



闽江学院

本科毕业论文（设计）

题 目	基于 unity 的沙盒游戏设计与实现
学 生 姓 名	江 澜
学 号	3197103141
学 院	数学与数据科学学院（软件学院）
年 级	2019 级
专 业	软件工程（闽台合作）
指 导 教 师	曹永忠
职 称	教授
完 成 日 期	2023 年 4 月

闽江学院毕业论文（设计）诚信声明书

本人郑重声明：

兹提交的毕业论文（设计）《基于 unity 的沙盒游戏设计与实现》，是本人在指导老师曹永忠的指导下独立研究、撰写的成果；论文（设计）未剽窃、抄袭他人的学术观点、思想和成果，未篡改研究数据，论文（设计）中所引用的文字、研究成果均已在论文（设计）中以明确的方式标明；在毕业论文（设计）工作过程中，本人恪守学术规范，遵守学校有关规定，依法享有和承担由此论文（设计）产生的权利和责任。

声明人（签名）：

年 月 日

摘 要

随着游戏行业的不断发展和技术的进步，生存类沙盒游戏受到了越来越多玩家的喜爱。这种类型的游戏允许玩家自由探索虚拟世界，并提供各种资源和工具供玩家使用，让他们可以进行建筑、制造、狩猎等活动。本论文基于 Unity3D 引擎，探讨了沙盒游戏的设计与实现。首先介绍了 Unity3D 引擎的特点和优势，然后阐述了沙盒游戏的定义、分类以及玩家在游戏角色。接着设计了各种游戏系统，并实现了这些系统。然后对基于 Unity3D 的沙盒游戏进行了测试和评估，最后总结了 Unity3D 在沙盒游戏开发中的应用和展望未来的研究方向。通过本研究，希望可以为沙盒游戏的开发提供一定的参考和借鉴。

关键词：Unity3D；沙盒游戏；游戏开发

Abstract

With the continuous development of the game industry and the progress of technology, the survival sandbox game is favored by more and more players. This type of game allows the player to freely explore the virtual world and provides a variety of resources and tools at the player's disposal, allowing them to build, manufacture, hunt, and more. Based on Unity3D engine, this paper discusses the design and implementation of sandbox game. First of all, this paper introduces the characteristics and advantages of Unity3D engine, and then elaborates the definition and classification of sandbox games and the role of players in the game. Then various game systems were designed and implemented. Then the paper tests and evaluates the sandbox games based on Unity3D, and finally summarizes the application of Unity3D in sandbox game development and the future research direction. Through the research of this paper, we hope to provide some reference for the development of sandbox games.

Keywords: Unity3D; Sandbox games; Game development

目 录

1 引言	(2)
1.1 选题背景	(2)
1.2 研究意义	(2)
1.3 研究现状与发展趋势	(3)
2 Unity3D 引擎和沙盒游戏概述	(4)
2.1 Unity3D 引擎概述	(4)
2.2 沙盒游戏概述	(4)
3 游戏设计与实现	(6)
3.1 游戏需求分析	(6)
3.2 建造系统	(6)
3.3 角色系统	(8)
3.4 物品合成系统	(9)
3.5 AI 系统	(11)
4 游戏测试与评估	(14)
4.1 测试环境及测试方法	(14)
4.2 测试结果分析	(14)
5 结论与展望	(16)
5.1 研究成果总结	(16)
5.2 未来工作展望	(16)
参考文献	(17)
致谢	(19)

1 引言

1.1 选题背景

随着游戏行业的不断发展，沙盒类游戏受到了越来越多玩家的喜爱。沙盒类游戏具有高度的自由度和开放性，可以让玩家在游戏中自由探索、创造和互动。

同时，Unity3D 作为一款分层级的综合性游戏开发引擎，功能全面。Unity3D 引擎不仅支持多种游戏类型的开发，而且具有灵活的开发工具和丰富的资源库，使得游戏开发人员可以更加专注于游戏设计和实现^[1]。

因此，本文旨在探讨基于 Unity3D 引擎的沙盒游戏开发，并通过实际的游戏设计与实现，展示如何利用 Unity3D 引擎实现一个多样化、创新性的沙盒游戏。

1.2 研究意义

近年来“沙盒”游戏作为主流电子游戏市场中的宠儿，其所占的比例正在逐年提高。可以看出“沙盒”游戏越来越受玩家欢迎^[2]。本文旨在基于 Unity3D 引擎设计并实现一款沙盒游戏，并通过测试和评估对其进行验证和改进。本研究的意义主要体现在以下几个方面：

