**基于unity3D的《我在修仙世界种田》游戏设计与实现**

**数学与信息学院 数字媒体技术**

**黄世琛 116062017081 指导老师：吴献**

**【摘要】**

**随着当今游戏行业的逐步发展，个人进行游戏开发的难度也正在逐步的下降，国内外进行独立游戏开发的工作室也越来越多。并且随着Steam和Epic等游戏商店逐渐的让国内更多的人熟知，买断制游戏也逐渐的被国内的游戏玩家们接受。而国内的游戏一直是游戏免费而包含内购的模式，对于买断制游戏的需求也随着网络的发展越来越高。然后则是目前的游戏的类型大多仍然是角色扮演以及动作，而模拟经营类的游戏相对比较小众，但是其游戏粘度较高，玩家也更加的会偏向喜欢该类型的游戏。而且对于国内的玩家来说修仙类型的游戏会更加具有吸引力，更加的适合国内的玩家群体。本游戏设计是使用可以跨平台进行游戏开发的Unity3D引擎来实现的模拟经营，修仙独立游戏。本文首先根据当前市场以及自身游戏特点进行分析验证其可行性，然后根据自身设计需求进行概要设计，最后进行界面的设计和代码的编写实现了整个游戏。**

**【关键词】：模拟经营；独立游戏；修仙；Unity3D；**

目录

[1. 绪论 4](#_Toc2434)

[1.1 课题背景和意义 4](#_Toc12230)

[1.2 课题概述 4](#_Toc23812)

[1.3 开发工具 4](#_Toc22173)

[2. 游戏开发工具概述 4](#_Toc392)

[2.1 Unity3D引擎概述 4](#_Toc9875)

[2.2 Microsoft Visual Studio 2017概述 4](#_Toc16188)

[2.3 数据存储概述 4](#_Toc12884)

[2.4 第三方工具概述 5](#_Toc25737)

[3. 需求分析 5](#_Toc9057)

[3.1 功能需求 5](#_Toc16033)

[3.1.1 场景切换功能 5](#_Toc7419)

[3.1.2 音效播放功能 5](#_Toc2892)

[3.1.3 游戏自动保存功能 5](#_Toc5930)

[3.1.4 建造功能 5](#_Toc22095)

[3.1.5 快捷物品功能 5](#_Toc31507)

[3.1.6 背包功能 5](#_Toc1765)

[3.1.7 商店功能 5](#_Toc21250)

[3.1.8 种植功能 5](#_Toc17952)

[3.2 性能需求 6](#_Toc13271)

[3.3 美术需求 6](#_Toc27948)

[3.3.1 3D美术需求 6](#_Toc29361)

[3.3.2 UI需求 6](#_Toc25285)

[3.4 数据需求 6](#_Toc4362)

[3.4.1 游戏数据需求 6](#_Toc18797)

[3.4.2 数据存储需求 6](#_Toc23202)

[4. 概要设计 7](#_Toc11286)

[4.1 游戏框架概述 7](#_Toc19442)

[4.2 游戏总体设计 7](#_Toc1377)

[4.3 游戏模块设计 7](#_Toc19708)

[4.3.1 事件管理模块 7](#_Toc7591)

[4.3.2 资源管理模块 7](#_Toc27557)

[4.3.3 数据管理模块 7](#_Toc24157)

[4.3.4 角色管理模块 7](#_Toc23054)

[4.3.5 地图管理模块 8](#_Toc28060)

[4.3.6 UI管理模块 8](#_Toc22659)

[5. 详细设计 8](#_Toc15050)

[5.1 详细设计概述 8](#_Toc12272)

[5.2 事件系统详细设计 8](#_Toc11710)

[5.3 资源管理详细设计 8](#_Toc18692)

[5.4 数据管理详细设计 8](#_Toc29653)

[5.5 角色系统详细设计 9](#_Toc3431)

[5.6 UI详细设计 10](#_Toc718)

[6. 结语 10](#_Toc27572)

[致谢 11](#_Toc26471)

[参考文献 11](#_Toc26471)

