

# 基于 Unity3D 的单机竞技游戏设计与开发\*

沈旭<sup>1</sup> 梁进大<sup>1</sup>(岭南师范学院 信息工程学院 湛江 524048)<sup>1</sup>

**摘要** 利用 Unity3D 引擎和 C# 开发一款竞技类单机游戏, 主要实现地图模块、控制模块、有限状态机模块、技能模块、属性模块、建筑物模块、士兵模块、伤害模块、防御模块、升级模块、金币模块、装备模块、恢复模块、计时模块、音效模块和辅助模块功能。可以通过键盘和鼠标来控制场景中的角色移动、攻击、释放技能和购买商品等操作。用户在游戏中可以体验到竞技游戏带来的主动性、操作性, 以及丰富的战斗方式、高质量的画面、音效和逼真的游戏体验。

**关键词** Unity3D C# Windows 平台 游戏

**中图分类号** TP301

**文献标识码** A

**文章编号** 6511

## Design and Development of Single Computer Competitive Game Based on Unity3D

SHEN Xu<sup>1</sup> LIANG Jinda<sup>1</sup>(Information Engineering School, Lingnan Normal University Zhanjiang 524048)<sup>1</sup>

**Abstract** Using Unity3D engine and C# to develop a competitive single game, it mainly realizes map module, control module, finite state machine module, skill module, attribute module, building module, soldier module, injury module, defense module, upgrade module, gold coin module, equipment module, recovery module, timing module, sound. The function of the effect module and the auxiliary module. Keyboard and mouse can be used to control the movement, attack, release skills and purchase of goods in the scene. In the game, users can experience the initiative and maneuverability of competitive games, as well as a rich way of fighting, high quality pictures, sound effects and realistic game experience.

**Keywords** Unity3D C# Windows platform Game

### 一、引言

Unity3D 是由丹麦 Unity 公司开发的一款全面整合的专业游戏引擎, 让开发者轻松创作的多平台游戏开发工具, 主要目的是用于开发 3D 游戏<sup>[1]</sup>。具体包含整合的编辑器、跨平台发布、地形编辑、着色器、碰撞器、脚本、网格、物理、UGUI 框架、版本控制等特性, 这些功能可以为开发者节省大量开发时间。Unity3D 得到许多开发者的支持, 不仅界面简洁友好、功能容易上手以及个人版免费提供给开发者, 而且还能将开发的游戏发布到多平台<sup>[2]</sup>。目前, Unity3D 游戏引擎在国内外游戏界均占主导地位, 掌握其操作以及功能, 对今后研发游戏和追求游戏品质具有很大意义。

### 二、系统设计分析

#### 1. 功能需求

本设计是一款单机竞技类游戏, 用户可以通过键盘和鼠标来控制场景中的角色, 实现移动、攻击、释放技能和购买商品等操作, 其主要包括地图模块、控制模块、有限状态

机模块、技能模块、属性模块、建筑物模块、士兵模块、伤害模块、防御模块、等级升级模块、金币模块、装备模块、恢复模块、计时模块和音效模块 15 个主要功能模块以及辅助模块。总体功能需求如图 1 所示:

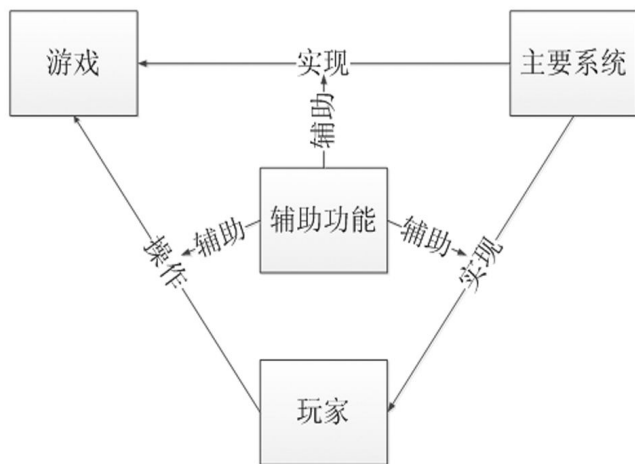


图 1 系统总体功能需求图

\* 基金项目: 沙画艺术的数据场图像分析与演化关键算法研究(61402399)、广东省哲学社会科学规划项目(GD17XGL33)、湛江市财政资金科技专项(2014A01010)。

