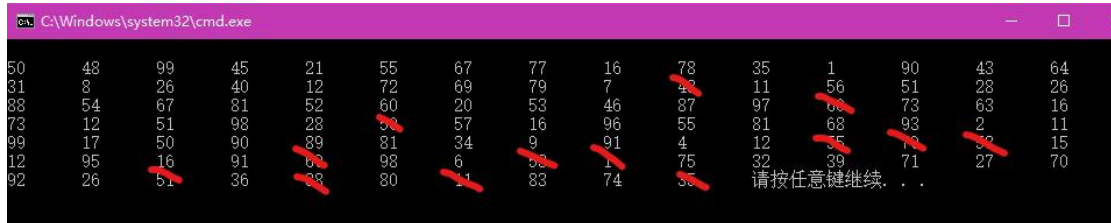


图 2-5 随机数重复测试

方法 2: Contains 判断是否出现重复数字

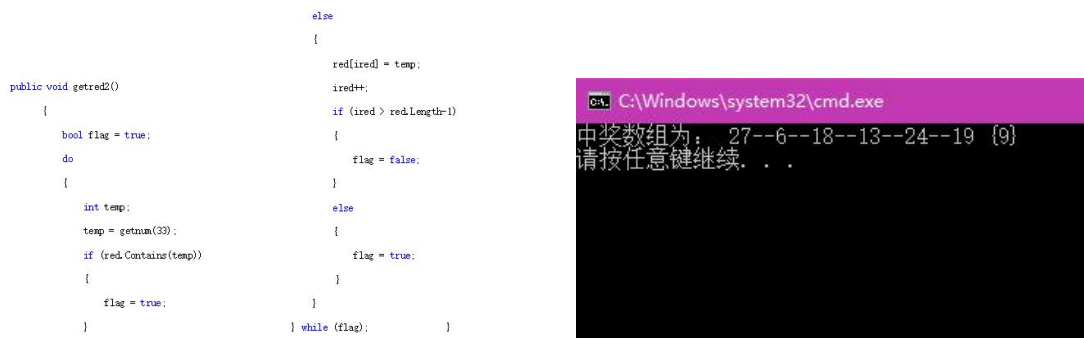


图 2-6 Contains 代码及结果

## 2.5 Random 算法优化

虽然上文提供了两种随机数的解决方法，但是没有从算法的根本上进行解决，接下来将会对随机算法的源头以及其他更加精准的算法方式解决。

### 2.5.1 双重随机数组



在知道随机数组是由随机种子所决定时，首先想到是否可以通过随机算法先得到随机种子再进行应用。

```
public int getRRnum(int max)
{
    int inum = 0;
    Random random0 = new Random();
    var seed = random0.Next();
    //Console.WriteLine(seed);
    Random random = new Random(seed);
    inum = random.Next(1, max);
    return inum;
}
```

在测试后，该方法失败，输出随机种子后发现，第一次产生的随机数无法更改，导致之后多次的随机算法都是以固定值作为依据产生，要想改变随机种子，应该将其数值进行多次的变化。

## 2.5.2 时间种子数组

原本的随机种子是以一组计算机当前时间为依据产生的，由于计算机性能较高，导致计算间隔非常短暂，那么如果将时间种子的精度提高与计算机运行速度相匹配时，就可以将随机种子进行不同的设置。

```
public int getTrnum(int max)
{
    int inum = 0;
    Random random = new Random
        (DateTime.Now.Millisecond);
    inum = random.Next(1, max);
    return inum;
}
```

在将时间精度提升至毫秒级别后，随机数组已经产生了变化，在不同的时间单位内获取的随机种子是可变的，该方法证明，只要随机种子产生了变化，最后所获得的随机数就更具有随机属性。

