南 阳 理 工 学 院

本科生毕业设计(论文)

学院(系)： 软件学院

专 业： 软件工程

学 生： 周亮

指导教师： 王超

完成日期 2015 年 04 月

南阳理工学院本科生毕业设计（论文）

太空大战游戏的设计与实现

Design and Implementation of the  
Space War Game

总 计：毕业设计(论文) 5页

表 格： 0个

插 图： 0幅

南 阳 理 工 学 院 本 科 毕 业 设 计(论文)

太空大战游戏的设计与实现

Design and Implementation of the  
Space War Game

学 院(系)： 软件学院

专 业： 软件工程

学 生 姓 名： 周亮

学 号： 1115115208

指导教师(职称)： 王超 讲师

评 阅 教 师：

完 成 日 期： 2015年04月15日

南阳理工学院

Nanyang Institute of Technology

太空大战游戏的设计与实现

软件工程 周亮

1. 随着电脑的普及，游戏得以其高速的发展，游戏也成了大部分人的娱乐方式之一。游戏是以手机和电脑为平台，通过人机互动的方式，将情感互动、故事铺展、视觉表现、用户界面、艺术风格融合在一起的一种大众娱乐方式。它综合运用了计算机图形学技术、人工智能技术、多媒体技术和网络与通信等技术。游戏的发展从设计思想上可以划分为传统游戏、网络游戏和虚拟现实游戏，从技术上又经历了2D和3D两个阶段。传统游戏从最初简单的电视互动游戏开始，经过了电子游戏、益智游戏、电脑单机游戏等发展过程。玩家通过自己的参与到游戏之中，全面的体验游戏带来的快乐。
2. 太空大战；游戏；Unity3D

Design and Implementation of the  
Space War Game

Software Engineering Major Zhou Liang

1. With the popularity of computer, the game to its high-speed development, the game has become one of the most forms of entertainment.Game on mobile phones and computers for platform, through the way of human-computer interaction, emotional interaction, the story spread out, visual representation, user interface, artistic style in a public entertainment.It integrated use of the computer graphics technology, artificial intelligence technology, multimedia technology and network and communication technology, etc.The development of the game from the design thought can be divided into traditional games, online games and virtual reality games, from the technology and experience of 2 d and 3 d two phase.Traditional games from the original simple television interactive games began, after the video games, puzzle games, such as computer LAN game development process.Players through their participation in the game, the experience of full enjoyment of the game.
2. Space War ; Game ; Unity3D

目 录

[1 绪论 1](#_Toc416646836)

1.1 引言 1

1.2 目的与意义 1

1.3 相关知识概述 1

1.3.1 Unity3D 1

[2 XXXX 1](#_Toc416646837)

[结束语 2](#_Toc416646838)

[参考文献 3](#_Toc416646839)

[附录 4](#_Toc416646840)

[致谢 5](#_Toc416646841)

# 绪论

## 引言

自从我们进入信息时代以来，计算机以及相关的产业和技术得以飞速发展，游戏以这些为载体也在不断更新换代，先是经历了红白机时代，这时候的游戏只能呈现简单的方块线条，而且还是单色的。改变历史的是日本任天堂（Nintendo），在上世纪八十年代，日本的任天堂开辟了一个崭新的红白机时代，这时，真正意义上的家用游戏机FC开始风靡世界，并逐步产生了一个庞大的家用游戏机产业。随后是16位主机时代，16位主机时代是以日本世嘉（SEGA）1988年推出的16位游戏机MD（Mega Driver）为起点的。对于世嘉而言，美国市场才是真正的战略重心。1989年9月15日该主机在全美上市，售价为190美元。由于当时美国游戏市场已经基本被任天堂垄断，世嘉决定开拓较高年龄层的新玩家群体，推出了大批面向成年玩家群体的游戏。到了上个世纪九十年代，电脑游戏业才刚开始变得成熟。随着电脑的普及，开始了它疯狂的生长。单机到网游、2D到3D以及现在的手游Xbox，游戏已发展成人们生活的主要娱乐之一。

## 目的与意义

太空大战游戏设计的初衷就是为了让玩家用户在紧张劳累之余放松娱乐的。玩家用户通过与游戏系统的直接交互是通过界面系统完成的，所以界面设计应该是游戏设计中非常重要的一个环节。游戏的界面跟产品的外观和功能一样，要能吸引玩家并且容易上手。在设计界面的过程中，要一直注重易用性设计原则，并且充分考虑玩家的感受，随时调整和修改界面的设计细节。

游戏界面既具有界面设计的一般规律，更具有游戏这一特殊开领域的个性设计原则。在界面的设计制作过程中，应在掌握基本游戏框架设定能力和对用户需求分析能力的基础上，了解并掌握游戏开发机制尤其是游戏界面的实现原理以及设计中常用的技术，例如：界面风格制定、美术需求文档制作、界面原型制作、程序坐标图制作等技术。

## 相关知识概述

### Unity3D 概述

Unity是由Unity Technologies开发的一个让玩家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。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。也可以利用Unity web player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。Unity3D原来是一个针对Mac、Windows和Linux的游戏开发套件，后来发展到能够在iPhone和Wii上部署，或者在Web和Android系统上部署。Unity使用了Mono作为脚本引擎的虚拟机，并以C#或者一种类似JavaScript的语言为脚本语言。

Unity有着简单的界面，友好的开发环境及对当前流行游戏平台的支持，它对于初学者或是那些希望使用一个简单、高效并且友好的游戏引擎创建游戏的人来说是完美的。Unity可部署在Mac OS或者Windows操作系统中,在这2个操作系统中，除了客户端操作习惯与界面有差异外，引擎自身的功能没有任何区别。

