

XXXX

工程实训说明书

题 目 基于 Unity 的 3D 机器人探险游戏

系 (部) 计算机科学与技术

专 业 (班 级)

姓 名

学 号

指 导 教 师

起 止 日 期 2014. 06. 11-2014. 07. 05

摘 要

本次工程实训的实训题目是基于 Unity 的 3D 机器人探险游戏。Unity 是一款很强大 3D 游戏开发引擎。在用 Unity 实现 3D 机器人游戏的过程中,在软件使用方面,需要用到 photoshop 和 3Dmax 等美工或建模软件;在脚本语言编写方面,需要涉及 JS 或 C#;本说明书从最初的 3D 探险游戏策划,到利用 Unity 相关技术实现场景物体绘制及渲染,再到 3D 探险游戏的测试,最后到游戏的完全实现及单机版的发布,对本次实训内容的实现和完成过程进行描述和说明。Unity 探险游戏主要包括如下内容:场景、检查点、widget 机器人、敌方角色。检查点用于 widget 机器人在游戏过程中的死亡后的复活。

关键字: Unity, 3D, JS, C#

目 录

摘 要	I
1 引言.....	1
1.1 编写目的.....	1
1.2 项目背景.....	1
2 需求规约.....	3
2.1 任务概述.....	3
2.2 需求规定.....	3
2.3 运行环境规定.....	5
3 游戏实现过程.....	6
3.1 游戏玩法设计.....	6
3.2 游戏内容设计.....	6
3.3 游戏界面设计.....	6
3.4 关卡设计.....	10
3.5 游戏基本架构和主要类设计.....	11
3.6 数据结构与算法设计.....	13
4 技术说明.....	15
4.1 3D 渲染.....	15
4.2 动画制作.....	15
4.3 数据存储.....	15
4.4 游戏特效.....	15
4.5 粒子系统.....	15
4.6 碰撞检测的实现.....	16
4.7 游戏 AI.....	16
5 测试用例设计.....	18
5.1 测试用例编写目的.....	18
5.2 测试用例.....	18
5.3 测试总结和改进建议.....	20
总结	21
参考文献.....	22
附录 游戏核心代码.....	23
(1) 机器人的角色控制核心代码.....	23
(2) 机器人的状态处理核心代码.....	26
(3) AI 控制核心代码.....	28
(4) 拾取物品和背囊管理核心代码.....	32
(5) 碰撞检测核心代码.....	35

1 引言

1.1 编写目的

本文档是进行项目策划、概要设计和详细设计的基础，也是编写测试用例和进行系统测试的主要依据。同时，该文档也是用户确定软件功能需求的主要依据。本文档的读者对象很多，例如，项目经理、设计员、程序员、测试员、销售员、用户、其他人员如公司领导等。因此，运用Unity进行3D探险游戏开发的开发人员进行开发之前，需要仔细阅读该文档，明确需求分析，并对场景构思有较好的理解。

1.2 项目背景

1.2.1 Unity 简介

Unity是一款跨平台的游戏开发工具，从一开始就被设计成易于使用的产品。作为一个完全集成的专业级应用，Unity还包含了价值数百万美元的功能强大的游戏引擎。Unity3D是一个让你轻松创作的多平台的游戏开发工具，是一个全面整合的专业游戏引擎。

1.2.2 项目背景简介

本项目的名称为用Unity进行3D探险游戏开发。由软件工程（数字媒体）教研室提出，有数字媒体方向学生具体实现，实现过程包括：游戏策划、场景设计、3D建模、渲染贴图、测试、用户体验等。

进入21世纪后，游戏动漫设计已经成为全球十分热门的行业。游戏动漫产业的巨大影响力不仅凸显在我们的生活中，其发展过程中产生的经济效益也吸引了越来越多投资者的眼光，成为了新一轮经济发展的增长点。而本次工程实训3D场景渲染的开发，最终将获得一个小型的游戏或动漫3D场景，而现在的游戏或动漫中3D产品已经成为主流。

飞速发展的游戏和动漫产业在市场上已经带来了非常可观的收益。

从市场规模来看，2009年中国网络游戏市场规模达到258亿元人民币，同比增长39.5%。其中：国产网络游戏市场规模达到157.8亿元人民币，比2008年增长41.9%，占总体市场规模的61.2%。