### 2.5.3 GUID 随机数组

与计算机的时间类似，每台计算机还会获得一个 GUID（全局唯一标识符）数值。而 GUID 数值具有极大的唯一性，理论上不会出现相同的 GUID 数组。

```
static void Main(string[] args)
{
    Guid guid = new Guid();
    Guid Nowguid = Guid.NewGuid();
    Console.WriteLine(guid+"\n"+Nowguid);
}
```

GUID 应用随机数组：

```
public int getGnum(int max)
{
    byte[] guid = Guid.NewGuid().ToByteArray();
    int seed = BitConverter.ToInt32(guid, 0);
    Random random = new Random(seed);
    int inum = 0;
    inum = random.Next(1, max);
    return inum;
}
```

从图 2-8 中看到实现了接近于真随机数的生成，并且完整的 128 位数字，具有重复的可能性极低，是目前最为精准、公正的随机数产生方法。

### 2.5.4 CSP 随机数组

在查阅 Random 方法的官方文档时，建议若要生成安全加密的随机数字时，应用 RNGCryptoServiceProvider 类[1]，也就是使用加密服务提供程序（CSP）提供的实现来实现加密随机数生成器（RNG）[2]。使用该类时，需要添加 System.Security.Cryptography

命名空间，它会根据计算机的当前总体状态，包括精准时间、运行程序、线程管理等多方面数据作为参考生成更加复杂随机的随机种子。

```
public void RNG()
{
    RNGCryptoServiceProvider csp = new RNGCryptoServiceProvider();
    byte[] byteCsp = new byte[10];
    csp.GetBytes(byteCsp);
    Console.WriteLine(BitConverter.ToString(byteCsp));
}

public int getCnum(int max)
{
    RNGCryptoServiceProvider csp = new RNGCryptoServiceProvider();
    byte[] byteCsp = new byte[10];
    csp.GetBytes(byteCsp);
    long Longseed = BitConverter.ToInt64(byteCsp, 0);
    int seed = (int)Longseed;
    Random random = new Random(seed);
    int inum = 0;
    inum = random.Next(1, max);
    return inum;}
}
```

图 2-7 CSP 代码

这是在所有方法上理论最为真随机的随机数组，但是高随机性也就造成了使用该方法时设备的高负担，在要求随机数字很多时，系统消耗会急剧增长。

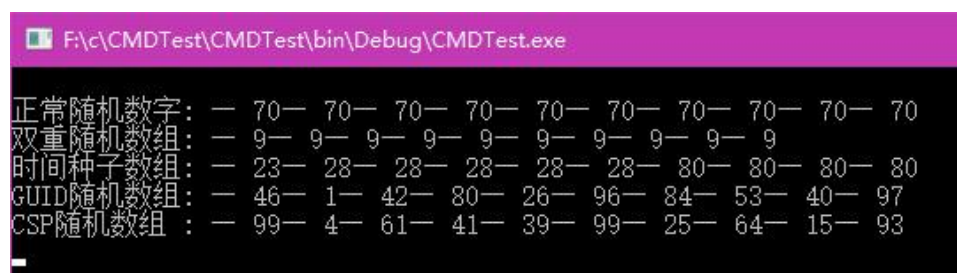


图 2-8 各算法输出结果

## 2.6 本章总结

脚本算法一直以来都是程序的运行基础，游戏也不例外。在本章中，我们通过一个简

单的抽奖算法来具体表现 C#语言在编写程序时的便捷。而在其中就程序中的随机算法进行了研讨，中奖概率、暴击概率、暴击倍率等等，随机数在游戏中的应用也是让无数的玩家吸引而来的重要原因，如何合理、公正的得到随机数字不仅是游戏机制的完整而且是作为销售者对于消费者应有的诚信。