2004年，英特尔、波音、Adobe、微软以及30余家公司在西班牙巴塞罗那共同宣布合作开发一个名为U3D的通用3D图形格式标准。旨在让用户无须专用软件，能够像MP3用于视频和JPEG用于静态图像那样，可以免费获得的3D数据编码方式。而且传统的三维文件就是基于CS结构，本机系统，脱离源程序无法打开和浏览，即使通过特定程序能够打开，也是以牺牲大量编辑修改功能作为代价，尤其是三维文件一般都体积巨大，无法通过互联网传播、浏览！U3D的目标是结束诸如AutoDesk、Catia、UG、Pro-E和其它软件公司开发的专有的3D图形格式，用所有的用户都可以使用的一种标准格式来取代它。在保留绝大多数功能的前提，无须专有程序即可打开，或在互联网进行3D视图的浏览和操作。换言之，U3D就是让3D文件能够象JPEG文件一样流行和易于使用！这种标准将使3D更容易结合到网络浏览器、手机浏览器等其它应用程序中，使3D图像能够更广泛地普及，从而促进对更快的处理器和图形芯片的需求。2005年，Adobe在业界率先发布了Acrobat 3D，即在Acrobat现有的基础上加入对3D物件的支持。从 Adobe Acrobat 7.0开始，Adobe的PDF格式支持在文档中嵌入3D内容。这些3D内容以“Universal 3D” (U3D)格式保存在PDF文件中。

### NGUI 概述

在Unity 3D中，可以使用代码控制其自身所携带的GUI来实现图形界面的搭建。但是，这些组件的搭建效率较低，而且已经无法满足现在市场对图形用户界面美感的要求，所以，引入了NGUI来增加所要开发的图形用户界面的美感。

NGUI是严格遵循KISS原则并用C#编写的Unity（适用于专业版和免费版）插件，提供强大的UI系统和事件通知框架。其代码简洁，多数类少于200行代码。这意味着程序员可以很容易地扩展NGUI的功能或调节已有功能。对所有其他用户而言，这意味着更高的性能、更低的学习难度和更加有趣。

### 界面的设计

游戏UI界面设计在一款游戏中是起着至关重要的作用。有一个良好的游戏界面，能够帮助玩家快速上手，能让玩家能够更加的喜欢这个游戏，不管游戏设计创意如何，如果游戏界面设计差，玩家对游戏的第一印象就产生了抵制心理。玩家会感觉游戏界面做的不好，游戏内容应该也不会好到哪里去的。所以游戏设计的第一个关卡就是要有良好的游戏交互界面，有良好的用户体验。如果这个条件达不到那么这个游戏应该已经算是失败了。玩家是不会去玩UI界面一塌糊涂的游戏的。UI界面包括游戏主界面、二级界面、弹出界面、地图界面等很多种类。Unity3D有很好的插件NGUI以及4.6之后的版本自带的UGUI，Uniyt3D本身GUI 系统一般用的比较少，Unity3Dd 动画系统还是能满足一般的动画需求的，其他复杂的动画可能就需要用到Maya或者3DMax软件了，总的来说Unity3D提供了相对完善的功能，因此受到了开发者的一致好评。可以实现场景切换及动画效果，从而快速的开发出游戏雏形，缩短游戏开发周期。本游戏利用NGUI游戏插件完成UI相关的交互展示设计。使用TexturePacker进行图片拼接，减轻游戏渲染负担。使用Ps等图片软件进行图片处理加工。将游戏界面设计的简洁友好。

### 功能的设计

本游戏为太空大战射击类游戏，所以游戏中有敌人类的功能、有积分类的功能，有关卡闯关类的功能。有背景音乐和音效设置的功能。

其中地人类的功能包括：敌机、敌方发出的子弹导弹等、关卡障碍物。

积分类的功能包括：击杀敌机奖励积分、获得道具奖励积分、积分排行榜。

关卡闯关类的功能包括：关卡最后设有BOSS用来卡住难度、小关卡大关卡BOSS难度不同、关卡数。

背景音乐和音效设置功能：设置背景音乐的播放、游戏关于、背景故事介绍。

### 实现系统的步骤

A. 进行游戏模块系统分析，资源文件制作。

B. 游戏开发环境搭建：这里的开发环境为Windows环境、Android环境和IOS环境。

C. 游戏项目创建，创建好项目后开始导入插件以及需要的资源文件。

D. 开始项目编写，用TexturePacker制作Atlas 然后使用NGUI插件编辑界面。使用MonoBehavior进行代码编辑与调试。

E. 测试进行各个版本的打包和上传。

# 系统开发环境

## 编程环境的选择

### 操作系统的选择

目前可供选择的操作系统有很多，有Windows、Linux和Unix等。Windows具有界面图形化、多用户、多任务、网络支持良好、出色的多媒体功能、硬件支持良好和众多的应用程序等特点。Linux是一种类Unix系统，是一个免费的操作系统，具有许多Unix系统的功能和特点，能够兼容Unix，但它没有Windows操作系统应用广泛，其上的应用软件种类较Windows少了很多，Windows上的很多应用软件在该系统上通常无法正常运行。而且许多硬件设备面对Linux的驱动程序也很不足。Unix适用于一些大型的网络。Unix与Linux这两种操作系统与Windows操作系统相比是比较复杂的, 而且Linux在技术上还不很成熟。经过多番比较，为了更好的开发和操作本系统，决定采用以下系统软件：

操作系统：microsoft Windows 7 Ultimate

### 开发环境的选择

游戏开发环境有很多，最近这几年比较常用的有Cocos2dx、Unity3D 等等。本游戏选择的开发环境是Unity3D，之所以使用U3D是因为它对网页游戏，手机游戏，对于各个平台支持都很好。并且支持flash，网页运行再也不用安装浏览器插件。U3D很容易上手，开发人员起点比较低。基本的资料文档十分丰富，很多官网论坛可以互相交流经验，有利于快速开发与提高。U3D支持的平台，无疑是当下较为流行的平台。满足绝大部分项目需求。早期的引擎，多以PC和CONSOLE为主。支持WINDOWS,XBOX,PS2已经是很不错了。U3D便利的多平台发布特性，也使得它成为了当前性价比最高的引擎的原因之一。也有许多公司正在自主研发引擎，或者是将先前的PC引擎修改为多平台（IOS+ANDROID居多）。