从主要细分游戏市场来看：大型多人同时在线角色扮演游戏(MMORPG)市场的主导力量，2009年MMORPG整体网络游戏市场的比例约为79%，市场规模达到203.8亿元，较2008年增长35.2%；高级休闲游戏(ACG)的增长速度在加快，2009年高级休闲游戏占总体网络游戏市场规模的比例约为13.8%，市场规模达到35.5亿元，较2008年增长49.7%。

因此，作为数字媒体专业的学生，在毕业前接触相关的游戏或动漫开发的案例，有利于熟悉游戏或动漫的开发流程，能为以后从事游戏或动漫方向的就业打下良好的基础。

2 需求规约

需求规约包括任务概述、需求规定、运行环境规定三个部分。任务概述将对 Unity 的 3D 探险游戏需要完成的任务做简要说明；需求规定将从业务需求方面对 3D 探险游戏的实现进行需求分析；运行环境规定，即实现 3D 探险游戏场景渲染需要的硬件和软件支持。

2.1 任务概述

Unity3D 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 MacOSX 下，可发布游戏至 Windows Mac, Wii、iPhone 和 Android 平台。也可以利用 Unity web player 插件发布网页游戏手机游戏 支持 Mac 和 Windows 的网页浏览。

本次实训的题目为基于 Unity3D 的机器人探险游戏技术实现，旨在成功的设计出一款可以运行在智能手机和平板电脑上的探险游戏。

该项目主要界面和功能包括：

- (1) 界面设计的美观 有一种三维仿真的效果。
- (2) 系统中的模型进行移动和碰撞检测。
- (3) 场景中的树实现风吹动的效果。
- (4) 机器人对小兔子的攻击
- (5) 拾取功能、背囊管理。

2.2 需求规定

2.2.1 对功能的规定

本项目主要功能有：开始游戏，帮助，退出游戏，机器人的移动控制，

背囊管理，小兔子的 AI 控制。

开始游戏，loading 后用户进入游戏。

表 1 游戏功能表

编号	功能名称	使用角色	功能描述	输入	系统响应	输出
1	开始游戏	玩家	点击按钮用户进入游戏选择模式	按钮输入	弹出游戏模式按钮	
2	帮助	玩家	点击按钮用户进入游戏帮助	按钮输入	进入游戏帮助	
3	退出游戏	玩家	点击按钮用户退出游戏，游戏结束	按钮输入	退出游戏，游戏结束	
4	机器人的移动控制	玩家	Wasd键控制机器人的移动	键盘输入	根据玩家的输入机器人移动	
5	背囊管理	NPC	保存玩家找到的奖品	无	奖品保存到背囊中	
6	小兔子的 AI 控制	NPC	控制小兔子的状态	无	小兔子的运动状态	

2.2.2 对界面的规定

本项目涉及界面包括开发菜单界面，游戏界面。菜单界面用于显示游戏的菜单，包括开始游戏，退出游戏。游戏界面用于 3D 显示游戏的各个场景，物体，地形等元素。

2.2.3 故障处理要求

软件可能出现的故障：

表 2 游戏故障预测表

故障类型	故障处理
机器人无法正常操作	重新运行

2.3 运行环境规定

2.3.1 硬件要求

操作系统 (Windows) XP、Vista32 位、Win7。CPU主频 P4 1.4GHz 及以上。
内存 256MB以上。显存支持独立、集成显卡，要求显存 32MB以上。硬盘大小 2.2GB 以上。屏幕分辨率 800*600 及以上。

2.3.2 软件要求

Unity4.0 或 4.0 以上版本，photoshop 和 3Dmax等设计或建模软件，绘制类图软件 StarUML，microsoft Word2003 等等。

3 游戏实现过程

3.1 游戏玩法设计

玩家进入游戏场景，玩家通过点击进入游戏，进入游戏界面后。我们会看到一个机器人。使用 **wasd** 键来控制机器人的行动方向。在界面的上方我们可以看到机器人的生命值。下面有可以使用的道具。明白以后你就可以开始我的探险游戏。在行走的过程中机器人会和山体树木发生碰撞，机器人的运动状态也会发生改变。该探险类游戏为单人游戏。该探险游戏的操作方式比较单一，主要是控制游戏角色的行走方向以及向敌人进行攻击。方向键盘包括上下左右，分别用于控制机器人的前进和后退，左右转角操作。而 **ctrl** 键用于攻击敌人。