## 2. 性能需求

(1) 可玩性: 对于一款游戏, 最吸引用户的就是可玩性, 用户对游戏的评价莫过于这款游戏好不好玩或者值不值得玩等等, 本课题开发的游戏是模仿现在竞技类中最流行的游戏: 英雄联盟和王者荣耀, 这两款游戏的战斗方式和其他玩法比较新颖, 而且特别考验用户的操作, 可玩性特别高<sup>[3]</sup>。

(2) 操作性: 整个游戏的玩法不是单纯地用键盘或者用鼠标操作, 而是键盘和鼠标一起使用, 这样才能让用户感受到该游戏的操作性, 对角色的操作不是一成不变的, 而是灵活地去改变操作方式从而产生不同的玩法。

(3) 流畅性: 流畅性是每款游戏的一个重要标准, 一个游戏必须保持运行流畅, 没有用户希望在游戏高潮的时候出现卡顿, 所以这方面必须从美术资源、代码等方面进行优化, 从而保证游戏的流畅性<sup>[4-5]</sup>。

(4) 平衡性: 一款游戏平衡也是很重要的, 用户不希望自己太过强大, 这样玩游戏没意思, 更不希望敌人太过强大, 这样玩不下去。所以一款游戏的平衡性是格外的重要, 其主要体现在“数值”方面, 必须保持双方数值的平衡, 本课题设计的游戏主要参考了英雄联盟和王者荣耀的各类“数值”, 进行微调从而保证整个游戏的平衡性。

## 3. 系统模块结构图

本游戏系统其主要包括地图模块、控制模块、有限状态机模块、技能模块、属性模块、建筑物模块、士兵模块、伤害模块、防御模块、等级升级模块、金币模块、装备模块、恢复模块、计时模块、音效模块和辅助模块。游戏系统模块功能图如图2所示。

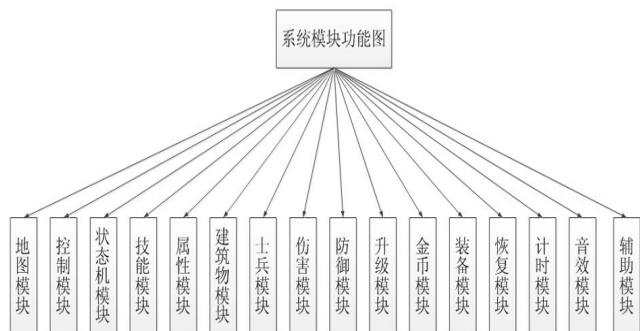


图2 系统模块功能图

## 三、游戏模块实现

### 1. 地图模块

地图分为蓝红双方（用户为蓝方），对称分布有血泉（英雄出生、复活以及恢复生命值和魔法值的区域）、三条

路（分上、中、下路）、野区、水晶、防御塔和其它场景装饰，如图3所示。



图3 地图模块

### 2. 控制模块

(1) 鼠标左键: 某些技能选择色目标用左键确定, 鼠标右键: 用户控制英雄移动; (2) A: 让用户确定攻击目标, 再用鼠标左键选定目标。若选定地点, 则英雄会在移动到该地点过程中, 只要在英雄可攻击范围内则会自动攻击敌方单位; 若选定单位, 则英雄会移动到该单位并发动攻击; (3) Q/W/E/R: 英雄技能施放; (4) Ctrl+Q/W/E/R 或直接点击各技能的升级技能按钮: 升级对应技能等级; (5) D/F: 召唤师技能施放; (6) B: 回城