### 编程语言的选择

编程就是让计算机为解决某个问题而使用某种程序设计语言编写程序代码，并最终得到结果的过程。为了使计算机能够理解人的意图，人类就必须要将需解决的问题的思路、方法、和手段通过计算机能够理解的形式告诉计算机，使得计算机能够根据人的指令一步一步去工作，完成某种特定的任务。这种人和计算机之间交流的过程就是编程。 从计算机发明至今，随着计算机硬件和软件技术的发展，计算机的编程语言经历了机器语言、汇编语言、面向过程的程序设计语言以及面向对象的程序设计语言阶段。具体的语言又是不胜枚举。因此，对于一个以编程为职业的人来说，了解程序的发展及文化，是非常有必要的。

C#是微软公司发布的一种面向对象的、运行于.NET Framework之上的高级程序设计语言。并定于在微软职业开发者论坛(PDC)上登台亮相。C#是微软公司研究员Anders Hejlsberg的最新成果。C#看起来与Java有着惊人的相似；它包括了诸如单一继承、接口、与Java几乎同样的语法和编译成中间代码再运行的过程。但是C#与Java有着明显的不同，它借鉴了Delphi的一个特点，与COM（组件对象模型）是直接集成的，而且它是微软公司 .NET windows网络框架的主角。

C#是一种安全的、稳定的、简单的、优雅的，由C和C++衍生出来的面向对象的编程语言。它在继承C和C++强大功能的同时去掉了一些它们的复杂特性（例如没有宏以及不允许多重继承）。C#综合了VB简单的可视化操作和C++的高运行效率，以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向组件编程的支持成为.NET开发的首选语言。[1]

C#是面向对象的编程语言。它使得程序员可以快速地编写各种基于MICROSOFT .NET平台的应用程序，MICROSOFT .NET提供了一系列的工具和服务来最大程度地开发利用计算与通讯领域。

C#使得C++程序员可以高效的开发程序，且因可调用由 C/C++ 编写的本机原生函数，因此绝不损失C/C++原有的强大的功能。因为这种继承关系，C#与C/C++具有极大的相似性，熟悉类似语言的开发者可以很快的转向C#。[2]

C#旨在设计成为一种“简单、现代、通用”，以及面向对象的程序设计语言，此种语言的实现，应提供对于以下软件工程要素的支持：强类型检查、数组维度检查、未初始化的变量引用检测、自动垃圾收集（Garbage Collection，指一种自动内存释放技术）。软件必须做到强大、持久，并具有较强的编程生产力。此种语言为在分布式环境中的开发提供适用的组件开发应用。

为使程序员容易迁移到这种语言，源代码的可移植性十分重要，尤其是对于那些已熟悉C和C++的程序员而言。对国际化的支持非常重要。C#适合为独立和嵌入式的系统编写程序，从使用复杂操作系统的大型系统到特定应用的小型系统均适用。[3-4]

# 游戏需求分析

《太空大战游戏》是一款太空射击类游戏，为了保证开发完成之后该游戏能以其可玩性、画面效果等吸引玩家，提高游戏的下载量，增加玩家的时间投入和金钱投入，,所以要求该游戏一定要画面精美、可玩度高、操作流畅、用户体验良好。

## 可行性分析

可行性分析是对项目的可行程度进行分析，其主要包括经济及市场可行性分析，技术可行性分析和法律可行性分析[4]。通过对需求分析的研究，本游戏项目的可行性分析如下：

经济和市场可行性分析：随着科技的发展，现在智能手机的功能也越来越多，越来越强大，越来越智能，给人们带来了很大的方便，让人对智能机产生了相应的依赖类。而手机游戏也远远不是我们印象中的什么“俄罗斯方块”、“贪吃蛇”之类画面简陋，规则简单、交互性不强和用户体验不好游戏了，进而发展到了可以和掌上游戏机媲美，具有很强的娱乐性和交互性的复杂形态了。而使用手机玩游戏的比例逐渐变大，这成为人们生活的常态。因为手持设备携带方便、功能也非常强大。所以手机游戏的市场是很大的，具有很好的可行性。

技术可行性分析：开发一款跨平台射击游戏（太空大战），在如今开发软件上是比较容易实现的，本游戏项目的规模不大，实现难度不是很高。Unity3D经过这几年快速的发展，如今已经趋于成熟，各种插件工具也非常齐全。Unity3D引擎在国内手机游戏市场占有率达75%，对各个平台的支持良好。故此，本游戏具有技术上的可行性。

法律可行性分析：本游戏面向的对象是游戏爱好者用户，用途主要用来单机运行，不存在可以暴露个人数据的外部网络接口，且用户具有游戏全部的权限，是在独立自主的情况下试玩本游戏。而且本游戏不会给社会带来危害和隐患。故此，现游戏项目具有法律上的可行性。

## 系统的功能性需求

功能需求是游戏软件需求中最常见和最主要的需求，不同层次的人群可以观察到功能需求不同程度的抽象，即功能需求具有层次性。功能需求通常体现为以下三个层次：

业务需求（Business Requirement）：表示组织或客户高层次的目标。它描述了组织为什么要开发游戏软件，即组织者希望达到的目标。对于本游戏软件来说，业务需求为实现一个塔防类游戏。

用户需求（User Requirement）：就是执行实际工作的玩家们对游戏功能具体任务的期望。业务需求一般是由组织的专门部门提出，但普通用户才是组织中任务的实际执行者，只有通过具体而且合理的业务流程才能真正地实现目标。也就是说用户需求描述了用户能使用系统来做些什么。就本游戏而言，首先将功能做好放到游戏市场中供玩家们体验，在游戏设定邮箱等联系方式，玩家对游戏的建议或是bug都能够及时的反馈给我们。通过不断收集玩家的需求来不断改善游戏。做到玩家们喜爱并且实用性强的一款游戏。