3.2 游戏内容设计

3.2.1 关于剧情

进入游戏后，玩家进入一个森林的游戏场景，玩家可以在道路上移动，在行走的过程中你可能会遇到一些物品，可以拾取到自己的背囊。还会遇到一个会攻击机器人的小兔子。在玩家战胜敌人后会有奖励。

3.2.2 关于角色

该游戏主要包括：玩家和兔子。

玩家控制机器人向上下左右四个方向移动，并对兔子进行攻击。兔子在遇到机器人的时候也会对玩家进行攻击。

3.2.3 关于场景

在 **unity** 中使用地形和模型的组合来雕琢游戏世界的空间，首先定制地形，在 **unity** 中可以使用高度图来创建高度，使用笔刷来绘制高度，绘制贴图等，并添加光照、阴影和天空盒和基于距离的无话效果，还有在地形上加入了水体。

3.3 游戏界面设计

3.3.1 游戏界面的框架设计

Game 该游戏没有复杂的游戏系统，更加侧重游戏可玩性的设计，并且由于使用 Unity3D 引擎，省去很多底层模版的设计，其中核心的木板为：图形渲染，物理仿真，网络联机。游戏系统的框架结构图如下：

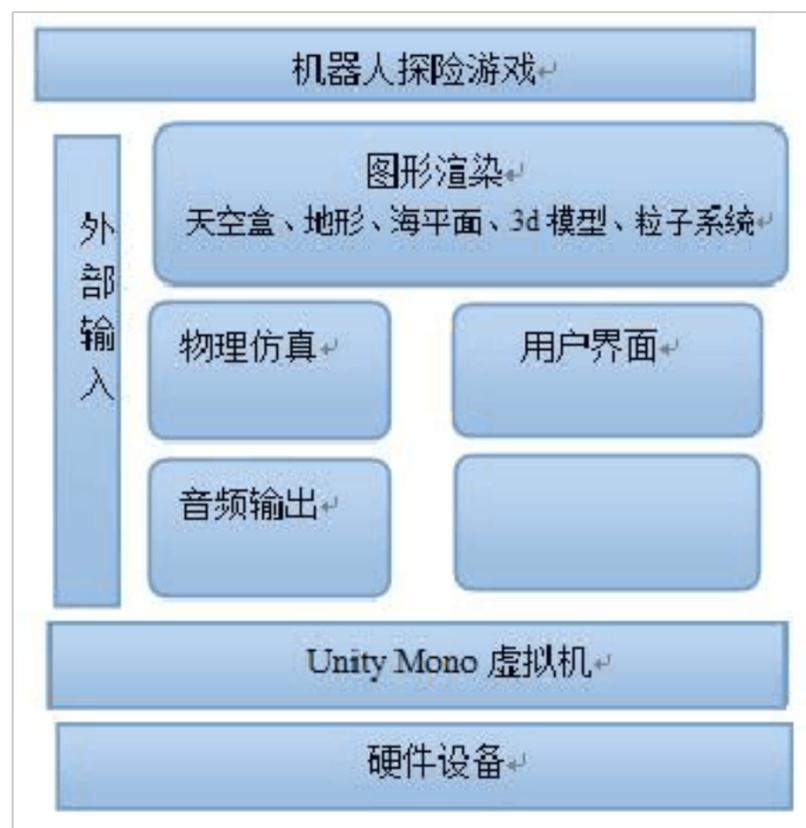


图 1 游戏系统的框架结构图

3.3.2 界面设计

本游戏作为 3D 游戏，3D 游戏界面中主要实现游戏中场景和各个部件的绘制和控制，包括地面，天空，山体、树木等。各种游戏模型及界面如下所示：



图 2 机器人角色模型



3 敌人角色模型



图 4 树角色模型



图 5 设置触发器

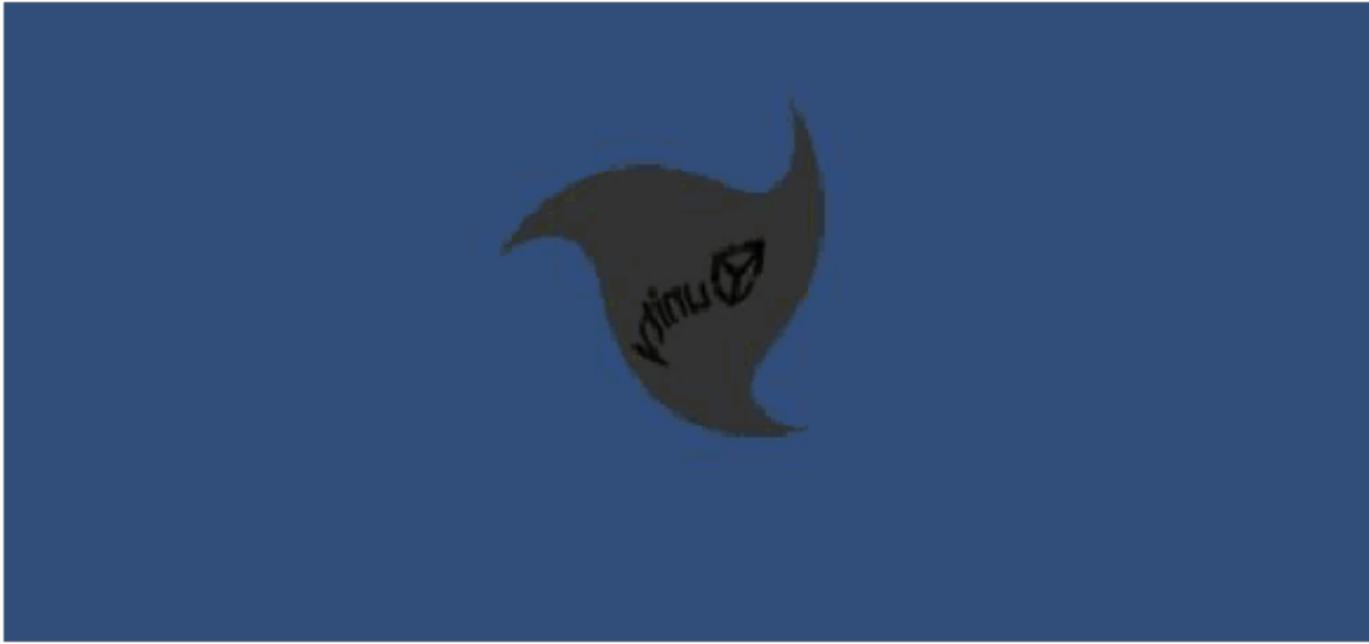


图 6 设置着色器



图 7 游戏菜单界面



图 8 积分榜界面



9 游戏主界面

3.4 关卡设计

3.4.1 关卡及关卡中游戏元素