鼠标左右键操作关键代码:

```
if (Input.GetMouseButtonDown(0))// 左击, 确定目标
{
    if (currentCursorIndex == 3 && currentRangeIndex == 0)//
    如果为攻击范围
    {
        MouseEvent(hitInfo);

        if (IsAttackTargetTag(target, new int[] { 0, 1, 2 } ))// 如
        果射线碰撞到的物体 Tag 等于攻击目标 Tag
        {
            AttackTarget(target); // 攻击目标
        }
    }
    else if (currentCursorIndex == 3 && currentRangeIndex ==
    1)// 如果为选择目标鼠标图标, 并且当前显示的范围时 E
    技能范围
    {
        MouseEvent(hitInfo);

        if (IsAttackTargetTag(target, new int[] { 0, 1 } ))// 如果
        射线碰撞到的物体 Tag 等于攻击目标 Tag
```

```

{
    attackTarget = target.gameObject; // 设定攻击目标
    heroSkillE.allowReleaseE = true; // 允许释放技能
}
}
else if (Input.GetMouseButtonDown(1)) // 右击, 到指定位置
{
    MouseEvent(hitInfo);
    if (IsAttackTargetTag(target, new int[] { 0, 1, 2 })) // 如果射线碰撞到的物体 Tag 等于攻击目标 Tag
    {
        AttackTarget(target); // 攻击目标
    }
    else
    {
        #region 显示移动的目标位置
        targetPos.gameObject.SetActive(true); // 显示
        targetPos.transform.position = hitInfo.point; // 显示的位置
        targetPos.DORestart(); // 播放动画
        #endregion
        attackTarget = null; // 设攻击目标为空
        heroAttack.allowAttack = false; // 不允许攻击
        damageCollider.allowTrigger = false; // 不允许触发
    }
}
}

```

### 3. 有限状态机模块

英雄动画状态机: 死亡动画、回城动画、空闲动画、跑步动画、技能动画、暴击动画、普通攻击动画以对应的关系和条件连接起来。士兵动画状态机: 普通攻击动画、跑步动画以对应的关系和条件连接起来。

游戏中的模型和模型动画是美工做好发到网上提供下载的, 所以这里主要展示同一个模型的不同动画之间是如何关联起来的; Unity3D 游戏引擎提供几种切换动画的触发条件, 分别有 Float、Int、Bool 和 Trigger, Float 和 Int 只要满足开发者填写的设定值即可触发, Bool 表示以 True 和 False 触发, Trigger 表示开发者用填写指定的字符串作为条件值来触发; 如图 5-3 所示, 是英雄模型和士兵模型所有动画的关联图, 动画与动画间的切换都是有方向的, 代表是否可以从该动画切换到另一个动画, 双向的则表示这两个动画可以互相切换。