[英文摘要 1](#_Toc26471)2

1. **绪论**

## 课题背景和意义

当今市场上的主流游戏任然是角色扮演、卡牌和动作格斗游戏。模拟经营类游戏如今虽然比不过其他类型的游戏，但是其自由发挥性强、公平性强、游戏黏度高，操作难度低等优点也吸引很多玩家。如今市场上的模拟经营类基本要素为模型仿真，核心玩法完善，细节精美。加上模拟经营游戏本身就是高度重复的作业，留下的玩家都是经的起单调循环的忠实玩家，很难被同系的游戏吸引。如果新的模拟经营游戏想要突出来，必须在游戏玩法上作出革新。本课题在原有的种田养成的基础上，添加了修仙的元素进去，有了创新的元素，势必可以吸引一批对修仙感兴趣的玩家。将中国文化的元素同独立游戏结合起来，可以将中国文化的美感和魅力更加趣味性的表达出来，创造沉浸的文化氛围，激发玩家对中国文化的兴趣[1]。开创性的玩法设计，可以让游戏内容更加的丰富有趣，并且有着更多的想象空间，让游戏可以不拘泥于普通的种田，而可以使用奇幻的手法来种田。同时添加有趣的剧情以及NPC让玩家更加有代入感和沉浸感。

## **课题**概述

游戏名字为《在修仙世界种田》，游戏类型为模拟经营，游戏为3D，游戏视角为2.5D固定俯视角，游戏面向的玩家群体为爱好模拟经营类游戏并且喜欢带有修仙元素的玩家们。游戏主要玩法为种田，玩家在游戏中需要合理使用工具和体力来进行种植，并在成熟后进行采摘，然后于NPC处换取金钱后去购买其他道具来更好的去经营自己的农场，并且添加部分修仙元素在内，丰富玩家除种田外的游戏时间以及改善玩家的种田体验。

## 开发**工具**

代码编写：Microsoft Visual Studio 2017

游戏引擎：Unity2018.4.28f1

数据存储：Excel，XML，二进制文件

其他插件：DOTween，protobuf-net

1. **游戏开发工具概述**

## **Unity3D**引擎**概述**

Unity3D是由Unity Technologies公司开发的跨平台专业游戏引擎，它打造了一个完美的游戏开发生态链，用户可以通过它轻松实现各种游戏创意和三维互动开发，创作出精彩的2D或者3D游戏【2】。可以进行多平台游戏开发。同时有着很好的可视化界面，可以方便的进行场景的编辑，游戏的预览运行与调试。同时还可以通过其自带的商城进行游戏资源的购买，方便的获取游戏所需的资源。并且该引擎的个人版是免费的，对于学生来说完全够用了。

## **Microsoft** Visual **Studio 2017概述**

该软件可以支持进行多种语言的程序开发，同时在安装了Unity3D的相关内容之后可以更好的适用于Unity3D引擎进行游戏开发。可以快速的查找代码问题和对代码进行调试，并且提供了方便的代码补充，使得代码编写更加的方便和迅速。同时可以快速的访问对应的代码块和快速的类调用。

## **数据**存储概述

本游戏据存储主要是通过Excel，XML和二进制文件进行存储，其中最终的数据将存储于二进制文件中。由于需要频繁的进行保存和访问数据所以使用二进制文件进行存储这样可以节省存储空间和转换时间[3]。二进制文件是所有类型文件中最小的文件，占用的存储空间非常小，并且不易被他人破解。Excel文件是表格类型的文件格式，该格式的文件便于阅读和编辑。XML类型则是另外一种文件格式，该文件相对Excel占用空间更小，并且格式严谨，便于阅读。

## **第三方工具概述**

第三方工具主要是他人开发的，可以用于辅助游戏开发的工具。其中DOTween插件主要用于实现一些动画效果，更加便捷的去实现移动，旋转，消隐等动画效果。Protobuf-net则主要用于数据的转换，便于将数据与二进制数据相互转换。PROTOBUF的全名是Protocol Buffer,是Google为网络中为开源目标提供的序列化模型,无视语言与信息平台限制,具有优秀扩展性和沟通性,可用于各种类型的通信协议,是信息存储的数据串联工具。可以通过序列化将数据转换为二进制，也可以将二进制数据反序列化回对应的数据[4]。

1. **需求分析**

## 功能**需求**

* + 1. **场景切换功能**

游戏中存在多个游戏场景，所以需要进行场景的切换，同时在场景切换完毕后需要对场景进行相应的修改。所以需要有一个场景管理对场景的切换进行相应的处理，统一所有对场景进行的操作。