系统需求（System Requirement）：规定开发人员必须在产品中实现的系统功能，用户利用这些功能来实现完成任务，满足业务需求，系统需求描述的是开发人员需要实现什么。就本游戏来说，主要就是实现玩家数据保存、游戏运行顺畅、用户体验强。要给玩家一种休闲畅快的体验。

### 游戏系统模块

根据前面各方面可行性的分析，下面进行游戏功能模块系统的全面分析。分析游戏中拥有的功能模块。本游戏的主要系统功能有：战斗系统、地图系统、玩家信息系统、游戏的开始退出功能、文本系统、音频系统、数据系统和资源加载系统八大系统。如图2-1给出详细的系统共之间的相互调用功能和游戏的系统之间的所有逻辑联系，并且也能表达游戏的整个框架。

### 主要游戏系统模块说明

#### 资源加载

该功能主要实现将游戏中用到的所有图片资源预先加载到内存中，这样在游戏中使用该资源时能够更加快捷的加载资源，保证游戏在运行时顺畅。能够保证玩家在游戏中体验。

#### 战斗系统

该系统实现玩家在某一关卡中进行战斗时，玩家通过己方战斗机发射子弹进行进攻，玩家通过消灭敌机获得积分，玩家消灭的敌机越多且漏掉的敌人越少，获得的积分就越高，收到攻击时己方战斗机的血值会减少，当减少到零时战斗结束，该关卡失败。

#### 地图系统

该系统实现是游戏地图的创建和关卡地图的创建。由于太空大战游戏地图与游戏并没有直接关联，只是用来显示关卡不同，所以这里采用动画模式播放各个关卡地图，地图多选用星空作为背景结合NGUI显示每个关卡将显示本关卡的基本信息。

#### 玩家信息系统

该系统主要是保存游戏中玩家的积分排行信息，游戏中的关卡信息。关卡的状态信息等。开启到哪一关卡，这样有利于游戏中数据的读取，和操作。

#### 文本系统

系统主要是实现游戏使用的中文对照表，将游戏中所有的中文保存在文件中（这里采用Unity中提供了一个用于本地持久化保存于读取数据的类——PlayerPrefs。 它是以键值对的形式将数据保存在文件中），在开始游戏之前将该信息读取到游戏中进行保存。当游戏中要使用某中文是，直接通过key来引用该中文，这样做能够很好的解决跨平台中文的支持，也能够实现多语言化，后期要想发布其他语言的版本直接修改该对应的文件中的内容，在游戏中只要该key值不变就能够正常的获取到该文本信息。

#### 音频系统

该系统实现游戏中背景音乐的播放，战斗音乐播放，音效的播放功能。游戏中使用直接访问该对象中的静态方法，传递相应的标识符就能给播放相应的背景音乐或是音效。这样做的好处是，将音频系统独立出来，降低与游戏系统本身的耦合度，使游戏系统不会臃肿，这样做的的好处还有就是当我们要增加音效时，直接在该系统中增加即可，在游戏中使用直接引用该音效的ID标识符来播放。能够使游戏系统更加的易维护，易后期开发。

#### 数据系统

该系统实现的功能是对游戏中玩家的数据、关卡数据等信息的保存和读取，一个游戏不能每次进入游戏都是一个全新的游戏，游戏关卡都是重新开始，这样会然玩家对本游戏失去玩下去的信息。这样做用户体验也不好，所以我们要将玩家的数据进行保存。

### 游戏主要系统模块用例图

本游戏面向的对象只有一个那就是游戏玩家，但是系统本身也算是一个使用对象。本游戏的功能模块就分发给了游戏玩家和系统两个用户。他们两者之间相互作用，通过玩家的操作和系统事物的处理，使游戏更加完美的呈现。游戏系统对游戏中文本系统的加载，战斗系统的判断，音效系统的管理等等操作来管理游戏，协助游戏玩家更好的操作本游戏。

#### 游戏玩家

基于本游戏的设计，游戏玩家是本游戏中核心的用例，拥有对游戏操作的最高权限。游戏玩家对游戏关卡的攻打，游戏关卡分数的高低取决于玩家通关关卡的多少漏掉敌机的多少。本游戏中游戏玩家拥有的功能有：关卡选择，音效开关控制，游戏开始和退出，如图3-2 1玩家用例图

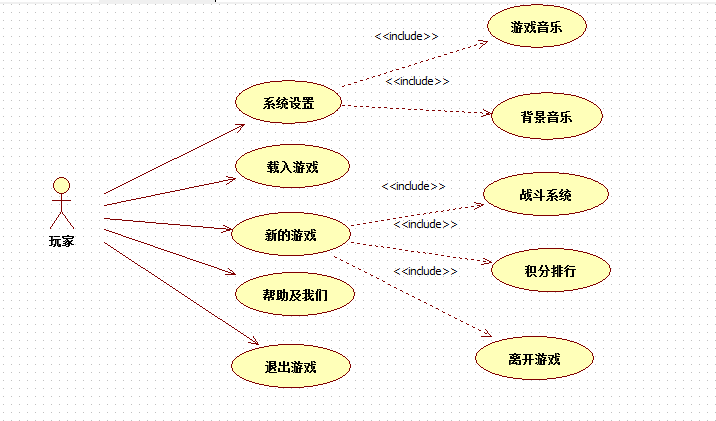


图3-2 1

#### 系统

基于本游戏的设计，游戏中的某部分功能玩家不肯拥有他，所以这里增加了一个用例就是系统，将除了玩家的功能外，将游戏本身其他的系统分发给系统用例。系统主要的功能有：资源加载（包括音频资源、图片资源、模型场景资源）、敌机进攻、生成障碍物、游戏输赢判断、积分排行榜管理、音频系统管理和数据存取。如图3-2 2系统用例图。