该机器人探险游戏设有一个关卡，关卡由地形，天空盒，光照和阴影，山体，道路，天空，水体等元素构成。

地形：Unity3D 中的地形模块不但可以导入高度图，还可以用笔刷为绘制

天空盒: 在 Unity 中, 天空盒是一个特殊的虚拟盒子, 它包裹在场景周围; , 在渲染设置中为整个场景一次性指定一个天空盒。

光照和阴影: 场景以光源的位置为视点被渲染, 每个渲染图像的像素的深度值被记录在一个“深度纹理”中, 被称为阴影贴图。

3.4.2 操作设计

关卡的操作是通过键盘实现坦克的运动和攻击。

3.4.3 运行时间

每关运行时间是由开始游戏为起点玩家的失败或者胜利最为结束点。

3.4.4 关卡设计流程

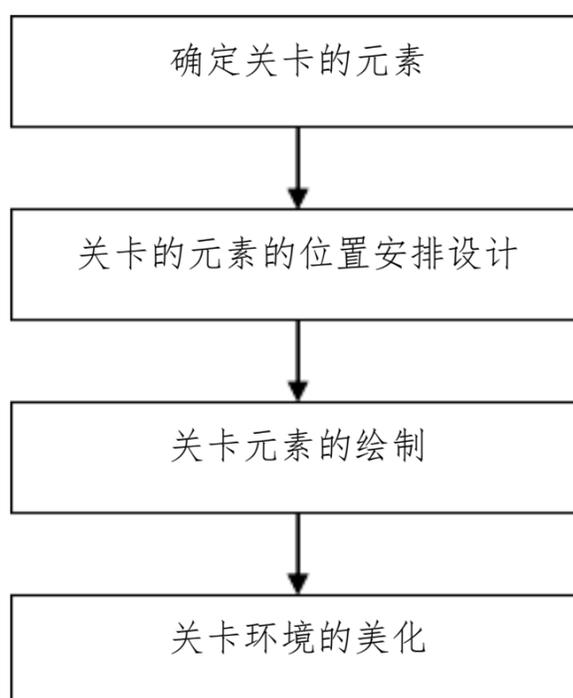
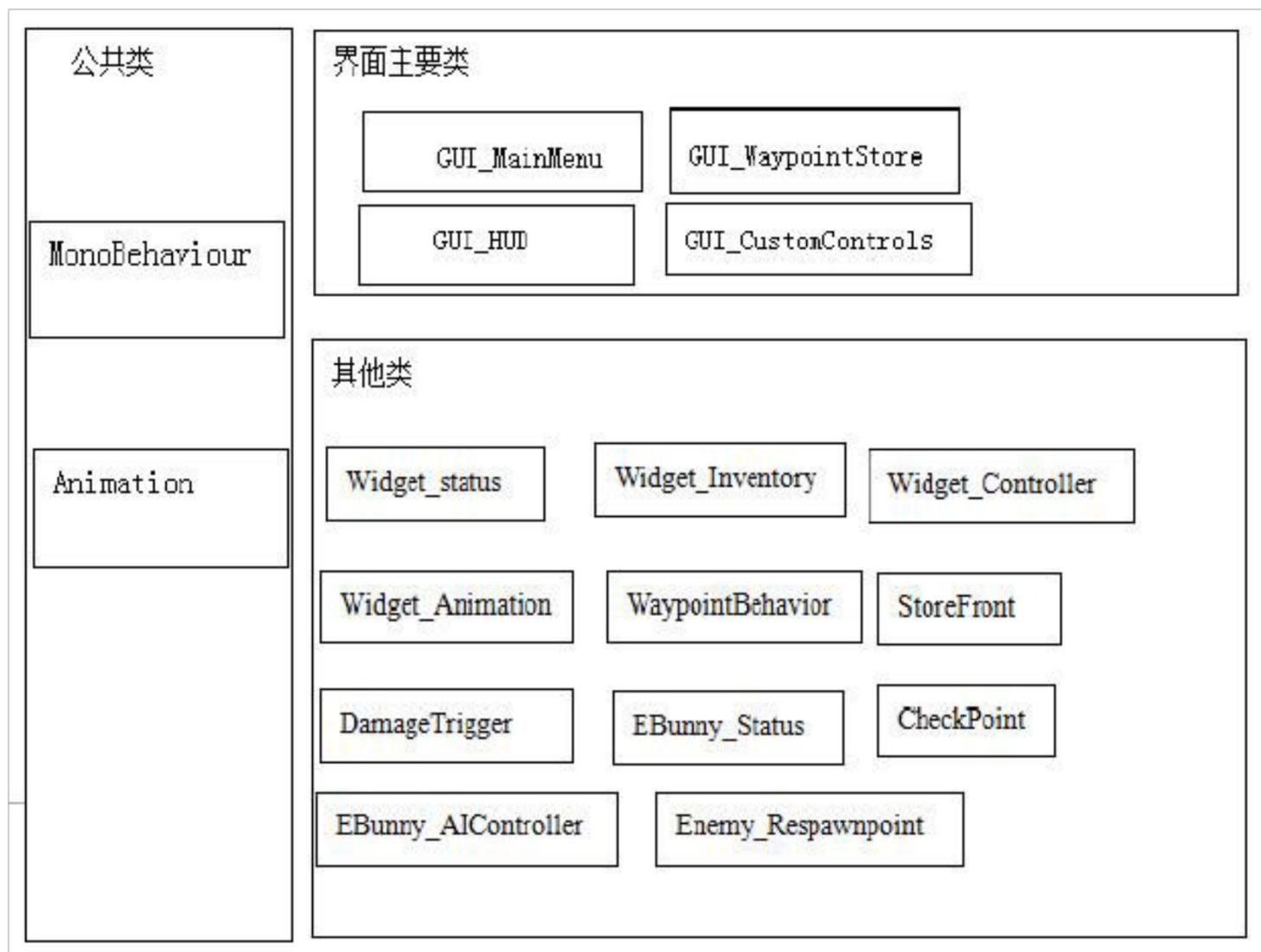