* + 1. **音效播放功能**

游戏的音效可以随着时间的推移会发生变化，同时可以自行设置音效，以及对声音的大小进行设置等功能。

* + 1. **游戏自动保存功能**

游戏存档是大部分游戏中不可或缺的一部分，不管是网络游戏，还是单机游戏，都需要记录玩家从开始游玩到结束游玩为止的数据[5]。本游戏中每过完游戏内部的一天时间，会对游戏进行存储，保存之前的游戏进度，从而让玩家可以再次从该存档点继续游戏。

* + 1. **建造功能**

在游戏中玩家可以通过收集材料获取对应的建筑物品然后放置与自己的家园中。玩家可以通过选择该物品来建造于自身的家园中，同时可以预览物品的摆放样式从而可以方便的查看摆放是否合理，是否可行。

* + 1. **快捷物品功能**

在游戏的主页面中，有着玩家的快捷物品栏，玩家可以通过输入指定按键快捷选择对应的物品进行使用。同时玩家拥有多个快捷物品栏页，通过按下对应的按键进行快捷物品的切换，可以快速的进行其他物品的使用。该快捷物品栏页可以通过游戏内部的NPC商店进行购买。

* + 1. **背包功能**

玩家有着储物戒指可以进行物品的存储，通过特殊渠道可以对储物戒指进行容量的扩大，同时可以将储物戒指中的物品放入快捷物品栏进行快速的使用。

* + 1. **商店功能**

游戏内有NPC商店可以进行物品的购买，主要出售种子以及部分特殊道具。商店功能可以有效回收玩家手上的货币，让玩家能将游戏内的货币转换成其他的更加有用的物品。同时商店还可以将玩家手中不需要的物品换成金钱，是一个玩家将自身物品转换为其他物品的一个中转站。

* + 1. **种植功能**

玩家可以将种子播种到可以播种种子的土地上，在种植的植株成熟之后玩家可以摘取果实，从而获取到可以与NPC进行交易的果实。

## **性能需求**

（1）能够实时相应玩家的操作，如进行锄地操作需要快速判断能否锄地成功并对对应地块进行快速的状态变更。

（2）可以快速的过渡场景，不卡顿，不掉帧，游戏可以流畅的运行下去。

（3）数据存储和读取速度及时，对于有修改的数据可以快速的进行存储。

## **美术需求**

电子游戏美术是指在游戏中所能看到的一切画面，美术设计在游戏中的应用不仅广泛, 而且特别重要, 决定了一款游戏的画面[6]。项目的游戏背景是在修仙世界中，所以游戏的美术风格主要为古代仙侠。

* + 1. **3D美术需求**

3D美术需求，包括场景，角色，植物，道具等。

场景包括一个玩家需要经营的家园，以及一个NPC生活的城。在玩家的家园是于一处山谷中的，所以主要是需要一个偏向于田园的场景布置，同时包括一个玩家休息的房间。NPC生活的城市主要为NPC进行活动的场景，所有的NPC全部存在于此处，所以此处的主要风格是大量的房间以及部分特殊NPC所需的场景道具。好的场景可以让玩家更有沉浸感。

然后是角色，角色包括主角和NPC，并且需要对主角和NPC进行骨骼的绑定，并为其制作骨骼动画，让角色可以动起来。角色能否在游戏里面显得自然，全都是靠骨骼动画来实现的[7]。角色的动作可以让整个游戏都活动起来，否则的话就好像在和木头说话，毫无反应又没有意思。

之后是植物，植物模型需要有植物的整个生长周期下的不同样子的模型，让植物随着生长逐渐改变模样，让玩家可以清楚的了解到植物当前的状态，并且对植物成熟更加抱有期待感。

最后是道具的模型。道具主要是一些农用的道具，包括锄头，浇水桶，炼丹炉等等。道具是玩家用来使用后达到某种目的的物品。农用道具是为了辅助玩家进行植物的种植，炼丹炉等其他道具则是为了扩展游戏的玩法。

* + 1. **UI需求**

UI的需求主要为游戏界面的UI需求，游戏图标的UI需求。游戏的设计应当遵循UI设计的一般性原则，设计出能够满足玩家需求的高质量游戏项目[8]。一个好的UI界面可以让玩家更好的了解游戏的操作，从而不需要更多的引导，玩家便可简单的上手游戏。游戏的图标则可以让玩家更好的分辨不同的游戏物品。