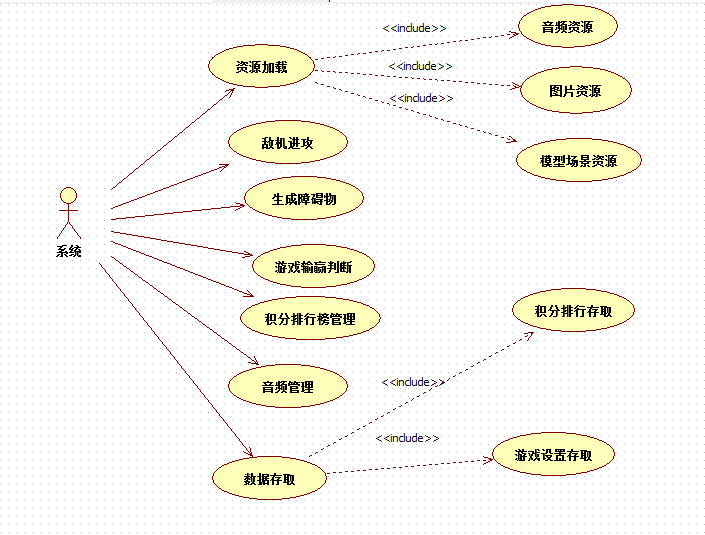


图3-2 2

### 游戏主要用例描述

下面给出用例图中的重要用例的描述，使其能够了解其中的功能和运行机制。详细见表3-1到3-6：

表3-1资源加载用例描述

|  |  |
| --- | --- |
| ID： | 1 |
| 名称： | 资源加载 |
| 参与者： | 系统 |
| 前置条件 | 游戏程序运行 |
| 后置条件 | 资源文件加载成功 |
| 操作流： | 1. 游戏程序运行 2. 将游戏模型资源以及总的图片资源和音乐音效文件加载 3. 加载成功进入游戏开始界面 |

表3-2敌机进攻系统用例描述

|  |  |
| --- | --- |
| ID： | 2 |
| 名称： | 敌机进攻系统 |
| 参与者： | 系统 |
| 前置条件 | 玩家运行游戏程序 |
| 后置条件 | 游戏登录成功进入战斗，能够通过key值获取配置及玩家信息 |
| 操作流： | 1. 游戏程序运行 2. 游戏加载 3. 读取游戏配置 |

表2-3生成障碍物系统管理用例描述

|  |  |
| --- | --- |
| ID： | 3 |
| 名称： | 生成障碍物系统管理 |
| 参与者： | 系统 |
| 前置条件 | 玩家运行游戏程序 |
| 后置条件 | 能够在程序中随机生成大小各异的障碍物 |
| 操作流： | 1. 游戏程序运行 2. 资源加载成功 3. 随机生成障碍物 |

表2-4游戏输赢判断管理用例描述

|  |  |
| --- | --- |
| ID： | 4 |
| 名称： | 游戏输赢判断管理 |
| 参与者： | 系统 |
| 前置条件 | 游戏运行 |
| 后置条件 | 游戏进行中玩家收到攻击后生命值是否为零 |
| 操作流： | 1. 游戏程序运行 2. 玩家每次收到攻击系统进行记录判断生命值 3. 玩家赢了进入下一关、输了显示操作界面 |

表2-5积分排行榜管理用例描述

|  |  |
| --- | --- |
| ID： | 5 |
| 名称： | 积分排行榜管理 |
| 参与者： | 系统 |
| 前置条件 | 游戏战斗结束 |
| 后置条件 | 玩家结束本次战斗后且本次所得积分比已有排行榜的最后一名分数高 |
| 操作流： | 1. 玩家开始游戏进入游戏战斗场景 2. 在收到多次攻击生命值为零结束本次战斗 3. 所得积分会与已有排行榜的积分进行比较得出是否进入排名 4. 排名刷新 |

表2-6音频系统管理用例描述

|  |  |
| --- | --- |
| ID： | 6 |
| 名称： | 音频系统管理 |
| 参与者： | 系统 |
| 前置条件 | 玩家运行游戏程序 |
| 后置条件 | 能够在程序中播放音乐或音效 |
| 操作流： | 1. 游戏程序运行 2. 资源加载成功 3. 通过key指定播放对应的音乐文件或者音效文件 4. 在游戏中成功播放音乐或者音效 |

### 系统的非功能性需求

#### 游戏运行环境需求

本有游戏支持跨平台运行（web、win32、IOS、Android），由于运行环境的不同，所以游戏对于不同平台有对象的安装包，在IOS设备中使用编译打包得到的ipa包进行安装运行，Android中使用编译打包后得到的apk文件进行安装，在Windows系统中直接双击exe程序进行运行，网页中需要安装Unity Web Player插件，这样就能够成功得在设备中运行游戏。由于游戏自身消耗比较大，建议玩家的设备性能高一点比较合适玩本游戏，设备性能偏低对体验本游戏会出现卡顿之类的现象。

#### 游戏易用性需求

游戏采用玩家同游戏即时交互，通过玩家的点击操作，反馈给玩家相应的内容。界面简洁明了，位置、大小适当。事前状态的指引反馈为玩家的下一步操作提供参考线索，以便做好准备，当下一次出现同样的反馈提示时，玩家便知道该作何反应或进行何种操作。玩家操作反馈，给玩家提供视觉、听觉或者触觉上的反馈，这样玩家便能清楚地知道自己当前的操作会有什么效果，从而更好地投入到游戏中去，投入其中才更能感受到游戏的乐趣，所以为玩家的操作提供感官反馈尤为重要。玩家完成某个动作之后需要通过反馈知道这个动作能达到什么效果，能获得什么回报。游戏吸引人的一点就在于，在其中的行为收益是即时性的，所以让玩家能够直观地感受到自己获得的收益很重要，这就需要反馈来做到，并且关于收益的反馈是需要强化的。