图 10 游戏关卡流程

3.5 游戏基本架构和主要类设计

3.5.1 游戏的基本架构

(1) 游戏的类框架



11 游戏的类框架图

(2) 游戏界面的基本跳转

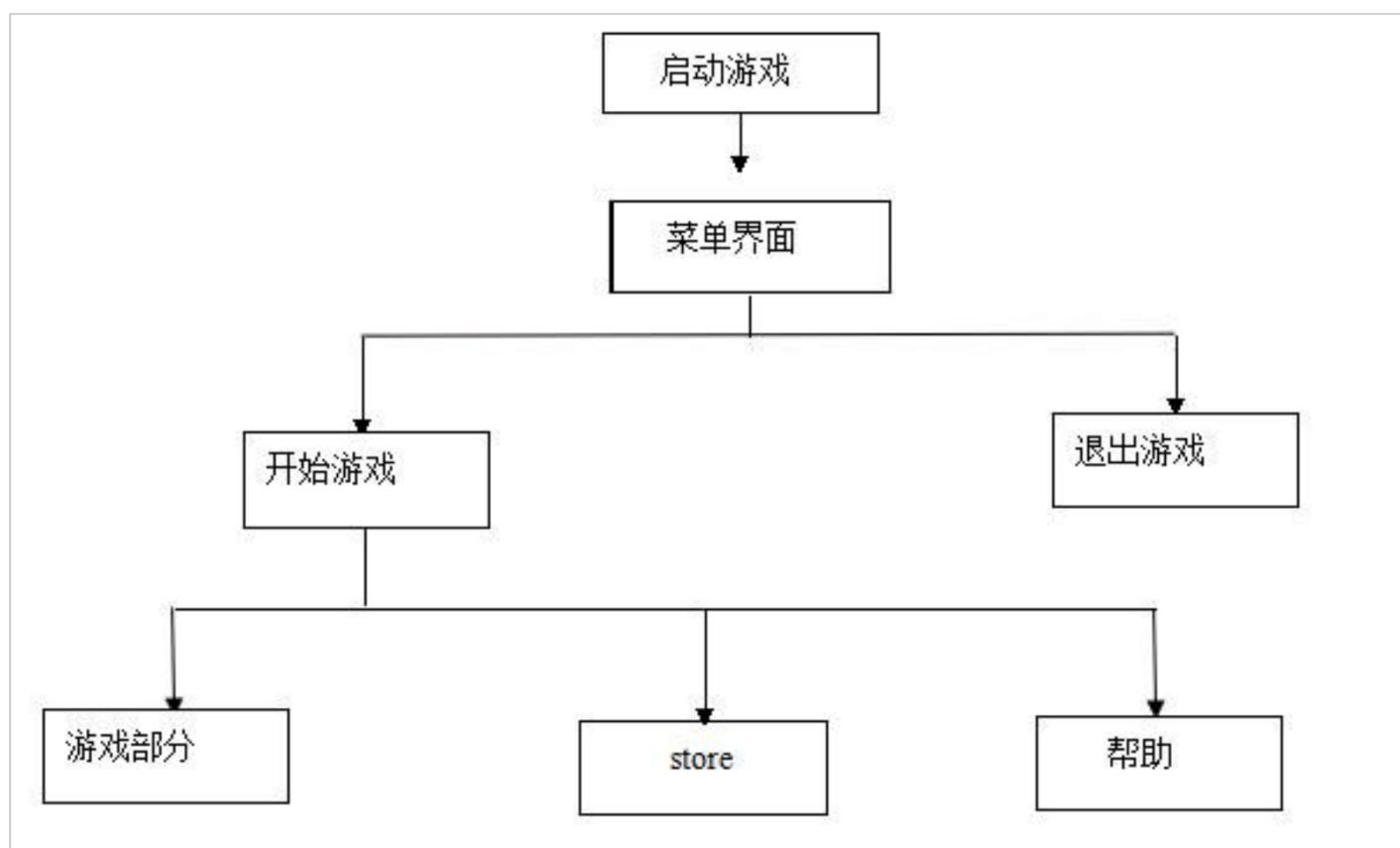


图 12 游戏界面跳转图

3.5.2 主要类的设计

游戏中开发的类比较多，为了让读者能够更好的理解各个类的在游戏中所扮

演的角色，本节将按照前面的框架分类对各个类的功能进行简要介绍。

(1) 公共类

MonoBehaviour 类：每当想要在 Unity 中创建一个新的 JavaScript 脚本时，会自动从该类继承，是的脚本可以访问所有该类的内置函数以及继承而来的成员，它控制了和碰撞检测、鼠标事件、摄像机事件、组件获取以及比较相关的绝大部分函数。

Animation 类：该类负责对动画剪辑进行播放、组合、和混合的所有功能。

(2) 游戏相关脚本代码

Widget_Cntroller.js : 用于处理部件的运动和玩家的输入

Widget_Status.js : 用于处理 widget 的状态。

Widget_Camera.js : 用于控制摄像头和使它顺利地跟随部件。

Widget_Inventory.js : 这里包括所有部件的收集的项目更新。

Widget_AttackController.js : 处理玩家的攻击输入和对目标敌人的伤害。

Widget_Animation.js : Widget 的动画状态管理器。

WaypointBehavior.js : 处理脚本在所有的 waypoint

StoreFront.js : 在 waypoint 处理采购变化

PickupItems.js : 处理所有 widget 可以拿起的部件

GUI_WaypointStore.js : 用于交易存储接口处理

GUI_MainMenu.js: 为场景增加背景和主要的导航按钮

GUI_HUD.js 显示 pertinent 信息窗口，他的项目，和任何现有的敌人。

GUI_CustomControls.js : 为在别处使用 `gui_customcontrols` 包含自定义复合控件类。

Enemy_RespawnPoint.js : 附加到场景中去作为敌人的重生点。当玩家走到指定的区域，一个新的敌人将重生。