游戏界面主要是游戏的开始界面，以及玩家主界面和商店界面，背包界面等。

游戏图标UI需求，主要为游戏内不同的物品道具的图标。

## 数据需求

* + 1. **游戏数据需求**

游戏中较为重要的一个部分就是游戏的数值策划了，合理的数值可以让游戏的过渡更加的平滑,让玩家保持对游戏的喜爱，而不合理的数值设计则会导致游戏进程的卡顿或飞跃，玩家缺少某些东西导致卡关，玩家东西非常多导致直接可以快速的越过游戏的阶段性内容等[9]。

游戏数据主要是一些不可修改的默认数据以及游戏运行过程中产生的数据，比如一个作物的总生长周期，作物的类型，物品的ID以及物品的价格等等就属于不可以修改的数据。不可更改的数据，在游戏运行过程需要的时候再去对其进行查找和使用。而像作物位置，作物当前的生长时间，玩家拥有的物品等等就属于游戏运行过程中产生的数据。这些数据则需要进行存储下来，以便玩家下次进入游戏时可以继续游戏。

* + 1. **数据存储需求**

数据存储主要是为了让游戏不必一次性的游玩，而可以分多次的去体验游戏内容。

数据主要采用二进制文件进行存储，减少数据占用的存储空间，也可以有效避免玩家直接修改数据的外挂行为。

1. **概要设计**

## 游戏框架概述

本游戏将整个游戏框架分为了数据、UI和角色三个部分，并使用事件系统为整个游戏的事件触发提供便利。这样的话可以达到较高的内聚和较低的耦合度，使项目更加容易修改和调试。

## 游戏总体设计

游戏的主页面主要包括新游戏，继续游戏，设置和退出游戏四个按钮

点击新游戏开始一把新游戏，同时会为主角生成初始数据并加载新场景。

点击继续游戏会弹出当前已经存在的游戏存档并提供玩家选择，玩家选择对应的存档后，将会读取数据对应数据并加载场景。

点击设置可以设置游戏的音量和摄像机的远近等游戏内信息。

点击退出游戏可以退出游戏。

游戏存在玩家生活的场景和NPC生活的场景，玩家初始出现在自身生活的场景，玩家可以通过传送阵前往NPC主城去与NPC进行互动。

玩家生活的场景包括一个休息的房间以及一片可以开垦的土地。

NPC有售卖一般物品的商人和特殊商人，有算命的道士，以及普通的群众等。

玩家的UI主要为一个快捷物品栏以及玩家的当前状态和游戏的状态，快捷物品栏可放置快捷使用的物品，玩家当前状态UI显示当前玩家状态，包括玩家的生命值、体力值。游戏的状态主要显示当前的季节和时间，以及当前的天气状态等。

玩家可以通过按键打开背包，查看背包物品，并对背包中的物品进行位置的移动。

## 游戏模块设计

* + 1. **事件管理模块**

该模块主要负责注册和广播事件，可以为不同的事件注册不同的响应函数，也可以为同一个事件注册多个响应函数。该模块可以减少程序的耦合度。

* + 1. **资源管理模块**

该模块负责对资源的加载和缓存，对于需要多次加载的物品可以将其缓存下来这样可以减少加载的次数，提升游戏的性能

* + 1. **数据管理模块**

（1）外部工具数据转换模块，该模块为外部辅助工具，通过读取Excel文件内容，对其进行解析后将其内数据先转换为XML同时生成Proto类文件，再生成一个读取XML文件并使用生成的Proto类将其内的内容转换为二进制的类。之后通过调用所有对应的转换类将所有的数据转换为二进制文件。