## 第三章 游戏程序设计

### 3.1 游戏设计文档

游戏名称：穿越模型

平台：PC 单机、安卓

游戏类型：随机积分制 3D 第一人称射击游戏

游戏对象：休闲玩家

发布时间：2018/12/15

玩法说明：玩家将扮演一个实验用机器人，通过摧毁游戏目标来获得积分，积分根据游戏目标类型决定积分上限，一共有三种类型的敌人而玩家也有三种类型的武器。

游戏目标：随机算法的具体应用，和跨平台技术实现

游戏规则：玩家在一个封闭的区域内，每一次会在随机的地点刷新游戏目标，玩家可以通过控制移动、跳跃来辅助射击，玩家的武器有子弹数量的限制，摧毁目标可以得到子弹补给和积分奖励。

游戏玩法：

（PC）方向键或 wasd 进行移动，空格进行跳跃，点击鼠标开火

（移动端）移动设备通过触屏控制，玩家可以用两个手指来操作。左边触控到屏幕的手指可以移动角色，右边触控屏幕的手指可开火/跳跃。

游戏摄像机：游戏共有两个摄像机组件，一个为显示 HUD 的 UICamera，仅显示游戏信息；另一个为玩家视角的 MainCamera，视角摄像机会跟随玩家进行移动。

平视显示内容：

游戏时间：从计算游戏开始后计算时长，精确到个位值。

游戏得分：计算玩家所获得的游戏积分。

重新开始：重置当前场景状态。

武器选择：通过点击不同的按钮来选择不同的武器。

表 3-1 数值设计

武器类型	武器名字	武器伤害	目标类型	生命值	获得积分最大值
Sweapon	Desert	10	enemy_S	50	50
Mweapon	M4A1	50	enemy_M	100	100
Bweapon	RPG	100	enemy_B	500	500

目标生成：在场景内共有八个生成点，在游戏开始时候，在场景的中心有一个游戏目标，摧毁后正式开始游戏，每一次在八个生成点中随机选取四个不相同的位置刷新游戏目标，当游戏目标全部摧毁后，重复上述生成步骤。

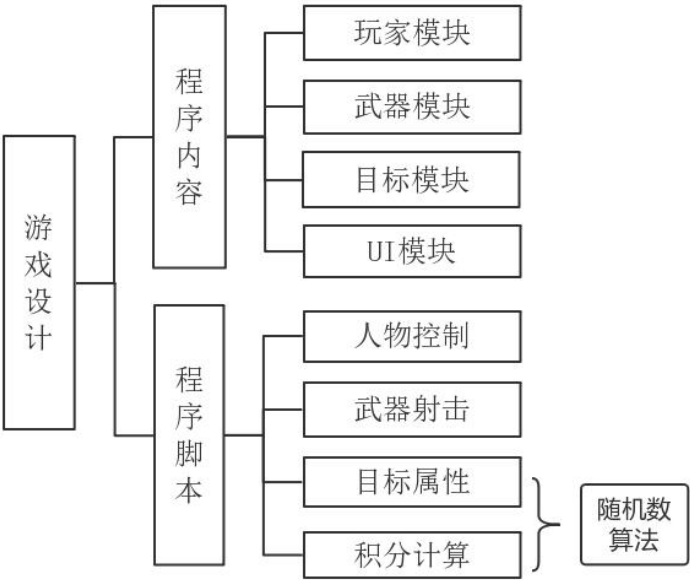


图 3-1 程序设计图

3.2 技术设计文档

游戏摘要：通过 unity 实现两项基本需求。1. C#随机算法的应用，尽可能得使用随机、复杂的随机数算法。2. 基本的 FPS 游戏要素。将随机算法与移动射击相结合建立一个手游的基本雏形。

游戏引擎：简单易上手，操作亲切化游戏制作引擎 unity，且具有优秀的跨平台能力。

技术需求：低边形建模，设置键鼠输入信息，添加物理引擎，碰撞效果判断，游戏内类型区分，随机算法应用，商业模型导入，UI 界面层级设置，SDK 接入等。

目标平台：Windows XP SP2 或以上；ANDROID OS 版本 4.1 或以上

表 3-2 文件命名规格表

资源类型	子类型	命名规范	文件类型
预制件	玩家	部位名称	FBX
	场景	场景名称	FBX
	子弹		FBX
	武器	武器类型	FBX
	目标	目标类型	FBX
脚本		管理项目	C#
材质		附着对象	.mat
纹理		附着对象	JPG
音效		附着对象	MP3
场景		场景名称	.unity