EBunny_Status.js : 小兔子敌人的信息状态控制。

EBunny_AIController.js : 用于处理电子兔子的 AI 和动画。

DamageTrigger.js : 一个简单的，可应用于任何类型的对象变量的损伤触发。用于检查损伤量的变化。

Checkpoint.js : 静态声明变量。

3.6 数据结构与算法设计

3.6.1 数据结构

链表，哈希表是常用的数据结构，在该游戏中未用到相关的数据结构。

3.6.2 算法设计

(1) 算法一：判断玩家是否在攻击范围内

首先设置小兔子的攻击半径，当小兔子查看到自己离玩家的目标距离小于它的攻击半径时，小兔子就开始攻击。

```
distanceToPlayer = transform.position - target.position;  
If(distanceToPlayer.magnitude < attackDistance)  
Return;  
Yield;
```

(2) 算法二：判断玩家和小兔子的相对角度，使他面向玩家并向玩家移动。

使用简单的三角函数恒等式，我们就可以找到对应的角度，在 `FacePlayer()` 函数中加入如下代码：

```
var relativeLocation =  
transform.InverseTransformPoint(targetLocation);  
Var angle = Mathf.Atan2(relativeLocation.x,  
relativeLocation.z) * Mathf.Rad2Deg;
```

4 技术说明

4.1 3D 渲染

绝大多数 3D 图形库实现都有相似操作顺序，一系列相关的处理阶段叫做 3D 图形的渲染管线，基于顶点的操作与基本装配：这个过程包括本地空间—世界空间—视图空间—背面拣选，主要是根据编程人员设置的摄像机位置、视锥属性和顶点索引对顶点数据进行一系列的数学运算，使 3D 图形的形状及位置得以正确显示。像素操作：该过程主要根据编程人员传入显卡的纹理及顶点的纹理坐标以及光照信息进行颜色计算、混合，并对线、面采用线性插值、二次线性插值等算法最终进行出像素的颜色。这个过程的计算一般由显卡固定管线的计算公式完成，如想得到更真实、更多样的效果，可采用可编程管线技术实现。