（2）数据读取模块，该模块为程序读取数据的模块。为每一个表格都创建一个相应的类对其数据进行读取，并将数据保存起来，在需要是进行获取即可。

（3）游戏内数据管理模块，该模块管理游戏内部的玩家数据。在游戏开始时对对应的数据进行读取，在游戏进行中时对游戏数据进行更新。

* + 1. **角色管理模块**

角色管理模块主要为主角和NPC的控制，包括AI模块和有限状态机模块。有限状态机 （finite state machine)是一种抽象机制，是处在各种不同的预定状态下的其中一种状态。有限状态机也可以定义一组条件，以确认何时应该改变状态。实际的状态会决定状态机的行为[10]。其中的AI模块负责角色的自动控制，让角色能够具备部分的AI行为。借助人工智能,可以增加电子游戏中的玩家挑战,构建不同智能等级的模拟敌人/玩家,从而增加游戏的乐趣[11]。有限状态机模块则负责角色的行为管理，对于角色的所有行为都有一个有限状态机类进行控制，作出对应的反应和处理。

有限状态机可以有效的控制角色的行为，并且将不同的行为处理分离开来。可以更好的区分不同行为所需作出的处理，并对其进行调试。

* + 1. **地图管理模块**

地图管理模块管理地图上的所有地块，以及其状态，以及所有地图上的物品的数据。通过该模块可以快速的获取地图上的物品数据。

* + 1. **UI管理模块**

对于每一个界面的UI都有一个对应的UI类对其和其子节点进行管理。同时对于所有的UI存在一个管理类，管理当前所有打开的UI界面，同时提供UI的生成，关闭，返回等功能。方便外部对UI进行调度。

1. **详细设计**

## 详细设计概述

接下来是游戏中各个模块的详细设计，主要包括具体的代码实现，UI搭建，场景的搭建。

## 事件系统详细设计

事件系统主要包括三个类，事件类型的枚举类、事件的方法即委托类、事件的管理类。

其中事件类型的枚举类包括所有时间类型的枚举，事件方法的委托类为同一个委托的多个重载组成的包含不同参数数目的泛型委托，事件的管理类包括一个存储所有的事件和委托的映射的字典，以及增加，删除，广播事件的方法。在其他语言中可以用函数指针指向一个函数，而在C#中则是可以使用委托作为参数去传递函数，可以将委托理解为一种函数指针，表示某一类型的函数方法[12]。

## 资源管理详细设计

资源管理主要包括两个，一个为资源加载管理，一个为缓存池。

资源加载管理包括一个枚举类ResourceType，表示各种不同的资源类型，对于不同的资源类型去不同的文件夹中进行资源的加载。一个资源缓存的字典，存储资源所在路径和其对应的Prefab的GameObject的映射。以及一个静态方法，输入参数为对应的资源类型，和相对路径名以及是否添加进缓存，该方法返回对应的资源，以供后续进行操作。

另外一个为缓存池，缓存池的作用是当一个物体需要不断的生成又消失时，该操作会占用大量资源，而缓存池是将生成的物体保存起来不进行销毁而是将其隐藏，当需要再次使用时再将其显示出来。缓存池类包括一个缓存池容器poolDic，用于存储缓存池中的同一物品列表和该物品的名字的映射。一个获取缓存池内物品的方法GetObj，该方法传入对应的缓存池物品的名称，找出其中对应的物品，如果未找到则对该物品进行加载并设置其名称。一个放入缓存池的方法PushObj，该方法传入对应的GameObject和其name参数，该方法会判断是否已存在对应的name字典key，如果存在则直接将该物品存入，如果不存在则在字典中添加一个映射。

## **数据管理详细设计**

数据管理模块包括两个部分，一个为其外部工具模块，该模块将Excel文档内的数据转换为二进制，二为内部的数据转换模块。

其中外部工具模块主要包括一个该工具页面管理类XlsxToXmlPanel，该类主要为管理该工具的页面，为该页面每个UI注册对应的事件，页面入图5-1。其中的浏览为选择对应的文件夹，开始转换按钮则会在按下时对所有在表格文件夹中的所有表格的数据进行转换，并将转换的数据保存到XML文件夹和proto文件夹中。



图5-1 表格转换工具页面

文件转换按钮点击后，先判断所选的各文件夹是否存在，如果不存在则不进行下一步转换。然后获取表格文件夹中的所有的后缀名为xlsx的文件对其进行转换。对其中的表格文件首先获取其所有的数据，然后根据获取的数据并利用提前写好的proto模板，将其中对应的字符串进行转换并保存为对应的proto文件，然后使用提前写好的BinaryWriter脚本文件模板，将其中对应的字符串进行转换，并保存为对应的C#脚本文件，用于读取生成的XML文件并使用生成的Proto类将读取的数据转换为二进制文件，最后生成对应的XML文件。