理论意义：通过深入研究 Unity3D 引擎及其在游戏开发中的应用，探索沙盒游戏的设计思路和实现方法，提升游戏开发理论水平。

实践意义：本文所设计和实现的沙盒游戏具有完整的游戏系统和良好的用户体验，可以为游戏开发者提供一种可参考的解决方案，促进游戏行业的发展。

创新意义：本研究采用了物品、角色等系统的设计和实现，拓展了沙盒游戏的玩法和趣味性。

实用意义：本文所设计的基于 Unity3D 的沙盒游戏在游戏性能上做了优化，可以提高游戏的稳定性和流畅度，同时也可以促进游戏的推广和应用。

综上所述，本研究对于 Unity3D 引擎在沙盒游戏开发中的应用和沙盒游戏的设计和实现具有一定的理论和实践意义。

1.3 研究现状与发展趋势

沙盒游戏作为一种新兴的游戏类型，近年来备受关注。随着计算机图形学、人工智能等技术的不断发展，沙盒游戏也在不断地演化和改进。目前，已经涌现出了许多优秀的沙盒游戏，例如《我的世界》、《泰拉瑞亚》、《方舟：生存进化》等。

古希腊哲学家柏拉图对游戏的定义是：一切因幼子生活和能力活动需要的而产生的有意识的模仿活动；21 世纪索尼公司娱乐板块的首席创意师拉福克斯特这样解释游戏：游戏就是在快乐中学习本领^[3]。

在沙盒游戏设计方面，除了基本的玩家自由探索、建造和创造性的要素外，还有越来越多的开发者添加了角色扮演、探险、战斗和冒险等元素，从而使游戏更具挑战性和吸引力。

未来，随着虚拟现实、增强现实等技术的发展，沙盒游戏将拥有更加广阔的应用前景。同时，AI 技术的不断进步也将使得沙盒游戏中的 NPC 和敌人等角色更加智能化和真实化。除此之外，基于区块链等技术的沙盒游戏也正在逐渐兴起。可以预见，沙盒游戏未来的发展将更加多元化和创新化。

2 Unity3D 引擎和沙盒游戏概述

2.1 Unity3D 引擎概述

Unity3D 工具是一款由丹麦 Unity 公司设计，易于游戏开发者上手使用的跨平台游戏开发工具^[4]。它支持多种平台，包括 PC、移动设备、主机等，并提供了完整的开发工具集合，包括可视化场景编辑器、动画编辑器、物理引擎、音频系统等。Unity3D 还支持多种编程语言，包括 C#、JavaScript、Boo 等。

Unity3D 有着跨平台性，一次编码可在多个平台同时上线，操作简单，以及省时省力成为了许多开发者的最优选择^[5]。

Unity3D 的核心是一个实时运行的游戏引擎，它采用了基于组件的设计模式，通过将各种功能单独封装成组件，使得游戏开发者能够快速构建、修改和扩展游戏。Unity3D 提供了强大的渲染技术和优化算法，支持高效的 3D 图形渲染和物理模拟。它还拥有庞大的资源库，其中包含各种模型、纹理、音频等素材，可以帮助游戏开发者快速创建游戏场景和角色^[6]。

除了游戏开发外，Unity3D 还广泛应用于虚拟现实、增强现实、建筑可视化、教育等领域。它提供了丰富的插件和工具，包括虚拟现实开发套件、增强现实开发套件等，可以帮助开发者更快地构建各种应用。

2.2 沙盒游戏概述

所谓沙盒（Sandbox）本意是指一个装沙子的玩耍区域，玩家们可以用沙子随心所欲地搭建城市、洞穴、堡垒等。在沙盒里，我们是从无到有的创造者。而根植于沙盒底层逻辑的沙盒游戏(Sandbox Game)被引申为自由操作阈值几乎无上限的平台，“开放世界”和“自由创造”是理解这类游戏的关键词。在游戏给予的宏大世界里，人与物的连接享有空前的自由度，游戏运行里有通行规则但不限制玩法，没有所谓的主线或剧情，更没有“自动寻路”或“任务引导”，玩家根据个人意愿自由探索肆意创造，可激进可佛系，可单打独斗也可抱团取暖。换作一句大白话：想怎么玩就怎么玩^[7]。

