

A circular wreath of various botanical illustrations, including green ferns, red and orange flowers, and large green leaves, framing a central white circle.

《计算机游戏开发》

近期教学安排

近期教学安排



期末大作业要求&评判标准



《计算机游戏开发》

实验2 点评



实验2 完成情况

截至2024.4.29





助教反馈信息

Positive：主要亮点在于道具的使用和UI。大部分都是增加音效以及数值化优化。

Negative：

1. 部分同学态度不够端正，有时间但未完成实验基本要求
2. 小部分学生未完成新增关卡部分，请仔细阅读实验要求，例如要求添加第二关能够体现难度，而不是在原有的基础上增加难度即可
3. 在实验结论和代码解释部分，如果使用gpt对代码逐行解释显得报告较为冗长。
4. 注意提交的实验附件是否出错




```

35
36 // 获取摇杆当前方向
37 Joystick_dir Joystick::getDirection()
38 {
39     if ((m_currentPoint - m_centerPoint).x > 0)
40     {
41         return Joystick_dir::_RIGHT;
42     }
43     if ((m_currentPoint - m_centerPoint).x < 0)
44     {
45         return Joystick_dir::_LEFT;
46     }
47     return Joystick_dir::_STOP;
48 }

```

将代码进行修改，添加上下左右移动功能。

在 Joystick_dir.h 中的 Joystick_dir 中添加_UP 和_DOWN。

```

8
9 enum Joystick_dir
10 {
11     _LEFT,
12     _RIGHT,
13     _STOP,
14     _UP,
15     _DOWN
16 };

```

将 Joystick.cpp 代码进行修改：

m_currentPoint 代表当前摇杆的位置，m_centerPoint 代表摇杆的中心位置。

m_direction 计算了当前位置与中心位置之间的差异，即摇杆的移动向量。

通过比较移动向量的 x 和 y 方向上的距离，判断哪个方向的移动距离更大。

如果 y 方向的移动距离更大，则根据 y 值的正负来确定向上还是向下移动。

如果 x 方向的移动距离更大，则根据 x 值的正负来确定向右还是向左移动。

如果移动距离相等或者没有移动，则返回 Joystick_dir::_STOP，表示停止状态。

```

57 Joystick_dir Joystick::getDirection()
58 {
59     auto m_direction = m_currentPoint - m_centerPoint;
60     // 比较x和y方向的移动距离的绝对值
61     int distanceX = abs(m_direction.x);
62     int distanceY = abs(m_direction.y);
63
64     // 判断移动距离更大的方向
65     if (distanceY > distanceX)
66     {
67         if (m_direction.y > 0)
68         {
69             return Joystick_dir::_UP;
70         }
71         else if (m_direction.y < 0)
72         {
73             return Joystick_dir::_DOWN;
74         }
75     }
76     else if (distanceX > 0)
77     {
78         if (m_direction.x > 0)
79         {
80             return Joystick_dir::_RIGHT;
81         }
82         else if (m_direction.x < 0)
83         {
84             return Joystick_dir::_LEFT;
85         }
86     }
87
88     return Joystick_dir::_STOP;

```

修改 HelloWorldScene.cpp 中控制角色移动代码。

代码实现了角色的移动控制。根据摇杆的方向和角色的当前位置，决定角色是否需要向特定方向移动，并且根据限制条件进行位置更新。

首先，使用 m_joystick->getDirection() 获取摇杆的方向。

如果摇杆方向为向右 (Joystick_dir::_RIGHT)，并且角色的当前位置加上角色宽度的一半小于或等于 680，那么角色将向右移动。

如果摇杆方向为向左 (Joystick_dir::_LEFT)，并且角色的当前位置减去角色宽度的一半大于或等于 200，那么角色将向左移动。

如果摇杆方向为向上 (Joystick_dir::_UP)，并且角色的当前位置加上角色高度的一半小于或等于 480，那么角色将向上移动。

如果摇杆方向为向下 (Joystick_dir::_DOWN)，并且角色的当前位置减去角色高度的一半大于或等于 0，那么角色将向下移动。

如果摇杆方向为停止 (Joystick_dir::_STOP)，则不需要进行任何操作，角色保持当前位置不变。



实验2 优秀作品视频





(UI美观、有交互效果、完成度高) 2021152027韦旭南.mkv



(关卡策划有新意、游戏完成度高、报告撰写优秀) 王曦_卡布不卡.mp4



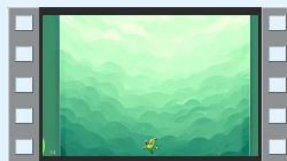
(使用ai生成开始页面) 2021152020_吴迪_实验二.mp4



(隐藏摇杆) 实验2 演示视频.mp4



(有创意、游戏完成度高、美观) 2021152036 刘涌铎.mp4



2021110101冯滨麟演示视频.mp4



2021150028_蔺傲雪.mp4



2021150065-石烁.mp4



2021190005 曹智杰.mp4



2021190063_许明浩.mp4



2021190065_邵奕达.mp4



2021192009_谢钺.mp4



2021280270_黄杰.mp4



叶子扬-1: 06开始为通关, 前面是死亡测试.mp4



实验2 优秀实验报告





2021192010 王曦 游戏 实验2.doc

8. 实验心得

8.1 核心心得

- (1) 做游戏最花时间的部分不是写代码，而是策划、制作音乐和美术资源、测试。
- (2) cocos-2dx 不好用。
- (3) 好像敲代码敲出肌腱炎了。

8.2 其它心得

本次实验中，我通过学习 Cocos2d-x 游戏引擎的使用，深入了解了交互界面设计的原理和方法，并掌握了 Cocos2d-x 中的用户交互、触摸事件、碰撞检测等机制。

通过本次实验，我不仅学会了交互界面设计的基本原理和方法，还掌握了使用 Cocos2d-x 进行游戏开发的基本技能。我相信这些知识和技能对我今后的游戏开发之路会有很大的帮助，我会继续努力学习，不断提升自己的技术水平，成为一名优秀的游戏开发工程师。

指导教师批阅意见：

成绩评定： $15+5+5+5+10+5+5+5+5+37-1=97$

评语：UI 界面较美观。游戏优化方向较新颖。内容详实，实验报告较规范。

四、 实验结论或心得体会

使用工厂方法进行场景创造，并且将各界面使用 layer 封装后，各个功能的实现会变得比较简单，代码间耦合性降低，会比较容易维护。

指导教师批阅意见：

成绩评定： $15+5+5+5+9+5+5+5+5+39=98$

评语：内容详实，实验报告较规范。整体 UI 界面契合游戏。在报告中可见同学的编程实践能力，从设计模式上进行代码的优化。游戏优化点较多。



2021192009_谢钺（代码封装思想）.docx



谢谢