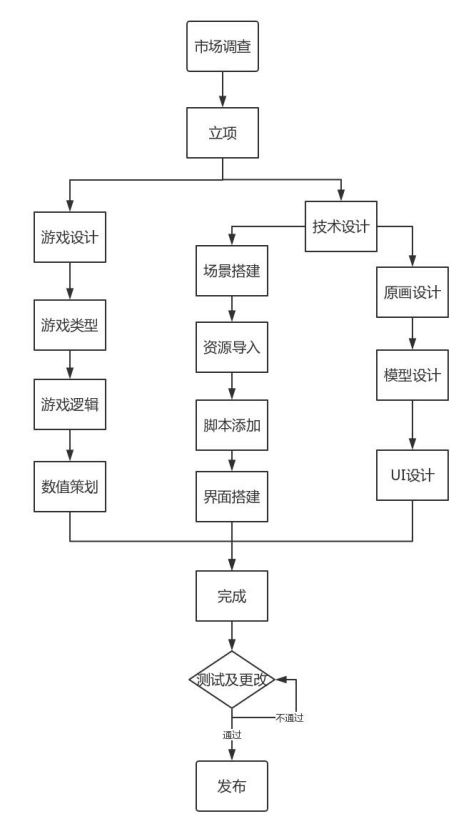


图 3-2 开发流程图

## 第四章 游戏程序实现

### 4.1 开发环境

开发引擎：2018.2.14f 个人版本

语言平台：mono

物理引擎：PhysX

PhysX, 音同 Physics, 是由 NVIDIA 提供的世界三大物理引擎之一，主要用于还原游戏中物体的运动更接近于现实中的状态。同时也应用在虚幻 3 引擎中，其参与开发的代表作有《镜之边缘》，《蝙蝠侠骑士：阿卡姆疯人院》。

### 4.2 建立场景

场景由四片墙壁和一片地面组成，提高墙壁的高度并添加刚体组件和碰撞器确保人物不会超出游戏场景。

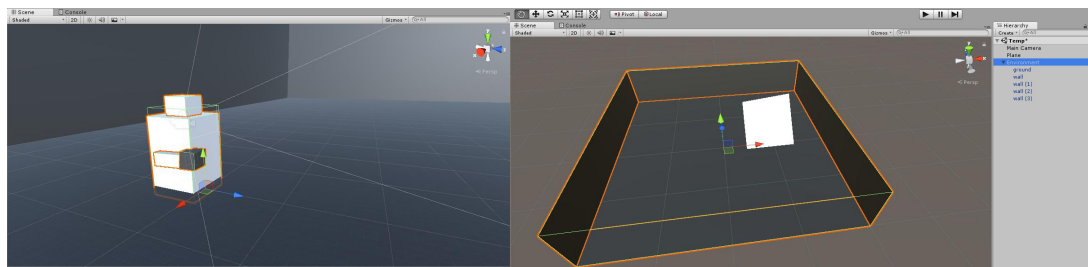


图 4-1 场景和人物开发图

### 4.3 角色添加

#### 4.3.1 建立人物模型

简单由几个立方体来组成人物模型，黑色的方块代表武器所在的位置，并给人物模型添加刚体和碰撞组件来检测是否与场景中其他物体进行物理互动。

将显示玩家视野的摄像头放置在模型头部保证第一人称视野。



### 4.3.2 添加控制脚本

创建 Movement 挂载至人物对象，获得键盘在 unity 中映射的方向轴数据。

```
v = Input.GetAxis("Vertical");//垂直方向
h = Input.GetAxis("Horizontal");//水平方向
float buffer = Time.deltaTime * speed;

//Time.deltaTime 时间间隔确保设备中数值的变化恒定
Vector3 V3_translate = new Vector3(0, 0, Mathf.Lerp(0, v, buffer));

//获得前后移动所需的向量
Vector3 V3_rotation = new Vector3(0, Mathf.Lerp(0, h, buffer), 0);

//获得旋转所需的向量
transform.Translate(V3_translate);//控制人物前后移动
transform.Rotate(V3_rotation * 10);//控制人物左右方向（四元数 Quaternion 亦可）
```