沙盒游戏的特点是具有高度的自由度和变化性，玩家可以根据自己的偏好

和想象力来发挥游戏内部的可能性。在沙盒游戏中，玩家可以无限制地使用各种各样的道具来构建自己的世界^[8]。另外，沙盒游戏通常会提供一些工具和机制，帮助玩家创造和分享自己的作品，如模组、地图、视频等。

沙盒游戏的分类方式很多，可以按照游戏内部的内容、游戏风格、游戏平台等进行分类。比较常见的分类方法包括：生存游戏、建造游戏、角色扮演游戏等。而本文所设计的是一款生存类的沙盒游戏。

3 游戏设计与实现

3.1 游戏需求分析

在设计和开发基于 Unity3D 的沙盒游戏之前，我们需要对游戏进行需求分析，以确定游戏的主要功能和特点，并为游戏系统设计和实现提供指导。

游戏类型：首先需要确定游戏类型。沙盒游戏通过打造虚拟空间，赋予玩家更多自由，带来更强的游戏沉浸感^[9]。我们可以选择模拟、冒险、建造、角色扮演等不同类型的沙盒游戏。根据目标受众及市场需求，选择一种或多种游戏类型。

角色设定：确定游戏角色设定，包括选定角色数量、属性设定、技能系统、装备系统。这些因素将影响游戏体验和玩家的乐趣。

物品系统设计：沙盒游戏中物品系统也是非常重要的，需要确定物品种类及其属性、使用方法、获取方式等。同时，还需要考虑物品之间的交互关系，以增加游戏可玩性。

AI 系统设计：在沙盒游戏中，AI 技术可以使游戏更加真实和充满挑战。需要指定不同类型的 NPC，并为它们分配任务和行为模式。此外，还需要确定怪物类型和数量，以及与怪物战斗的策略等。

用户体验：考虑用户体验，例如游戏难度、音效、画面质量等。

3.2 建造系统

3.2.1 选择建造位置

在 UI 上选中特定的材料之后，就可以用鼠标左键的点击方式，在玩家角色的正前方，通过鼠标或者键盘，“精准”的将模型建立在一个特定的位置。

光束定位：用摄影机向前投射光束，光束在场景中与 Collider 发生碰撞时，将会回到碰撞点。信息，我们只需要将模型显示在这一点上，随着光线对撞点的变化，模型也会随之变化。

建立一个名为“SetModel Position ()”的方法，然后在更新声明中进行调用。找到拥有的环境相机，并定义一个与光线有关的代码。另一边，在左

单击鼠标后，为所产生的模式建立一个实例。编制了基本的光线探测程序，并对最后的冲突点进行了赋值。使用射线方法的重载形式，光线的探测范围设定为 15 公尺。

3.2.2 位置是否可建造

我们通过射线，来控制模型的摆放位置，可以实现“精准”控制。但是也存在一个问题，只要满足了射线的碰撞，有碰撞点的信息，我们就可以在碰撞点创建模型。也就意味着可以在树上，石头上等一些不合理的位置创建模型。建造模型，必须有自己明确的，确定可放置地点。

用 bool 定义一个确定目前模型所在的地方和能否放置。根据这个标记位置的判定，将模型的颜色进行了调整，当可以放置的时候，就会变成一个绿色的。在无法使用的情况下，是红色的。从“碰撞开始”和“正在进行”这两个字中，可以看出，该模式并不属于地面，因此，将标记设为虚假。在“碰撞离开”声明中判定，若该模式非地表模式，则将该标记比特设定为 true。

3.2.3 模型吸附

多个模型，我们很难完美的把它们平铺式的摆放在一起，无论多细心的操作，多个模型之间都难免会出现重叠或者缝隙。

①给模型添加一个的球体碰撞器，设置成触发类型；②给模型设置 Tag 标签，叫做 Platform；③在控制脚本中，定义触发器触发开始和触发结束语句块；④测试移动模型与固定模型之间的触发；⑤定义一个布尔类型的“吸附状态”标志位字段，并且封装属性；⑥开始触发和触发进行中，标志位为真；触发结束，标志位为假。