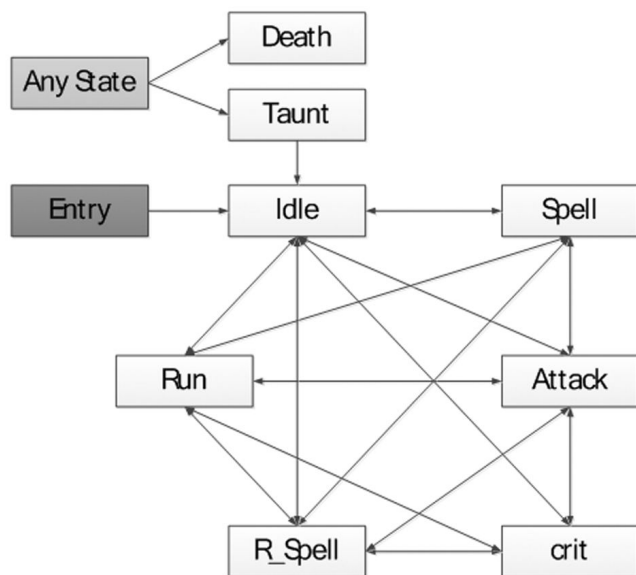


图 4 英雄有限状态机

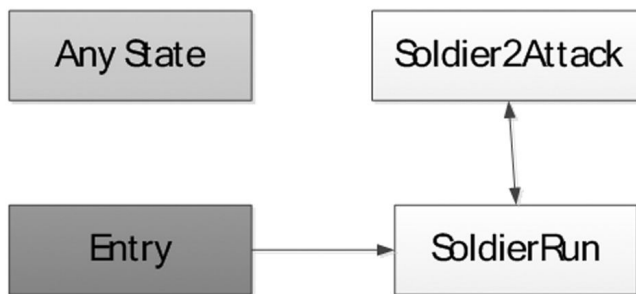


图 5 士兵有限状态机

### 4. 技能模块

(1) 英雄技能 (英雄: 赵子龙, 除被动技能外皆可升级):

① 被动技能 - 战个痛快: 赵子龙的普通攻击和 E 技能 - 无畏冲锋会对目标发起挑战, 减少当前攻击目标 15% 的护甲值, 持续 3 秒;

② Q 技能 - 三重爪击: 赵子龙的 3 次普通攻击能够对当前攻击目标造成 15/30/45/60/75 (+1.2\* 赵子龙总攻击力) 物理伤害, 减少除该技能以外的技能 1 秒的冷却时间, 并且释放该技能在有效时间内的第三次普通攻击将会把当前目标击飞。消耗: 40 法力。冷却时间: 9/8/7/6/5 秒;

③ W 技能 - 狂战怒吼: 被动: 每第三次普通攻击会使赵子龙对当前攻击目标进行暴击, 造成 25/37.5/50/62.5/75% 额外伤害。每命中三次敌人, 赵子龙就会治疗自身 30/35/40/45/50 (+0.4\* 赵子龙总法术强度) 生命值。主动: 提高自身 40/45/50/55/60% 的攻击速度, 持续 5 秒。消耗: 30 法力。冷却时间: 16/15/14/13/12 秒;

④ E 技能 - 无畏冲锋: 赵子龙对一名攻击目标发起冲锋与挑战。冲锋会对当前攻击目标及其技能有效范围内的

所有敌人造成 70/110/150/190/230 (+0.6\* 赵子龙总法术强度)魔法伤害,并减少他们的 25/30/35/40/45%移动速度,持续 2 秒。消耗:60 法力。冷却时间:12 秒;

⑤R 技能-新月横扫:赵子龙将长枪挥舞一圈,对周围的敌人造成 75/175/275(+1\* 赵子龙总攻击力)加上当前攻击目标 15%生命值的物理伤害,并在击退他们的同时使赵子龙长枪触碰到的所有敌人晕眩(对小兵最多造成 600 伤害)。战个痛快:如果一个被挑战的目标中了该技能,那么该目标将不受击退效果影响。消耗:100 法力。冷却时间:120/110/100 秒。

(2)召唤师技能(不可升级):

①D 技能-闪现:使英雄朝着你的指针所停的区域瞬间传送一小段距离。消耗:0 法力。冷却时间:300 秒;

②F 技能-幽灵疾步:增加英雄移动速度增加 27%,持续 10 秒。消耗:0 法力。冷却时间:210 秒;

③B 技能-回城:8 秒后将英雄传送回召唤师平台,任何伤害技能都会打断技能施放;消耗:0 法力。冷却时间:0 秒。

最后是显示每个技能效果、消耗以及冷却时间所有内容和部分技能的范围,随着技能的升级和英雄自身属性的改变而改变具体显示的内容,用鼠标移动到相应的技能图标上即可显示,如图 5-3 所示。



图6 技能信息界面

关键代码实现:

/// 释放 Q 技能

```
public void ReleaseSkillQ()
```

```
{
```

```
int currentMp = heroPro.hero.CurrentMp;
```

```
int consumeMp = (int)heroPro.GetSkillQInfo(GetSkillInfo-  
Type.Consume);
```

```
if (currentMp > consumeMp)// 如果够法力值
```

```
{
```

```
heroPro.hero.CurrentMp -= consumeMp;// 扣法力值
```

```
skillBtns[0].interactable = false;// 不允许点击
```

```
isAddAttackTimes = true;// 开始累加
```

```
}
```

```
}
```