图 4-2 人物控制代码

### 4.3.3 射击功能实现

创建 shoot 脚本，并将其挂载在武器对象中。

```
if (Input.GetMouseButtonDown(0) && bulletnum != 0)
{
    //鼠标点击数据            子弹数不为空

    GameObject rt_bullet = GameObject.Instantiate
        (bullet, transform.position, transform.rotation);

    //在该脚本对象所在的位置实例化子弹预制件
    Rigidbody rg = rt_bullet.GetComponent<Rigidbody>();//获得子弹对象的刚体组件
    rg.velocity = transform.up * speed;//对子弹施加推力
    Destroy(rt_bullet, 3f);//在三秒后销毁子弹对象
    bulletnum--;//子弹数量减一
}
```

图 4-3 射击代码

### 4.3.4 实现摧毁目标

创建属性脚本，挂载至游戏目标对象中，检测到子弹撞击游戏目标后，扣除一定的血量。当游戏对象的血量低于 0 时，摧毁该目标。

### 4.3.5 跳跃功能

在正常的 3D 游戏中，人物均可进行跳跃操作。与方向移动相似，在人物控制脚本中通过检测到 Jump 输入项，人物有一个向上的力模拟跳跃。但是当人物已经跳跃至空中时，检测到没有触碰地面，无法在空中再次跳跃。

```
void OnCollisionEnter(Collision rigidbody)
{
    var name = rigidbody.collider.name;
    //Debug.Log("Name is " + name);
    if (name == "ground")
    {
        ground = true;
    }
}
```

## 4.4 其他物体

游戏目标创建：由正方体为模型建立游戏目标（玩家所需击中的目标），为了方便玩家区分标记为不同的颜色材质，并根据不同的目标属性添加相应的属性。添加碰撞器组件，检测子弹的撞击。

武器模型导入：游戏的画面是最为吸引玩家的地方，但是在开发过程中发现，游戏中物体的外观与游戏逻辑没有任何的关系，即便是最普通的小方块也可以发射火箭弹，但是为了遵循现实情况，这里将三种武器和子弹的模型进行替换。（资源来自于网络）在游戏逻辑内也根据不同种类的武器给予不同的伤害值。武器模型对应不同的物体标签，在属性脚本碰撞检测中加入对碰撞物体的判断，根据不同的伤害来源对游戏目标造成不同的伤害。