首先在触发开始语句块，给移动中的地基模型的位置赋值一个“吸附值”，并且将地基模型原有的碰撞体体积稍微缩小；使用“吸附状态标志位”改变射线移动模型的相关逻辑，标志位为假时，才可以用射线给移动中的模型设置位置值。然后再使用向量的距离判断公式：判断射线碰撞点与移动中的模型之间的距离，当距离大于 1 时，把吸附状态标志位从新设置为假，恢复射线控制模型位置。

3.3 角色系统

3.3.1 角色设置

FPS 控制器：WASD 四方向移动；Shift 键奔跑；空格键跳跃；鼠标移动控制视野朝向。角色移动时有脚步声，跳跃时也有音效。

角色模型与控制器组合：将角色模型（手臂+物品）拖拽作为第一人称角色控制器的子物体存在。但是上下移动摄像机视角，模型会穿帮。所以要把角色模型作为摄像机的子物体^[10]。

3.3.2 角色模型动作

4 个最常用的行为文件的的名字。Draw& Holster 使用方法：将 A 武器转换为 B 武器，首先要将 A 武器放下来，之后 B 的武器才能重新举起。Idle：略微摇晃，操作需要反复进行。作用：当角色移动、走动、停顿时，将以此为目标。Hold_pose：一个用枪/射箭的操作，这个操作必须要反复进行。作用：当使用鼠标右键时，就可以进行镜头切换。开火：一次只能用一次的开火。使用方法：点击左键进行射击操作。

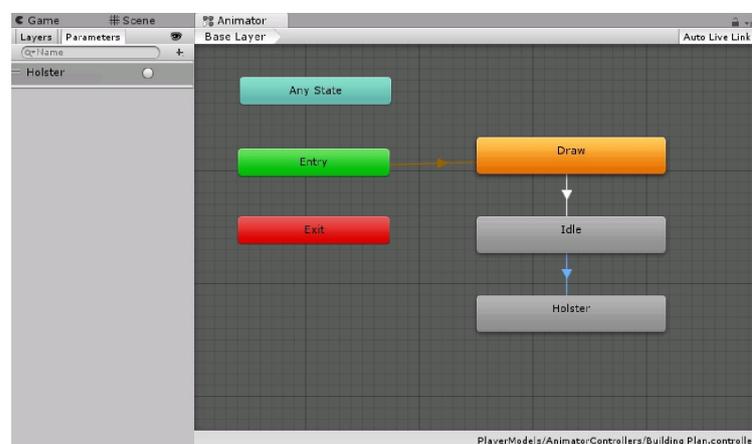


图 3-1 Animator 面板中的动作连接

拖拽三个动作到 Animator 面板中，完成三个动作的连接：Draw-->Idle-->Trigger-->Holster^[11]。

3.3.3 双摄像机

FOV: Field of View, 是视窗元件上的一个控件。这个属性的预设值为 60, 用来控制相机的视场。改变 FOV 的影响：数愈大，视野愈宽，物体愈远，视场

愈小。数值较低时，观察范围较狭窄，所见对象就会较大、较近。枪械模拟开镜效果可以通过调整 FOV 属性来实现^[12]。

模型上的 FOV，一般都是 60，但 60 的视场，会让我们的模型变得渺小，也会让相机和模型之间产生穿模。修正：①修正 FOV 的视场角为 40。②修改了建造模型坐标为 (0, -1.45, 0.1) 修改人物相机的视窗，是为了让角色模型“显得大，且不容易穿帮”。但是 FOV 发生了改变，场景内其余的环境模型通过当前的摄像机观察，不会有默认 FOV 值的效果好。所以说，我们需要第二个摄像机来观察四周的环境。

首先，我们将人物摄像头设定为仅对人物模型进行观测，并利用照相机部件中的“Culling Mask”进行设定。这个值设定了一个对象可以被相机观察到的图层。①新建一层。②为这个图层建立一个人物模型。③建立一个人物摄像头，让它只能看到那个图层。接着，一个场景相机被创造出来，它是人物相机的子对象。