#### 游戏安全性需求

游戏安全是一个游戏核心的问题，也是一个比较棘手的问题，本游戏战士不存在网络数据的安全，因为本游戏属于单机游戏，占体不提供网络这块的功能，所有数据都是本地化存储，要保证游戏数据的安全，保证本地数据不会被恶意删除或是串改，使用一定的加密算法来对游戏数据进行加密处理，保证游戏数据的安全性，防止游戏数据被恶意串改。

# 游戏设计

做完游戏的全方面需求分析，确定游戏的开发平台，编程语言，游戏引擎和数据库的等信息后，下面就开始进入游戏最重要的环节，游戏设计阶段。游戏系统设计分为游戏总体设计和游戏详细设计。游戏总体设计的主要任务是从宏观上对该游戏进行分析工作，包括游戏框架设计和函数接口设计，确定所有需要使用到的结构；游戏详细设计的任务是确定各个功能模块的数据结构和操作，即从微观上对各个游戏功能模块通过编码实现。下面分别来结束这两种设计的详细内容。

## 游戏系统设计概要

通过上章节对本游戏的分析，本游戏使用Unity3D引擎作为游戏的主题框架，使用以前提供的各个功能模块来实现本游戏中的所有游戏功能模块。本游戏框架分为4层，第一层是我们操作系统，游戏依赖最底层的，因为本游戏平台所以这里的最底层依赖平台系统。第二层为我们的引擎层，Unity3D引擎帮我们封装好了所有游戏的单元元素（例如Sprite精灵直接使用Sprite来创建一张图片，并且显示在屏幕中）游戏中我们只要对其的成员变量等的操作。第三层是本游戏中的功能模块，第四层是游戏的表现层，就是游戏的界面层，玩家能够直接看见并且能够操作的层。下面给出这四层的层次图，能够清晰的了解本游戏中的所有功能，也是本游戏的功能模块。本游戏总体分为一下模块：数据管理模块，处理游戏中的所有数据，数据保存，数据读取操作；游戏显示模块该模块处理游戏中所有的界面功能；游戏控制模块该模块处理游戏的逻辑部分，通过该模块实现游戏显示模块和数据管理模块的联系。游戏模块如图4 1游戏功能模块图。

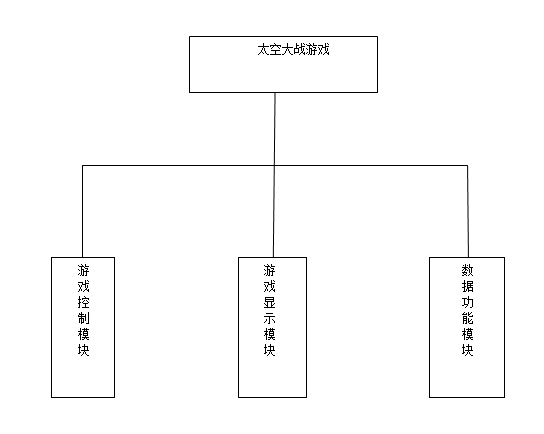


图4 1游戏功能模块图

## 游戏总体设计

### 游戏主要流程图

如图4 2-1游戏流程图 本游戏的执行流程图，能够清楚的表现出游戏中各个界面功能的跳转。游戏中玩家操作相应功能后跳转的界面提示。体现整个游戏的流程框架，本游戏操作简单，复杂度不高，对于有一定玩游戏经验的玩家来说是操作很容易，很容易上手，而对于不熟悉该类型游戏的游戏玩家来说，游戏中有详细的提示功能，玩家一看就一目了然，也能够很快的上手试玩本游戏。

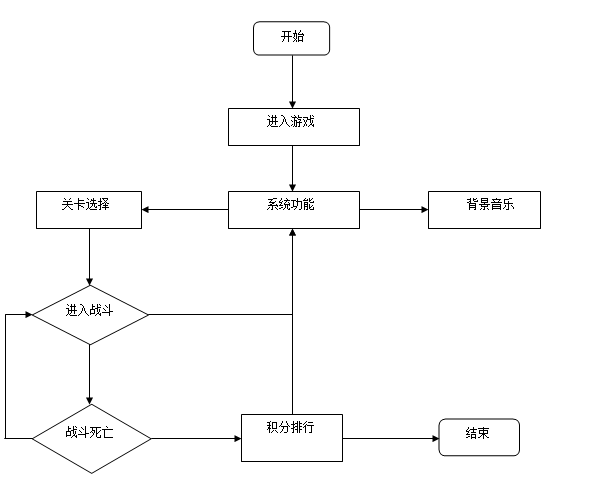


图4 2-1游戏流程图

### 游戏界面设计

本游戏中的界面友好、简洁、美观、大方，主要的游戏界面有以下界面：加载界面、游戏设置界面、游戏关于界面、关卡选择界面、游戏战斗主界面和胜利或失败弹出界面。下面详细介绍每个界面的功能和作用。

#### 加载界面

该界面实现的功能是加载游戏资源数据，将游戏中用到的图片资源、音频资源、模型资源进行加载，界面中有一个进度条，玩家能够看见游戏资源加载的进度，在该界面中我们将游戏中的数据管理模块进行实例化，该模块是一个单例，将游戏管理模块中的所有数据进行初始化。当进度条显示100%时，游戏资源加载完毕和游戏初始化完毕，并且自动跳转到菜单选择界面中。加载界面流程图 如 图4 2-2加载流程图。

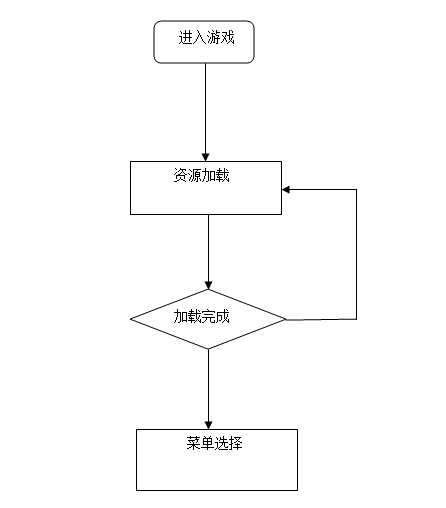


图4 2-2加载流程图

#### 设置界面

在设置界面中玩家可以在这里进行背景音乐的屏蔽和游戏音效的屏蔽。当然后也可以在该界面中恢复音效和背景音乐。点击对象的按钮进行对象的操作，该界面退出返回到游戏菜单界面，同样可以使用设备返回键进行返回。