/// 释放 B 技能

```
public void ReleaseSkillB()
```

```
{
```

```
StopMove();
```

```
teleportHomeTime = 8.0f;// 设置回城时间
```

```
teleportHomeTimer.gameObject.SetActive(true);
```

```
anim.TeleportHomeAnim();// 回城动画
```

```
UpdateTeleportHomeTimer();// 更新回城时间
```

```
}
```

```
private void UpdateTeleportHomeTimer()
```

```
{
```

```
teleportHomeTimerText.text = new StringBuilder().Append-  
Format("回城 {0} 秒", teleportHomeTime.ToString("0.0")).
```

```
ToString();
```

```
teleportHomeTimer.value = teleportHomeTime / (float)8;
```

```
teleportHomeTime -= 0.1f;// 减 0.1 秒
```

```
if (teleportHomeTime > 0)
```

```
{
```

```
Invoke("UpdateTeleportHomeTimer", 0.1f);// 0.1 秒更新
```

```
}
```

```
else
```

```
{
```

```
heroPro.ResetHeroPosition();// 重置英雄位置
```

```
CancelTeleportHome();// 取消回城
```

```
}
```

```
}
```

/// 取消回城

```
private void CancelTeleportHome()
```

```
{
```

```
CancelInvoke("UpdateTeleportHomeTimer");// 取消计时
```

```
teleportHomeTimer.gameObject.SetActive(false);
```

```
teleportHomeTime = 8;// 重置回城时间
```

```
anim.IdleAnim(true);// 回城动画
```

```
}
```

## 5. 属性模块

生命值: 生命值为零时作为死亡或者消失的标准; 魔法值: 释放某些技能需要消耗魔法值; 攻击力: 增加普通攻击或物理伤害技能的伤害值; 法术强度: 增加魔法伤害技能的伤害值; 物理抗性: 该属性值表示减少自身所受物理伤害(普通攻击或物理伤害的技能)指定的百分比伤害; 魔法抗性: 该属性值表示减少自身所受魔法伤害(魔法伤害的技能)指定的百分比伤害; 攻击速度: 英雄、士兵或建筑物的攻击速度; 冷却: 减少技能释放的间隔时间; 暴击几率: 暴击会产生普通攻击两倍的伤害, 该属性是表示触发暴击的几率; 移动速度: 英雄或士兵在地图上的移动速度<sup>[6-7]</sup>。

如图 7 所示, 英雄模型头上和技能下方为英雄的生命值和魔法值, 英雄头像左方显示的图标和数值从左往右, 从下往上分别为英雄的攻击力、法术强度、物理抗性、魔法抗性、攻击速度、冷却、暴击几率和移动速度属性; 小兵和建筑物模型上为其血量没有魔法值以及鼠标移动他们身上时屏幕左上方会显示它们的攻击力、法术强度、物理抗性、魔法抗性、攻击速度、冷却、暴击几率和移动速度属性。



图 7 游戏属性信息界面

## 6. 建筑物模块

防御塔: 拥有较高生命值、较高伤害、高防御、较高攻击速和大攻击范围的属性, 地图上三条路都分别布置三座防御塔, 根据与水晶的距离不同而属性不同, 其主要职责是保护水晶; 水晶: 拥有超高生命值, 超高防御属性, 蓝红双方各布置一座, 其存亡作为游戏胜负的标志。如图 8 所示, 防御塔与水晶建筑物。

## 7. 士兵模块

近程兵: 拥有低生命值、低伤害、高攻速和小攻击范围属性; 魔法兵: 拥有较低生命值、较低伤害、较低攻速和中等攻击范围属性; 炮兵: 拥有高生命值、高伤害、中等攻速和大攻击范围属性<sup>[8]</sup>。