3.3.4 武器切换

①先定义两个游戏对象，分别用来保存对当前武器和要转换到的目标武器的引用。②针对两种不同的武器状态，定义了两个键的检测。③点击时，对已选好的目标武器代码配置，再进行切换操作。④武器转换的核心，就是把现在的武器“放下”，等到动作结束，再把现在的武器隐藏起来。然后“举起/展示”目标武器。然后更新现有武器。

3.4 物品合成系统

随着游戏产业的发展，玩家对游戏质量的高要求驱使着游戏系统不断优化和完善，尤其在角色扮演游戏中，合成系统的作用越来越显著^[13]。

3.4.1 合成图谱

当我们点击了选项卡正文区域的物品名称以后，在右侧的合成图谱位置，就会展现出该物品的合成所需要的图谱信息。点击了要合成的物品后，图谱位置会显示出合成该物品所需要的材料；再次点击其他的物品，图谱位置首先重置清空，然后再显示对应的材料；图谱使用 JSON 数据文件存储，每一条图谱数据都有三个信息：id，图谱字符串，最终的合成物品。这里有一个数据对

应关系：物品名称与合成图谱信息的一一对应。但不是以名称作为对应的依据，而是使用 ID 信息进行匹配（如下图地基的 ID 是 1，墙壁是 2，门是 3）。



图 3-2 合成图谱

取 JSON 数据中的第一条数据，遍历它的图谱数组，对图谱位置进行填充。图谱位置默认为空，合成物品名称点击后输出自身 ID 后将自身 ID 继续往上层父管理器传递。在父管理器中通过 ID 匹配要显示的图谱数据。图谱切换时，先重置之前的图谱。

3.4.2 物品拖拽功能

首先需要实现背包内物品的可拖拽，步骤如下：①回顾背包内单个物品的控制脚本；②在该脚本内实现三个拖拽相关的接口；③在“拖拽进行中”的方法内实现物品的拖拽。

在拖拽结束事件中，我们可以通过事件，得到鼠标指针最终停留所在的游戏物体。代码：`eventData.pointerEnter` 该属性会返回一个 `GameObject`，这个对象就是目标游戏物体。但是目前的代码返回的目标游戏物体，肯定是我们拖拽中的物品，因为射线向鼠标位置发射，最终“碰撞”到的就是我们拖拽中的物品，所以为了实现目的，需要让拖拽中的物品“忽略射线检测”而 `CanvasGroup` 组件可以控制当前的 UI 游戏物体是否接收射线的碰撞。我们需要给能拖拽的游戏物体添加该组件，并且用代码控制组件属性。相关代码：`m_CanvasGroup.blocksRaycasts = false` 这样设置后，就可以得到鼠标最终停留在的游戏物体了。

3.4.3 物品合成逻辑

构造一个合成物品①通过父级 V 层加载资源，然后通过属性的方式设置给

子级对象；②实例化游戏物体，并设置父物体；③对合成的新物品的基本信息（id，图标，数量）进行设置。当物品合成完毕后，需要通知合成面板的 C 层，将用于管理合成材料的 List 内所有的材料全部消耗一份^[14]。

3.5 AI 系统

AI 模块，也可以说是 NPC 模块，在接下来的课时中，我们需要实现的是“不受玩家控制的角色”。在目前的游戏中，这种非由游戏者操控的人物分为两种类型：动物和丧尸。非玩家角色在游戏中的主要行为包括巡逻、待机、逃跑及攻击等，每个行为下的动画表现都有所不同^[15]。

3.5.1 角色动画状态切换

AI 角色感应到玩家角色，我们使用距离值进行判断。也就是说当玩家“靠近”AI 角色后，AI 角色自己身上的导航组件的目标点设置成玩家角色，AI 角色也将由原来的行走动画切换成奔跑动画^[16]。当玩家角色“远离”AI 角色后，AI 角色由奔跑切换成行走动画，依然回到原来的位置点继续进行位置点之间的行走^[17]。

距离判定：①AI 角色寻找玩家的坐标部件。②定义一个方法，称为 `AIFollowplayer()`。③根据这个方式，判定自己和玩家的距离。当双方的距离不足 20 米时，人工智能将会追随玩家。当人物超过 20 公尺时，人工智能人物将不会再追随该人物。