#### 游戏关于界面

在该界面中主要实现的功能是显示游戏相关介绍，游戏的玩法提示，该界面中的内容是变动的，具体想要显示的内容在文本系统中进行配置，也就是在text表中进行配置，然后在程序中进行显示。

#### 关卡选择界面

该界面主要是显示游戏关卡信息，本游戏总共12关卡，每个关卡对应不同的按钮，点击相应的按钮实现不同关卡战斗，战斗信息保存在游戏数据中，交由数据管理系统控制，开启的关卡显示游戏关卡信息，包括分数、星级，没有开启的关卡显示封闭，并且不能进行操作，只有前一个关卡通关后才能开启。点击相应的关卡进入游戏战斗界面。

#### 游戏战斗主界面

该界面主要显示战斗画面，玩家通过消费金币来创建防御塔来阻止亲亲(敌人)的进攻，当亲亲（敌人）攻入么么房间时，么么受到伤害。该界面显示游戏的关卡数据包括：亲亲（敌人）回合，当前金币。当玩家阻止亲亲进攻波数完毕后，取得游戏胜利，相反在亲亲攻击的波数以后导致么么血量为零，游戏失败显示失败界面结束游戏。

#### 胜利或失败界面

这两个界面是在游戏结束后对应显示的提示框，胜利显示胜利界面，并且显示游戏的分数，星级，显示胜利的详细特效和音效，失败后显示失败按钮，也显示游戏玩家本关卡的游戏数据，但是该数据不进行保存，显示失败特效和音效来表示失败。

### 游戏功能模块设计

太空大战游戏中的主要功能模块有逻辑处理模块、数据管理模块、地图处理模块、碰撞检测模块和图片及动画管理模块。使用Unity3D游戏引擎来对这里模块进行管理。

#### 逻辑处理模块

逻辑处理模块主要实现程序各个模块的整合以及界面之间的切换。在本游戏中，涉及的界面不是很多，直接使用Unity3D的场景加载来进行游戏场景及界面切换，Unity3D场景切换基本操作有Application.LoadLevel("场景名称")切换指定场景。除了场景切换还有常用的界面切换用到实现已有的界面进行显示和隐藏来达到界面跳转切换 gameObject.SetActive(false / true)。

#### 数据管理模块

本游戏涉及的游戏数据不多，主要数据包括游戏关卡数据，游戏设置数据。这些数据主要分为会改变数据和固定数据，根据游戏操作需求将这些数据以键值对形式存入Unity3D自带的PlayerPrefs类中。PlayerPrefs类支持3中数据类型的保存和读取，浮点型，整形，和字符串型。

分别对应的函数为：

SetInt();保存整型数据；

GetInt();读取整形数据；

SetFloat();保存浮点型数据；

GetFlost();读取浮点型数据；

SetString();保存字符串型数据；

GetString();读取字符串型数据；

#### 地图处理模块

该模块处理游戏中游戏地图的加载，每个关卡有对应的地图和背景图片，该模块的功能使用一个管理类来直接管理数据。游戏中通过访问该类，传入相应的参数获取游戏地图和背景图片。由于本游戏地图与游戏操作相关性不打，所以部分地图由地图处理模块以动画（Animation）的形式进行播放。

#### 碰撞检测模块

本有系涉及到大量碰撞，包括敌人与玩家、敌人发射的子弹与玩家、障碍物与玩家、玩家子弹与障碍物、玩家子弹与敌人等等，为了保证游戏的刺激与真实性，碰撞检测与碰撞处理起着重要的作用。因此，做好碰撞检测模块的优化决定了整个游戏运行时性能的好坏。本游戏碰撞都是直接碰撞，不会出现穿过物体的情况，碰撞检测判断根据两个或者多个碰撞体来进行碰撞处理。

#### 图片及动画管理模块

游戏中会中会使用到很多的图片资源和动画，合理的管理图片资源能够使游戏运行更加的顺畅，能够使游戏表现的更加的完美。Unity3D通常将一些相关的图片制作成Atlas，通常我会使用TexturePacker软件进行图集的制作，TexturePacker是一款游戏开发中很实用的图片打包工具，他可以将很多散碎的图片打包成一张2的n次方大小的纹理图片，这样大大减少了游戏中图片的数量，并且也提高了游戏的内存优化。

## 游戏详细设计

通过以上对游戏的总体分析设计，下面来进行游戏中类的详细设计，通过游戏中功能的划分可以将游戏中所有类分为四种，第一种是游戏加载相关的类加载类，实现游戏中相关资源等等的加载功能。第二种是游戏控制类，该类实现对游戏整个逻辑数据等的控制，保证游戏的顺利执行。第三种类为游戏显示类，该类别的类实现的功能就是游戏我们能够看见的部分，也就是游戏中的界面部分。游戏中通过使用该三类类别的类来进行游戏的所有功能，并且相互协调工作，保证游戏的顺利执行。下面将对这三个类别类进行详细分析。

### 设计类

#### 加载类别类设计

游戏资源预加载类PreLoadding，在该类中实现所有资源的加载，将当前所需要的图片、音频、模型等资源先加载到内存中，以便用时直接从内存中读取。游戏音乐音效存放类MusicManager，MusicManager类进行背景音乐以及音效的播放操作。游戏数据加载类GameDateManager，这个类主要实现游戏设置信息的记录操作。