然后则是一个FileHandler类，该类为文件转换类，一为将所有的XML文件全部转换为二进制文件，二为将对应的二进制文件转换为对应的Proto数据。

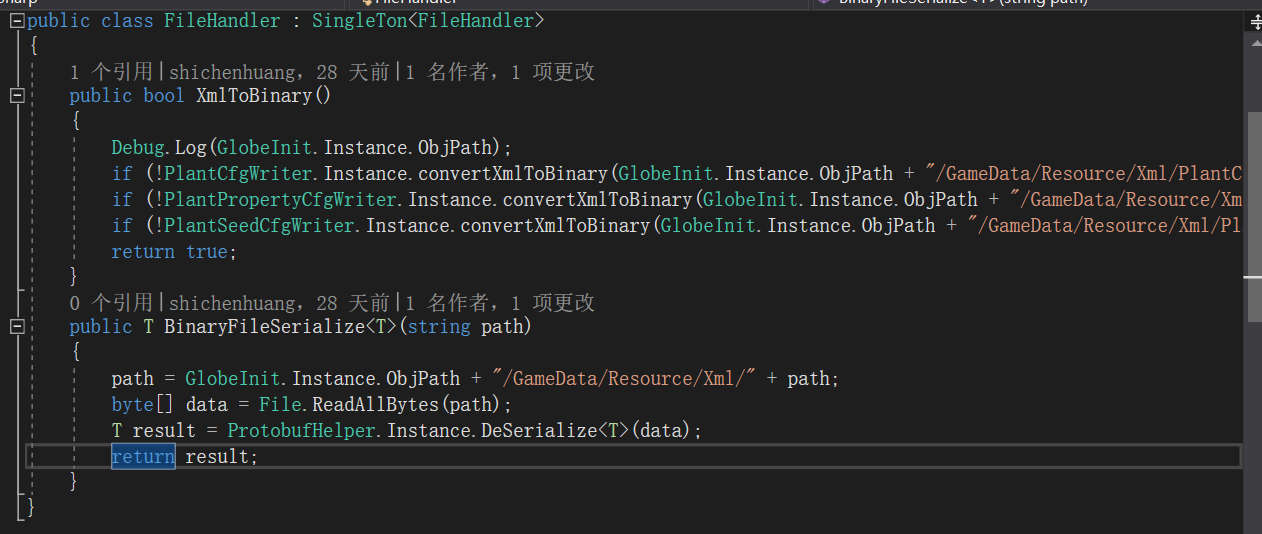


图5-2 FileHandler类

其次为每个表格的读取类，为每一个表格创建一个对应的读取类，将其数据存储起来，对于每一个类，主要包括一个对应表格数据的数据类，以及一个存储对应ID和其数据的映射的字典，以及对应的数据存储处理类，例如通过ID返回对应的数据，或者对应类型的全部数据等接口，以及数据中可能存在的全局变量。

最后为主角，商店，背包，地图创建对应的数据管理类，存储对应的所有数据，对于每个数据管理类在游戏开始时都会进行初始化，在初始化时会将对应二进制文件内的数据转换成对应的数据。对于主角的数据包括走路速度，跑步速度，金币数目，体力值，生命值等数据。地图数据包括地图内所有对应位置的物品，在游戏开始时会读取该数据并生成地图。商店数据管理类，主要为对应的商店的物品数据，包括物品的编号，物品的数目信息，在打开商店是读取数据并生成对应的商品UI，并为其注册对应的点击事件。背包数据则是存储对应的背包位置，对应位置物品ID，对应位置物品数目。在打开背包时使用该数据进行背包UI的创建。