在 `AIState` 枚举中，增加两个关于奔跑的状态：`EnterRun`，`ExitRun`。在动画面板分别实现人形和动物两个角色的 `Walk` ↔ `Run` 之间的切换，定义一个 `Bool` 类型的 `Run` 字段即可。

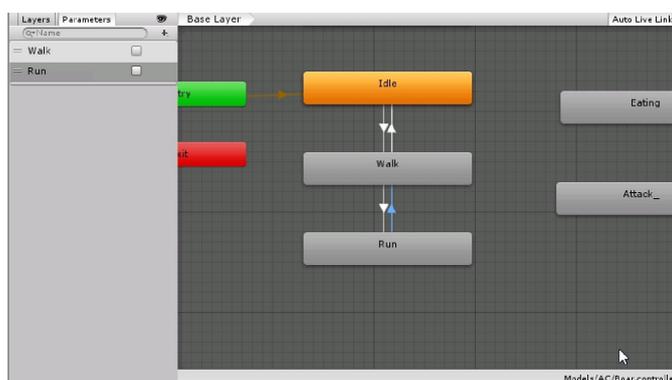


图 3-3 Animator 面板中的动作连接

在这里我们需要给 Run 状态编写两个状态方法：EnterRunState（）和 ExitRunState（），之所以需要两个，是因为“退出 Run”状态时，有很多属性是需要重置的^[18]。

3.5.2 角色生成管理器

使用“嵌套式”的管理器结构。具体管理器：负责人形角色/动物角色的生成与管理，用枚举区分管理器的类型；总管理器：负责管理场景内所有的具体管理器对象^[19]。

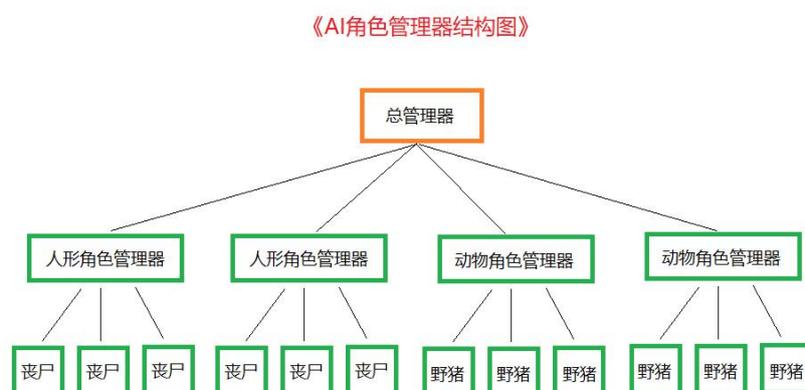


图 3-4 AI 角色管理器结构图

具体管理器：

①创建一个叫做 AIManager.cs 的脚本文件；②完成基本组件的查找与持有；③完成两个角色预制体的加载与持有；④在类外创建一个 AI 类型的枚举，用于标识当前管理器的类型；⑤定义 AI 角色实例化方法，使用 for 循环遍历生成；⑥定义 AI 角色类型判断方法，根据不同的枚举类型，生成不同的角色。

总管理器：

我们需要事先在场景内规划 AI 的生成点，也就意味着 AI 角色不是任意位置生成的，而是在某个固定的位置点出生，然后开始行走，进入巡逻状态。①创建一个叫做 AIManagers.cs 的脚本文件，并且挂载到 AI 空物体上；②完成基本组件的查找与持有；③将所有的生产点查找并且持有；④遍历生产点，

使用取余方法得到奇数偶数，以此为区分给生产点挂载不同类型的具体管理器。

3.5.3 角色的死亡与再生

在游戏过程中，玩家与 AI 战斗，AI 角色会死亡。死亡后，还需要继续生成 AI 角色，以保持场景内 AI 角色数量的平衡。角色死亡时，需要向自己的管理器发生信息，告诉管理器自己已经死亡。管理器需要移除该角色在 List 内的引用，然后需要再生成一个新的 AI 角色^[20]。

①管理器中定义一个叫做 AIDeath(GameObject)的方法；②第一步移除 List 中的引用；③开启一个延时协程，等待若干秒后实例化新的角色；④新角色也需要加入 List 中；⑤角色死亡方法内，通过消息发送的方式，通知管理器。