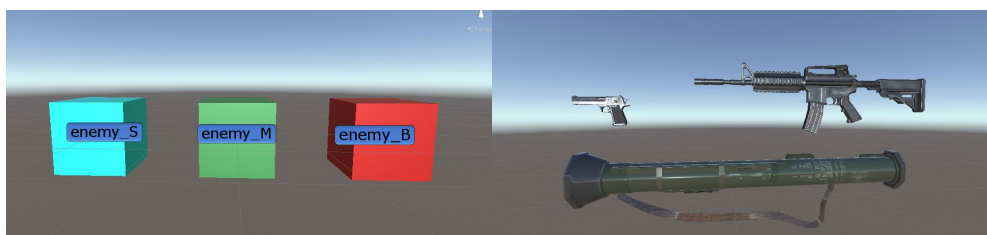


图 4-4 目标及武器模型

## 4.5 算法应用

为了游戏的趣味性，在目标属性脚本中进行随机数算法应用修改，将摧毁目标得到的积分设置为随机数，越是高 hp 的目标具有更高的积分获得上限。

修改后脚本（选用了具有代表性的 GUID 算法）：

```
void Start () {  
    gm = GameObject.Find("GameManage").GetComponent<Gamemanage>();  
    GuidNumber guidnum = new GuidNumber();  
    if (this.name == "enemy_S")  
    {  
        score = guidnum.getGnum(50);  
        hp = 50;  
    }  
    else if (this.name == "enemy_M")  
    {  
        score = guidnum.getGnum(100);  
        hp = 100;  
    }  
    else  
    {  
        score = guidnum.getGnum(500);  
        hp = 500;  
    }  
}
```

图 4-5 随机数组的应用

## 4.6 游戏逻辑

创建游戏管理器脚本，将其挂载在场景空物体内管理整个程序的运行流程，脚本逻辑为每次在所有目标点数组中随机生成四个编号，在这些编号中创建四个随机类型的目标物体，在每次检测到场上物体数量等于 0 时重复执行上述操作。



图 4-6 出生点示意图

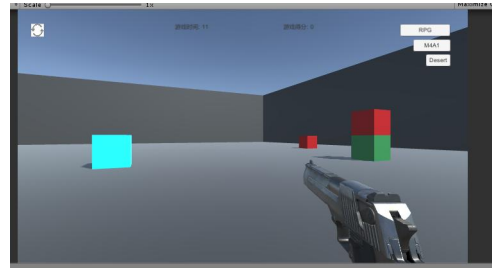


图 4-7 物体出生重复 bug

图 4-7 错误产生原因：随机数产生重复

```
public void Arepetition()
{
    EnemyCopy =
    GameObject.FindGameObjectsWithTag("Enemy");
    GuidNumber way02 = new GuidNumber();
    var EnemyType = way02.getGnum(3);
    var EnemyPoint = way02.getGnum(8);
    if (EnemyCopy.Length < Maxnum)
    {
        GameObject EnemyObj =
        GameObject.Instantiate(Enemy[EnemyType],
        EnemyPoints[EnemyPoint].position,
        EnemyPoints[EnemyPoint].rotation);
    }
}
```

无论是哪种方法数组中的数字都有可能会出现重复，因此需要在算法中通过两次循环遍历数组来去除重复数字

```
public void Nrepetition()
{
    GuidNumber way02 = new GuidNumber();
    int[] tempArray = new int[4];
    bool nrepetition;
    for (int i = 0; i < tempArray.Length;)
    {
        int tempNum = way02.getGnum(8);
        nrepetition = true;
        for (int j = 0; j < tempArray.Length; j++)
        {
            if (tempNum == tempArray[j])
            {
                nrepetition = false;
            }
        }
        if (nrepetition)
        {
            tempArray[i] = tempNum;
            var EnemyType = way02.getGnum(3);
            GameObject EnemyObj =
                GameObject.Instantiate(Enemy[EnemyType],
                EnemyPoints[tempArray[i]].position,
                EnemyPoints[tempArray[i]].rotation);
            EnemyObj.name = Enemy[EnemyType].name;
            i++;
        }
    }
}
```

## 4.7 UI 搭建

游戏内程序均已完成，添加相应的 UI 来方便玩家的操作。先在 UI 背景面板中放置所需用到的 UI 组件，三个选择武器按钮、一个重新开始按钮、游戏时间和游戏得分的文本显示框。并在相应的位置设置锚点自适应不同的屏幕分辨率。而为了避免在游戏中场景物体覆盖 UI 组件，单独创建一个 UI 摄像机来显示 UI 层面。

## 4.8 添加音效

最后在射击脚本中添加游戏的音频文件使其开火时播放相应音效。当音效添加完成之后，在 PC 平台上的开发就都已经完成了，接下来将对程序进行测试，和手机平台的移植。

表 4-1 音效资源

名称	类型	描述
Sweapon2	Sweapon	Desert 音效
Mweapon2	Mweapon	M4A1 音效
Bweapon2	Bweapon	RPG 音效