#### 控制类别类设计

控制类别类只有一个类GameControlManager类，该类控制本游戏中的所有数据的操作。该类中的成员变量有：MusicConfig类，GameDateManager类，关卡数组LevelArray，关卡信息数组LevelArrayMs，对游戏数据的访问全部通过GameDateManager类来进行访问，并且该类是一个单例，这也保证了数据的安全性，和统一的管理，游戏中只有一个该对象，这样保证数据统一无误。方便游戏中数据的管理，该类的功能等同于MVC模式中的controller控制类。处理游戏中的数据逻辑操作，得到用户的请求，并将相应请求数据发送给显示类进行显示。在该类的提供的方法有，获取TextConfig类数据getTextConfig()，得到音效播放控制类getMusicConfig()，获得关卡信息数据getLevelArrayMs()，获取关卡数组getLevelArray()。在游戏开始是通过init()方法进行数据加载初始化，在游戏退出的时候将数据进行保存SaveDate()，

#### 显示类别类设计

### 类关系图

# 游戏功能设计

## 数据块功能实现

## 界面块功能实现

### 界面加载和动画播放

### 控制获取和时间添加

### 地图功能说明

# 游戏测试

游戏测试作为软件测试的一部分，它具备了软件测试所有的共同的特性：测试的目的是发现软件中存在的缺陷。测试都是需要测试人员按照产品行为描述来实施。产品行为描述可以是书面的规格说明书，需求文档，产品文件，或是用户手册，源代码，或是工作的可执行程序。介于本游戏，进行如下相关测试：跨平台测试和游戏内容测试。本游戏主要是运行在Android平台和IOS平台中，测试需要相应的真机设备来进行测试。功能测试直接使用windows平台来进行测试。下面给出详细的测试用例和测试结果。

## 游戏测试的概念与目的

游戏测试作为软件测试的一部分，它具备了软件测试游戏测试游戏测试所有的一切共同的特性：

1、测试的目的是发现软件中存在的缺陷。

2、测试都是需要测试人员按照产品行为描述来实施。产品行为描述可以是书面的规格说明书，需求文档，产品文件，或是用户手册，源代码，或是工作的可执行程序。

3、每一种测试都需要产品运行于真实的或是模拟环境之下。

4、每一种测试都要求以系统方法展示产品功能，以证明测试结果是否有效，以及发现其中出错的原因，从而让程序人员进行改进。

总而言之，测试就是发现问题并进行改进，从而提升软件产品的质量。游戏测试也具备了以上的所有特性，不过由于游戏的特殊性，所以游戏测试则主要分为两部分组成，一是传统的软件测试，二游戏本身的测试，由于游戏特别是网络游戏，它相当于网上的虚拟世界，是人类社会的另一种方式的体现，所以也包含了人类社会的一部分特性，同时它又是游戏所以还涉及到娱乐性，可玩性等独有特性，所以测试的面相当的广。 我们称之为游戏世界测试，常由真实用户参与完成，多以封测内测等形式出现主要有以下几个特性：

1、游戏情节的测试，主要指游戏世界中的任务系统的组成，有人也称为游戏世界的事件驱动，我喜欢称为游戏情感世界的测试。

2、游戏世界的平衡测试，主要表现在经济平衡，能力平衡（包含技能，属性等等），保证游戏世界竞争公平。

3、游戏文化的测试，比如整个游戏世界的风格，是中国文化主导，还是日韩风格等等，大到游戏整体，小到NPC（游戏世界人物）对话，比如一个书生，他的对话就必需斯文，不可以用江湖语言。

G.J.Myers 在他的文献中给出了测试定义：“程序测试是为了发现错误而执行程序的过程。”所以根据这一定义，测试的目的和任务可以描述为：

目的：发现程序的错误；

任务：通过在计算机上执行程序，暴露程序中潜在的错误。反馈给开发者及先关人员进行相应的处理。

## 测试过程

本游戏测试的主要内容有游戏功能测试和不同平台的运行性能测试，兼容性测试。由于手机市场的多元化，手机的配置有高有低，手机的屏幕也是各种各样，所以本游戏在设备测试时，选择比较大众的手机配置和手机屏幕来进行游戏全方面测试。下面给出本游戏测试的所有过程和测试结果。

### 测试设备

手游的测试首先是选择测试设备，要想游戏能够在不同类型、品牌、配置的手机中运行，能够有更多的测试设备来进行测试是最好的，但是由于资源有限，本游戏使用了几款常见比较大众的手机来进行测试。并且通过使用ios模拟器和Android，模拟器来来模拟更多的设备来进行测试。

### 测试用例与结果

# 总结与展望

## 总结

## 后续工作展望

结束语

参考文献

1. c#多线程参考与实例 csdn网 [引用日期2012-09-21]
2. 关于c sharp的一系列文献 领测网 [引用日期2012-09-21]
3. 计算机发展史 csdn博客网 [引用日期2012-11-27]
4. 8月编程语言榜:.Net前景如何？ 中文业界网 [引用日期2012-010-9]

附录

致谢

在这里，首先要感谢我的导师——王超，导师严谨治学的态度，渊博的知识，无私的奉献精神使我深受启迪。他在我研究、开发本游戏的过程中一直给予我各个方面的大量支持，给我指明了“基于Cocos2d-x引擎的跨平台塔防游戏设计与实现”的设计方向和思路；大体开发流程。在设计过程中，当我遇到困难时，又及时帮助我解答疑难，最终顺利的完成了本游戏的设计；在最后毕业论文的写作过程中，帮助我查看格式，提出建议进行修改，十分仔细的帮我查看论文。从尊敬的导师的身上，不仅学到了扎实、宽广的专业知识，也学到了做人的道理。在此我要向我的导师致以最衷心的感谢和深深的敬意。

另外，我还要真心感谢在我编程过程中，曾经帮助过我的老师和同学，感谢他们无私的教诲和帮助。

最后衷心的感谢在百忙之中评阅论文和参加答辩的各位老师、评委！