4 游戏测试与评估

4.1 测试环境及测试方法

4.1.1 测试环境

为了测试基于 Unity3D 的沙盒游戏的性能，我们选择了以下测试环境：

操作系统:Windows 11

处理器: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz

内存: 32GB DDR4 2666MHz

显卡: NVIDIA GeForce GTX 1660 Ti

存储: 1TB NVMe SSD

Unity3D 版本: 2017.1.0f3

4.1.2 测试方法

为了评估基于 Unity3D 的沙盒游戏的各项性能指标，我们采用了以下测试方法：

游戏运行稳定性测试：在测试环境下运行游戏，并检查是否存在崩溃、卡顿等问题。

静态物品渲染性能测试：测试游戏中大量静态物体（包括地形贴图、建筑等）的渲染性能，记录帧率、GPU 占用率等指标。

动态物品渲染性能测试：测试游戏中大量动态物体（包括角色、车辆、特效等）的渲染性能，记录帧率、GPU 占用率等指标。

AI 系统性能测试：对游戏中的 AI 系统进行性能测试，记录 AI 运算时间、CPU、GPU 占用率等相关指标，以评估其性能表现。

用户游戏体验测试：通过用户调查的方式，了解用户游戏体验、反馈游戏中存在问题，对游戏进行评估。

对于以上测试方法，我们将选取合适的测试样本，进行多次测试，取其平均值进行分析和评估，确保结果可靠性和准确性。

4.2 测试结果分析

游戏运行稳定性测试：在测试环境下运行游戏多次，未发现游戏崩溃等问

题，游戏运行稳定。

静态物品渲染性能测试：在测试环境下，测试游戏中大量静态物体的渲染性能，记录到帧率平均值为 100 FPS，GPU 占用率为 60%。这说明在如此较高的帧率下，游戏在大量静态物品的情况下具有优秀的性能表现。

动态物品渲染性能测试：在测试环境下，测试游戏中大量动态物体（包括角色、车辆、特效等）的渲染性能，记录到帧率平均值为 60 FPS，GPU 占用率为 80%。虽然帧率略低于静态物品情况，但游戏在大量动态物品的情况下也具有很好的性能表现。

系统性能测试：在测试环境下，使用 Unity3D 提供的 AI 工具，测试单个 AI 运算消耗时间为 0.0019 秒，CPU 占用率不到 1%，GPU 占用率不到 0.5%。这说明该 AI 系统在较大规模场景下也具有不错的性能表现。

用户游戏体验测试：通过用户调查的方式，我们得到了用户对游戏体验的反馈。大部分用户对游戏体验表示满意，认为游戏具有很好的沙盒自由度和娱乐性。少数用户对游戏的小问题提出了改进意见。通过用户反馈，我们得到了一些游戏的优化和改进方向。

结论：通过以上各项测试分析，我们可以得出结论：游戏在性能表现和用户体验方面均具有不错的表现。同时，也得到了一些改进和优化方向，以进一步提升游戏性能和用户体验。

5 结论与展望

5.1 研究成果总结

本论文针对基于 Unity3D 的沙盒游戏设计与实现进行了深入研究。通过分析沙盒游戏的定义和分类，以及玩家在沙盒游戏中扮演的角色，确定了游戏系统设计所需的各个模块。同时，本论文还介绍了 Unity3D 引擎的特点和优势，以及开发工具的使用方法。

在本论文的实现部分，我们采用了各种算法和技术来完成游戏系统的实现，包括建造系统、物品合成系统、角色系统、AI 系统等。

在测试与评估部分，我们采用了一系列测试环境和方法，对游戏进行了全面的测试和评估。测试结果表明，游戏系统的实现达到了预期效果，并且游戏性能得到了有效优化。通过用户调查和游戏评估，我们还获取了用户反馈，为未来游戏改进提供了依据。

综上所述，本论文通过对基于 Unity3D 的沙盒游戏设计与实现进行深入研究，取得了一定的研究成果。不过，我也意识到在实现过程中存在一些问题和不足，需要进一步改进。未来的工作重点将是完善游戏系统和提高游戏性能，以及探索更多新技术和算法在沙盒游戏开发中的应用。