## **角色**系统**详细设计**

角色系统设计分为主角，NPC，AI，有限状态机。

主角设计为按上下左右或WASD进行位置的移动，并且根据当前所选的快捷物品和鼠标点击为判断角色当前的行动，包括吃东西，锄地，浇水，种植，睡觉，送礼，对话。并对对应的行动设计对应的有限状态机类。其中在进行走和跑的动作时将会根据当前按的方向进行移动并设置角色的当前的速度。吃东西则会根据当前所选食物，在吃东西的动作播放完毕后，将物品数目减一，并修改对应的显示UI等信息。锄地会根据当前玩家道具以及主角是否点击到了对应的地块来判断当前是否可以锄地，如果可以则可以锄地，在播放完对应的角色动画后，修改当前选择的地块状态和数据。浇水、种植和锄地的逻辑一样，种植还会判断当前地块是否存在植物，如果存在植物则不进行种植动作，浇水不会判断当前的地块情况，而是直接作出动作并减少体力。睡觉则会恢复当前所有的状态，并保存游戏，让游戏的时间向前推进一天，并且刷新游戏内的商店和植物的状态。送礼物则是判断当前物品，然后判断是否点击了NPC如果点击了NPC则可以对NPC进行送礼提升NPC好感度，解锁商店的商品等NPC功能。

NPC在游戏中主要是沿着预定好的路线进行移动，使用人工智能中的固定位置的寻路方式，让角色在路线上循环移动，表现出一定的人工智能。并设置有特殊NPC固定位置开放商店，当主角点击该NPC时将会读取商店的数据并显示对应的UI，为对应的UI注册对应的点击事件。

有限状态机系统则是包括一个有限状态机的基类，一个有限状态机管理器，并且对应每一个行为有一个有限状态机类继承自有限状态机的基类。

其中有限状态机的基类RoleStateAbstract包括当前角色有限状态机管理器，当前状态机管理器所在角色的动画状态信息两个属性，一个构造函数，以及一个进入该状态时执行的函数OnEnter，一个当在该状态时一直不断执行的函数OnUpdate，和一个退出该状态时的函数OnExit。

有限状态机管理器则包含一个当前角色的角色控制器RoleCtrl类，主要属性包括当前角色的状态RoleState枚举类，当前角色正在进行的有限状态机类，以及对应的角色状态和其有限状态机类的映射字典。主要方法包括一个带有一个参数为角色控制器的构造函数，该构造函数会将所有的对应的状态和对应的有限状态机类映射都添加进去，并设置当前的有限状态机状态；一个OnUpdate方法，用于调用当前有限状态机的OnUpdate类，并且在对应的角色控制的Update方法中调用不断调用从而让当前有限状态机不断更新；一个状态变更方法，再进入该方法时会将前一个状态退出，并且进入新的状态，即调用新状态机的OnEnter方法。

## **UI详细**设计

UI包括一个UI管理类，管理所有UI的显示和关闭，以及每个UI的一个对应panel类。

UI管理类主要包括一个链表存储所有打开的Panel，属性名为m\_LOpenPanel。方法包括打开Panel、关闭Panel、查找Panel，打开时生成对应的Panel并显示，然后存储于m\_LOpenPanel链表中，关闭是先进行查找如果找到则从m\_LOpenPanel中移除，并销毁该Panel，查找则是对所有打开的Panel进行遍历查找。

对于每个UI对应的Panel创建一个脚本用于控制，该脚本继承自UIBase，在Awake时对自身进行初始化，为所有的按钮注册事件，获取部分需要在脚本中需要被修改的UI控件便于调用，在Start中进行显示后的初始化，在Update中进行部分涉及时间的UI显示，并在Destroy之前调用BeforeOnDestroy将需要手动清除的数据进行清理。

1. **结语**

经过对本游戏的开发和研究，首先是对于一个游戏的开发过程有一个基本的认识：从策划，到准备资源，到各项功能的设计以及最终的代码编写。其次则是对于使用Unity更加的得心应手，不论是对Unity基本的API的运用，还有对C#语言的熟悉，使用Unity去读取各种类型的文件，去浏览文件并获取文件的路径等等。同时对于各种类型的文件数据的处理也有了一定的了解，并且可以让其进行相互的转换。而关于委托，泛型，拓展函数也较为熟悉，并且可以运用到代码当中去。在代码编写前一直有进行游戏代码框架的思考，不仅提升了自身的逻辑思维能力，而且对于面向对象的思想和程序模块设计的了解也更加的深入。通过自己去进行骨骼动画的制作以及对模型进行贴图，对于美工方面的辛苦也是深有体会，一个动作的制作，一个模型的贴图都是非常的麻烦且耗时的。同时也对模型的UV展开操作有了一个熟悉的过程，并且对于骨骼的假设和绑定都有了一定的了解。