## 第五章 Android 系统平台移植

### 5.1 程序测试

在 unity 中所有需要用到的程序部分都已编写完成，最后对游戏进行运行。

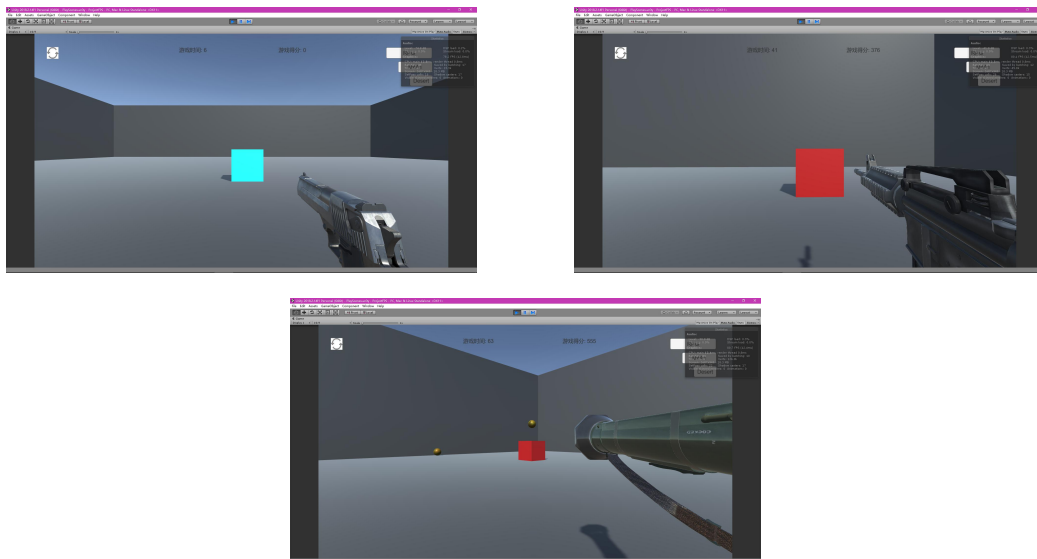


图 5-1 PC 端完成测试图

经测试，在玩家点击 UI 按键时会触发场景中物体的开火行为，在射击脚本中加入对开火行为的判断限制。

```
if (Input.GetMouseButtonDown(0) && bulletnum != 0 &&  
EventSystem.current.IsPointerOverGameObject() != true)
```

检测玩家点击时是否在游戏界面，当点击 UI 界面时，该输出为 false, 不执行开火方法。

在多次测试中，fps 均保持在 80 左右，而 drawcall 也没有超过 70，完全符合在手机运行的条件。

### 5.2 手机端移植

在完成单机 PC 的发布后，将游戏程序移植到手机端，而对于游戏内部的调整只有手机的输入方式进行修改。Unity 有着极其强大的扩展功能，在其 AssetStore 中就有着 EasyTouch 这款外部插件。对于玩家的移动，使用其虚拟摇杆组件，直接映射在 unity 的方向轴体上，改变玩家物体的移动方向和旋转角度。类似于选择武器，使用 EasyTouch 中

UI 按钮，点击模拟空格和鼠标点击，来实现手机端跳跃和射击功能。

在上述的修改完成后，运行无误后即可进行手机端的发布。得益于 Unity 强大的跨平台能力，在手机上的发布只需要输入 SDK、JDK 来获取 JAVA 和 Android 系统的运行环境，系统就可以创建应用程序，无需任何 JAVA 代码。



图 5-2 移动端完成测试

### 5.3 拓展功能

在后续的版本中拟计划加入更多元的元素。有目标模型优化、子弹物理效果增强、地形修改、动画系统、特效系统、光照系统、AI 系统。由于这些功能复杂性，可能会导致程序的异常，无法正常运行，所以目前仅完成了能够将程序整体控制在最小单位的 1.0 版本。