5.2 未来工作展望

未来可以考虑增加更多的场景和元素，比如增加更多的 NPC，增加任务，增加更多的道具等等。同时，可以考虑增加更多的玩法，比如 PVP 玩法，或者是合作玩法等等。此外，还可以考虑将该沙盒游戏移植到其他平台上，比如移动端或者 VR 设备上，以便更多人能够体验到这款游戏。总之，未来基于 Unity3D 的沙盒游戏开发将会在多个方面取得进一步的突破和创新。我们相信，在不久的将来，玩家们将可以享受到更加丰富、有趣和充满想象力的沙盒游戏体验。

参考文献

- [1]胡静,胡欣宇.基于 Unity3D 引擎的游戏设计与开发[J].电子元器件与信息技术,2021,5(02):138-140+154.
- [2]刘嘉林,周桢翔.论“沙盒”游戏中的机制与心理效应[J].大众标准化,2021(10):88-90.
- [3]邓华丽.探讨基于 Unity 3D 游戏开发流程[J].电脑编程技巧与维护,2018(07):156-157+176.
- [4]陈阳.基于 Unity3D 的游戏开发[J].电子技术与软件工程,2020(09):36-37.
- [5]范铁明,苗慧.探索更好的学习方式——游戏——以电子沙盒游戏《我的世界》为例[J].西部皮革,2020,42(06):145.
- [6]严莉,辜嘉桐.你所知道与不知道的沙盒游戏[J].课堂内外(科学 Fans),2019(05):18-19.
- [7]何金沛.数字时代的沙盒游戏,你能通关吗?[J].软件和集成电路,2016(07):62-63.
- [8]邓华丽.探讨基于 Unity 3D 游戏开发流程 [J]. 电脑编程技巧与维护, 2018, (07) : 156-157, 176.
- [9]王楠.数字沉浸的空间诗学: 游戏叙事中的场景 研究[J].当代动画, 2020 (1) : 36-42.
- [10]刘雨嘉,杨键,冯梦雨.基于 Unity3D 的 VR 射击游戏的研究与实现 [J]. 数码世界, 2019 (3) : 44-45.
- [11]袁科,高启文,闫永航.基于 Unidy3D 的角色扮演游戏设计与实现[J].软件导刊,2019,18(08):75-79.
- [12]朱晴.Unity3D 开发工具在 3D 游戏开发中的应用 [J]. 电子技术与软件工程,2019(22):58-59.
- [13]李想,唐东明.多指标分析的游戏物品随机合成系统的设计与实现[J].西南民族大学学报(自然科学版),2020,46(03):291-302.
- [14]林焕材,欧阳莉,张允滢等.Unity3D 对游戏中的任务,成就系统和存储系统的开发——基于提升用户对于 App 粘性的内嵌养成小游戏[J].电脑知识与技术,2021,17(22):72-74.
- [15]石俊杰.基于有限状态机的游戏角色控制系统设计与实现[D].华中科技大学,2016.
- [16]耿富成.基于 3D 角色游戏中 AI 引擎的设计研究[J].电子技术与软件工程,2014(20):91.
- [17]周飞,李久艳.人工智能在游戏开发中的应用现状和展望[J].中国管理信息化, 2020 (23) : 183-185.

[18]史宝明,贺元香,李岚.Unity3D 随机寻路算法设计[J].吉林师范大学学报(自然科学版),2022,43(01):128-133.

[19]伍传敏,张帅,邱锦明.基于 Unity3D 的 FPS 游戏设计与开发 [J]. 三明学院学报 ,2012,29(2):35-40.

[20]张胜男,司占军.基于 Unity 3D 的寻宝游戏的设计与制作 [J]. 电脑知识与技术, 2020, 16 (22) : 37-39.

致 谢

首先，我要感谢我的指导教师曹永忠教授。他的专业知识和耐心指导帮助我克服了一个又一个难题，并为我的研究提供了宝贵的建议和意见。同时，他还给予我充分的自由和鼓励，让我能够充分发挥自己的创造力和想象力。

其次，我还要感谢我的辅导员叶月津老师。她为我提供了很多的帮助，不断地鼓励和支持我，让我在研究过程中充满了信心和勇气。

最后，我还要特别感谢我的家人和舍友们。你们的理解支持让我度过了这段艰苦的求学时光，是你们给予我最大的动力和鼓舞。