图 8 防御塔与水晶建筑物

近程兵和魔法兵为开局 5 秒后产生第一波, 然后每隔 40 秒产生一波, 炮兵为每三波普通兵(近程兵和魔法兵)派上场后会产生一个; 如果没有炮兵情况下, 每条路会产生三个近程兵和三个魔法兵, 如果有炮兵情况下, 每条路会产生三个近程兵和两个魔法兵以及一个炮兵, 如图 9 所示。



图 9 近程兵、魔法兵与炮兵

## 8. 伤害模块

英雄随着等级上升或者购买装备决定攻击力和法术强度值, 相应等级有对应的攻击力和法术强度加成, 装备也有固定的属性值加成; 建筑物和士兵根据其不同的攻击方式(指近程兵、魔法兵和炮兵)和攻击范围决定不同的伤害值, 并且在游戏中都是预先固定的值, 不受其它因素影响。

## 9. 防御模块

英雄、建筑物和士兵都会有不同的防御值, 包括: (1) 物理抗性: 减少自身所受物理伤害(普通攻击或物理伤害的技能)指定的百分比伤害, 再进行扣除攻击目标生命值; (2) 魔法抗性: 减少自身所受魔法伤害(魔法伤害的技能)指定的百分比伤害, 再进行扣除攻击目标生命值。此外, 英雄有技能可以较少其指定防御类型的值。

## 10. 升级模块

建筑物和士兵消失后可增加英雄的经验值, 但英雄必须在指定范围内并且是未死亡状态; 英雄每级的所需经验

值随着等级提高而增加;英雄升级后,会获得各属性(生命值、魔法值、攻击力、法术强度、物理抗性、魔法抗性、攻击速度)的增加,并且会获得一个技能点,一个技能点可以升级一个英雄技能(R技能为大招,需要英雄等级为6级或者以上才可升级)。如图10所示,英雄模型上方生命值左方以及英雄头像下方显示的数字是英雄等级,而英雄头像下方的等级右方为经验条。



图10 等级与升级

### 11. 金币模块

游戏每秒增加1金币,金币用来购买装备,使英雄属性得以增强。属于英雄杀死的士兵或者建筑物(即士兵或者建筑消失前最后的伤害来源属于英雄)都会增加用户相应的金币。购买装备后,会扣除装备相应价格的金币;同时,出售装备也能得到装备原价百分之四十的金币。如图11所示,右下方金币图标右边则是游戏中的金币。



图11 金币

### 12. 装备模块

装备,是增加英雄属性的另一种方式,代价是需要金币,增加属性越高则价格越高,用户可以根据游戏中经济情况进行选择购买,同时也可以撤销购买(即撤销购买并返回全部金币)。

商店,主要功能是购买和撤销购买装备,其次是显示可购买的装备以及相应装备的详细信息;并且英雄必须在指定的范围内,否则无法购买和撤销购买装备。

背包系统,顾名思义就是装载物体的系统,具体是显示已购买的装备的区域,让用户知道已购买了哪些装备。

出售,如果用户差点钱买更高级的装备,可以出售相对差点的装备获得其原价的百分之四十的金币,再去购买其他装备。

如图12所示,点击金币处会弹出商店界面,用户点击相应装备即可查看装备的详细信息,双击某个装备或者选中某个装备再点击购买即可购买到对应的装备;在右下方的装备栏(即背包系统)显示已购买的装备,点击装备栏里某个装备会弹出出售界面,具体根据用户需求进行操作即可。



图12 装备

### 13. 恢复模块

血池恢复(英雄出生地):用户可通过回城或操作英雄回到血池有效范围内,即可提升英雄生命值和魔法值的恢复速度,速度为每秒200个单位;