### 5.4 项目总结

在本次的论文研究和实例开发中，主要探讨了移动端手游的市场现状、核心随机算法和算法脚本的应用。虽然整个开发只实现了较为简单的功能，但是其中程序多次出现不符合预期的现象，且这里没有涉及到复杂的游戏机制、精良的画面甚至没有多人的联机，所以一款合格的手游是很多不同领域的人才共同合作的智慧结晶，更不要说在众多的手游产品中脱颖而出。游戏不再是曾经声名狼藉的玩物，而是有着足够受众人数、市场规模、技术需求的智慧结晶，在这个完全虚拟的世界中程序员需要用一个个字母来筑起万丈高楼，从而让更多的普通用户跨越地形障碍、跨越语言障碍来实现曾经只是童话故事中的天方夜谭。历史的车轮从未因外部的原因而停滞不前，面对任何新兴的事物，不应该抱着陈旧的思想排斥它，雏鹰必会翱翔，电子游戏的发展是可预见性的必然趋势，需要的是积极的社会认知和正确的理论指导。在新生的网络世界，有太多太多值得探索的地方，游戏更是只有不足 70 年的历史，我相信在以后，网络会更加贴近我们的生活，游戏的发展只会更加的快速、普及、智能，到了那个时候，电子游戏将为成为一种普遍的生活方式，而更多的人都会以游戏从业为荣。



## 参考文献

- [1] 邹光临.Android 与 iOS 对比分析及竞争优缺点[J].科技与创新,2017,2095-6835(2017)10:65.
- [2] 孙晓文.iOS 与 Android 操作系统的优缺点比较[J].无线互联科技.2013,1672-6944:51-69.
- [3] 涂正宏,李瑜庆.基于 Android 系统的射击游戏 App 开发与设计[J].电子世界,2018,1003-0522:123-124.
- [4] 郑铭楷,吴子朝.基于 unity3d 的射击游戏设计与实现[J].电脑迷,2018,1672-528X:29.
- [5] 乔羽茜.手机网络游戏中创新性营销策略及改进途径[J].传播与版权,2018,1674-473X:132-135.
- [6] 智能手机对人们生活的影响[EB/OL]<https://wenku.baidu.com/view/4f4bb85bbceb19e8b9f6ba7e.html>,2018-06-27.
- [7] 游戏设计师要学什么[EB/OL]<https://zhidao.baidu.com/question/1509811253685135820.html>,2018-08-21.
- [8] 全球电子游戏发展简史[EB/OL][http://www.360doc.com/content/17/0901/10/8250148\\_683779152.shtml](http://www.360doc.com/content/17/0901/10/8250148_683779152.shtml),2017-09-01
- [9] 用 Unity3d 开发游戏的优点都有哪些? [EB/OL]<https://blog.csdn.net/JtNbCOC8N2I9/article/details/78557144>,2017-11-14
- [10] C#中生成随机数的几种方法[EB/OL]<https://www.cnblogs.com/jackcheblog/p/7417632.html>,2017-08-23

## 致 谢

首先感谢我的指导老师华中伟对我的悉心指导，在撰写论文的过程中，给予了多方位的指导，从论文的选题方向到论文的整体结构和后来的论文修改，华老师都投入了非常多的精力认真审阅。

感谢我的班主任老师胥薇教给我们丰富的软件开发现代化思想。同时胥老师积极的督促我们完成学校相关的流程，合理的安排论文日程，能够在美好的大学生活不留遗憾。

感谢年纪主任徐云龙老师，在百忙的行政安排中仍然抽取时间走进教室、走进学生集体，亲自教授计算机领域的专业知识，并结合项目经验传授在工作中的会用到的实用技巧，为今后学习、工作道路上点亮一盏明灯。

最后感谢所有关心和帮助过我的老师与同学们。