4.2 动画制作

Unity 中的动画是由关键帧控制的。他们定义了动画曲线。关键帧基本上保存了一个给定对象在某个特定时刻的状态的快照，曲线定义为链接两个或多个关键帧，以便对给定对象的完整运动路径进行插值，一个对象的每个部分或者是组件都可以为任意给定的动画剪辑包含自己的关键帧和曲线。

4.3 数据存储

文件存储。

4.4 游戏特效

在游戏中，游戏特效用于制作火花效果，很引人注目。游戏特效使游戏画面更加充满艺术的真实感，使玩家得到实时的沉浸感，更酣畅淋漓地体验游戏的乐趣。在实时环境下有许多有趣并具有创造性的方法来实现游戏的特效，。下面主要介绍粒子系统。

4.5 粒子系统

粒子系统即一组从特定物体或者位置发射出小的颗粒，游戏中这些小的粒子

通常用的是小的二位图形和点图形，这些图形都带有小的纹理和 Alpha 通道，另外粒子具有生命周期、速度等属性，并受重力等影响。粒子系统的应用是最为广泛的，游戏中常常用来表现烟花、炮火、喷泉等等。游戏中为 widget 的攻击设置了火花的效果。

4.6 碰撞检测的实现

在 Unity3D 中，能检测碰撞发生的方式有两种，一种是利用碰撞器，另一种则是利用触发器。碰撞器的基本概念：

(1) 碰撞器是一群组件，它包含了很多种类，比如：BoxCollider，Capsule Collider 等，这些碰撞器应用的场合不同，但都必须加到 GameObject 身上。

(2) 所谓触发器，只需要在检视面板中的碰撞器组件中勾选 IsTrigger 属性选择框。

(3) 在 Unity3d 中，主要有以下接口函数来处理这两种碰撞检测：触发信息检测，碰撞信息检测。

在该游戏中，我们创建一个齿轮这个预制模型，在 project 是视图中选中它，并选择 Component Physics → Sphere Collider，为这个物品赋予一个新的 Sphere 碰撞体。在 Sphere Collider 组建的 Radius 设置上面，单机 Is Trigger 框把它设置为 True。这样这个碰撞体现在的行为就像一个触发器形体一样。两个 GameObject 发生碰撞，要想检测到触发信息，最少要有一个碰撞器并且勾选了 IsTrigger 复选框，另一个最少要有一个碰撞器组件。

4.7 游戏 AI

AI 的全称是 Artificial Intelligence（人工智能），AI 是计算机学科的一个分支，它主要模拟人的思维过程，使计算机可以像人一样行动和思考。

AI 在游戏中就是所编写的代码，如在给定的情形下，当游戏有多种选择时，计算机控制的角色采取的是看起来比较聪明的决策，产生与之相关的有效和有用的行为。

在游戏中，AI 一般用在以下几方面：游戏中智能体的设计（当游戏世界环境变化时，该智能体能够感知环境并自动做出相应的动作）；游戏世界中的寻路或者导航；其他 AI 技术如模糊逻辑的运用。

游戏 AI 技术通常可以分为两种，定性和非定性。定性 AI 适用于环境完全确定的情况，但现在这样的环境其实是非常少的。非定性 AI 则可以适用于一些不确定的环境，但相对应的，其做出来的决策可能并不是最优的。

在以前，游戏中最广泛使用的 AI 技术就是作弊，例如在战争模拟游戏中，由计算机控制的玩家，可以得知对手的所有信息，这样根本就不需要侦察兵去收集这类情报。可是这样会造成游戏中很多的不公平，而导致可玩性的下降。现有的 AI 技术，以及某些人工生命技术，种类繁多，这里不再一一罗列。

在机器人探险游戏中，我们实现小兔子自由的四处游荡，直到它看到一个玩家并且上去追捕它。小兔子在设定的区域范围内转向并向目标移动，在与目标发生碰撞就开始攻击，当移动到范围外，继续游荡，如果被打败，丢下随机小物品并且消失。

AI 在游戏中就是所编写的代码，如在给定的情形下，当游戏有多种选择时，计算机控制的角色采取的是看起来比较聪明的决策，产生与之相关的有效和有用的行为。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/286154024211010125>