虽然课题的基本设计已经进行了基本的完善，但是对于除种植和NPC的交互外的需求设计完全没有去进行展开，任然有很大的提升空间。

首先是关于植物的种植地形，目前只有一个普通的地形，之后可以设计出有不同属性的地形，可以加速对应属性的植物的成长，以及始终保持湿润的土壤等。

其次是特殊道具，可以在地面上放置阵法，设计多种不同的阵法，比如改变阵法内部的灵气属性，加速植物生长阵法，自动收获阵法，防止枯萎阵法，自动浇水阵法，四季常青阵法等，然后则是炼丹炉可以将种植出的特殊植物通过炼丹炉炼制成对应的丹药，并且给炼丹过程设计小游戏，让炼丹也变得较为有趣。

然后是植物，目前只有普通的植物，但却没有一些特殊的植物，可以增加植物的属性，并根据植物的属性种植与对应属性的土地上才可让植物顺利成长。并且可以使用种植出的植物去炼制丹药，获取丹药来提高属性。

最后是多人游戏系统，目前游戏还是只支持单人进行游玩，但是现在大部分人任然是喜欢和朋友一起玩游戏，所以可以吸引更多的人去进行游玩。

**致谢**

毕业设计是对整个大学学习的一个总结，也是整个大学交给学校，交给自己的一份结果。在制作过程中遇到了很多的困难，在这里要感谢指导老师吴献，感谢老师对我论文的指导，然后要感谢身边的同学和朋友们，在制作该项目时对资源准备方面提供的帮助，减少了大量的寻找资源的时间，同时还有美工方面知识的教导。

# 参考文献

[1] 叶玲红,谈薛莲.中国风独立游戏中传统文化元素的应用及意义[J].包装工程,2019,40(24):144-149+163.

[2] Goldstone W. Unity Game Development Essentials[M]. Birmingham: Packt Publishing Ltd，2009.

[3] 王小峰,施珍妮.深入理解文本文件和二进制文件[J].科技信息(科学教研),2008(17):392+391.

[4] 黄斌,谢艳新,唐友,李颜甲.数据存储信息序列化完整性及效率评估仿真[J].计算机仿真,2020,37(04):159-163.

[5] 曹宇东. 解谜类独立手游的交互机制实证设计研究[D].华东理工大学,2018.

[6] 杨秋兰.浅谈电子游戏美术设计[J].艺术科技,2014,27(01):75.

[7] 王中州.浅谈CG游戏中角色动作设计的技术实现[J].信息系统工程,2019(09):161.

[8] 王涵.手机游戏UI设计的一般性原则研究[J].卫星电视与宽带多媒体,2019(21):29-30.

[9] 贾颖. 游戏策划理论及方法研究与实现[D].上海交通大学,2009.

[10] David M.Bourg&Glenn Seemann.《AI for Game Developers》[M].Copyright 2004 O’Reilly Media,Inc.0-596-00555-5.2004。

[11] 曹坤泽.人工智能及其在游戏领域中的应用[J].科技传播,2020,12(08):143-144.

[12] 周岚.C#程序设计语言中的委托教学设计与分析[J].电脑编程技巧与维护,2017(23):28-29.

**Design and implementation of the game "I cultivate fields in the world of immortality" based on unity3D**

116062017081 HuangShiChen Tutor:Wu Xian

(School of Mathematics and Information, Fujian Normal University, Fuzhou 350108)

**Abstract:**With the gradual development of today's game industry, the difficulty of individual game development is gradually decreasing, and there are more and more independent game development studios at home and abroad. And with the game stores such as Steam and Epic gradually becoming more familiar to more people in China, buyout games are also gradually accepted by domestic gamers. The domestic games have always been free to play and include internal purchases, and the demand for buyout games is getting higher and higher with the development of the Internet. Then the current game genres are still mostly role-playing and action, while simulation games are relatively niche, but their game viscosity is higher, and players are more inclined to like this type of game. And for domestic players, the immortality genre is more attractive and more suitable for the domestic player base. The game design is a simulation and immortal cultivation independent game using Unity3D engine which can be developed cross-platform. This paper first analyzes and verifies the feasibility of the game according to the current market and its own game characteristics, then outlines the design according to its own design needs, and finally designs the interface and writes the code to realize the whole game.

**Keywords:**Simulation management; independent game; fairy cultivation; Unity3D.