自动恢复:只要英雄处于非死亡状态,在地图上任何位置,其生命值和魔法值每秒恢复1个单位。

以上两种情况的前提是英雄受到了伤害和释放过需要消耗魔法值的技能,即生命值和魔法值都不是最大值的前提下,英雄才可恢复。

### 14. 计时模块

计时模块记录游戏进行的时间,提高用户积极性争取用最短的时间去结束游戏。如图13所示,右上角有个计时提示,用户可以通过查看该提示了解游戏进行的总时间。

### 15. 音效模块

背景音乐:游戏持续进行中的背景音乐;

触发音乐:游戏中某种条件触发的音乐,例如:英雄复活音乐等。



图 13 游戏计时

#### 16. 辅助模块

各大系统模块有密切的联系,例如 UI 界面需要获取相关信息进行显示,摄像机的控制,用于直接升级的功能等辅助功能。如图 14 所示,英雄头像上方为直接升级按钮,用于调试英雄技能和等级升级后所提升属性的数值是否平衡等。



图 14 直接升级功能

### 四、结束语

早期由于计算机硬件的限制,游戏画面上还没有 3D 的概念,仅停留在 2D 的显示状态,随着个人电脑配置的提升和游戏开发的进一步完善,用户开始对电脑游戏的画面产生了更为苛刻的要求,以往的 2D 画面已远远不能满足这些挑剔的游戏用户的眼光,所以这时候很多 3D 游戏引擎应运而生。90 年代初国外 3D 游戏已经开始起步,第一款 3D 第一人称射击游戏是美国的卡马克的《重返德军总部》<sup>[9]</sup>。国内 3D 游戏起步于 2003 年,较于国外的 3D 游戏开发起步要晚很多,无论是从游戏开发理念上还是技术支持上都还有不可逾越的差距<sup>[10]</sup>。本文介绍了基于 Unity3D 游戏引擎设计开发一款竞技类单机游戏的全过程。可以通过键盘和鼠标来控制场景中的角色移动、攻击、释放技能和购买商品

等操作,在游戏的过程中可以体验到竞技类游戏带来的主动性、操作性和真实感,以及更加丰富的战斗方式和高质量的画面以及生动的音效。希望本文能对 3D 游戏开发者起到借鉴的作用。

#### 参考文献

- [1]朱玉.基于 Unity3D 漫游系统的设计与研究[J].信息安全与技术,2014,(12):78~81.
  - [2]蒋元,蔡胜军,何绍勇,祁玮.基于 Unity3D 的虚拟地理环境构建[J].现代计算机(专业版),2014,(33):52~55.
  - [3]J.G.Liu.Applied Research of Somatosensory Game Based on Kinect and Unity 3D Data Integration Technology[J].Applied Mechanics and Materials,2014,667:177~182.
  - [4]刘孟全.Unity3D 虚拟现实平台的重放功能设计[J].广西科学院学报,2013,(1):34~36.
  - [5]何岗,李俊,张锋.基于 Unity3D 的消防 3D 情景实训系统的设计与实现[J].软件产业与工程,2014,(6):43~48.
  - [6]赵治羽,陈倩云,李婷婷,张娟.基于遗传算法的 Unity3D 装箱系统设计[J].计算机与现代化,2014,(6):141~144+149.
  - [7]熊耀.基于 Unity3D 粒子系统的三维影视特效开发研究[J].软件导刊,2012,(11):134~136.
  - [8]张典华,陈一民,李磊.基于 Unity3D 的多平台三维空战游戏的开发[J].计算机技术与发展,2014,(1):192~195.
  - [9]陈豪.基于 Unity3D 的 Android 飞行类游戏设计[J].电脑知识与技术,2015,(30):168~170.
  - [10]Cheng Z, Dan H.Virtual Campus Based on Unity3D[J].Advanced Materials Research,2014,1049~1050:1856~1859.
- 作者简介**
- 沈旭(1979-),男,汉,山东单县,讲师,硕士,CCF 会员(66155M).研究方向:嵌入式系统、物联网、Android 开发等。
- 梁进大(1995-),男,汉,广东茂名人,本科生,研究兴趣:移动应用开发。