南开大学 计算机大类

姓名：谭煜

学号：2413075

班级：3205

2025年5月17日

高级语言程序设计

实验报告

**目录**

* [一、实验项目名称](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec1)
* [二、开发环境](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec2)
* [三、项目要求](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec3)
* [四、主要流程](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec4)
  + [1. 整体流程](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec4-1)
  + [2. 核心算法或逻辑](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec4-2)
    - [（1）地图切换与过渡效果](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec4-2-1)
    - [（2）玩家移动与碰撞检测](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec4-2-2)
    - [（3）子弹发射与反弹](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec4-2-3)
    - [（4）游戏状态管理与结束条件](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec4-2-4)
  + [3. 单元测试](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec4-3)
* [五、项目成果](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec5)
  + [1. 游戏运行效果](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec5-1)
  + [2. 代码结构与可维护性](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec5-2)
* [六、收获与体会](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec6)
  + [1. 面向对象编程的应用](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec6-1)
  + [2. 游戏开发的基本流程](https://statics.moonshot.cn/kimi-web-seo/runner.html#sec6-2)

高级语言程序设计大作业实验报告

**一、实验项目名称**

派果大乱斗

**二、开发环境**

Visual Studio 2022

**三、项目要求**

* 面向对象：使用面向对象的编程思想，通过类的继承和多态实现不同地图、场景以及玩家等功能。
* 单元测试：对关键功能进行单元测试，确保代码的正确性和稳定性。
* 模型部分验证：通过实际运行游戏，验证游戏逻辑、角色行为、地图交互等功能是否符合设计预期。

**四、主要流程**

**1. 整体流程**

* 游戏初始化：加载游戏所需的资源，包括地图、按钮、玩家角色等的图片资源，以及背景音乐等音频资源。
* 场景管理：通过场景管理器（SceneManager）来切换不同的游戏场景，如菜单场景（MenuScene）和游戏场景（GameScene）。每个场景都具有自己的进入（on\_enter）、更新（on\_update）、绘制（on\_draw）、输入处理（on\_input）和退出（on\_exit）等方法。
* 地图管理：使用地图管理器（MapManager）来管理不同的地图（如Map1和Map2），实现地图的切换以及地图特有的进入和退出逻辑。
* 玩家控制与交互：玩家类（player）负责处理玩家的移动、射击、反弹（Rebound）等操作，以及与地图的碰撞检测、生命值管理等功能。玩家之间可以通过发射子弹进行攻击，子弹类（bullet）负责子弹的移动、检测是否击中玩家等逻辑。
* 游戏运行与结束：游戏在主循环中不断更新和绘制，根据玩家的操作和游戏逻辑进行状态变化。当玩家生命值为零或掉出游戏区域时，游戏结束，显示获胜玩家的动画和提示信息。

**2. 核心算法或逻辑**

**（1）地图切换与过渡效果**

* 在地图管理器中，通过调用不同地图的on\_enter和on\_exit方法来实现地图的切换。在切换过程中，使用渐变的透明度效果来实现平滑的过渡。例如，在Map1的on\_enter方法中，通过逐渐增加地图图片的透明度来显示地图，而在on\_exit方法中则逐渐降低透明度直到完全隐藏。
* 渐变透明度的计算公式为：alpha = 255 \* (delay - timer) / delay，其中alpha表示当前的透明度值，delay表示过渡的总时间，timer表示当前已过去的时间。

**（2）玩家移动与碰撞检测**

* 玩家的移动通过键盘输入控制，根据玩家的输入方向和速度来更新玩家的位置。在移动过程中，需要检测玩家是否与地图的边界或其他障碍物发生碰撞。
* 碰撞检测采用矩形碰撞检测算法，将玩家和地图中的障碍物都视为矩形区域，通过比较矩形的边界来判断是否发生碰撞。例如，在Map1的on\_update方法中，检测玩家是否与地图底部发生碰撞，从而实现玩家的落地和反弹效果。
* 碰撞检测的条件为：if (player.x <= bx + bw && player.x >= bx && player.y - player.r + player.vy <= by + bh && player.y > by)，其中player.x和player.y表示玩家的当前位置，bx、by、bw和bh表示障碍物矩形的边界坐标和宽高，player.r表示玩家的半径，player.vy表示玩家在垂直方向上的速度。

**（3）子弹发射与反弹**

* 玩家可以通过按键发射子弹，子弹的发射方向根据玩家的朝向而定。子弹类负责子弹的移动和检测是否击中目标玩家。
* 子弹的移动通过简单的位移计算实现，根据子弹的速度和方向更新子弹的位置：b\_x -= bullet\_speed（向左移动）或b\_x += bullet\_speed（向右移动）。
* 子弹的反弹逻辑通过玩家的反弹技能实现。当玩家处于反弹状态时，检测子弹是否进入玩家的反弹范围，如果进入，则改变子弹的方向并增加子弹的速度：bullet\_dir = temp ? false : true（改变方向），bullet\_speed += 20（增加速度）。
* 子弹是否击中玩家的检测采用圆形碰撞检测算法，计算子弹与玩家之间的距离，如果距离小于子弹和玩家的半径之和，则认为子弹击中玩家：if (distance(b.b\_x, b.b\_y, player.x, player.y) <= b.plus\_b\_r + player.r)。

**（4）游戏状态管理与结束条件**

* 游戏的状态通过全局变量（如gamerunning、gameover等）来控制。在游戏主循环中，根据这些状态变量来决定游戏的运行流程。
* 游戏结束的条件是玩家生命值为零或掉出游戏区域。当游戏结束时，显示获胜玩家的动画和提示信息，并等待玩家选择重新开始或退出游戏。

**3. 单元测试**

* 玩家移动测试：测试玩家在不同输入情况下是否能够正确地上下左右移动，以及在地图边界处是否能够正确地停止移动。
* 子弹发射与移动测试：测试玩家发射子弹后，子弹是否能够按照正确的方向和速度移动，并且在超出游戏区域后是否能够被正确地销毁。
* 碰撞检测测试：测试玩家与地图障碍物之间的碰撞检测是否准确，包括玩家与地图底部的碰撞、玩家与地图侧面的碰撞等场景。
* 子弹击中玩家测试：测试子弹是否能够在正确的时机击中玩家，并且玩家的生命值是否能够正确地减少。
* 反弹技能测试：测试玩家的反弹技能是否能够正确地触发，以及子弹在反弹后的方向和速度是否符合预期。

**五、项目成果**

**1. 游戏运行效果**

* 游戏能够正常启动，并且在菜单场景中可以切换地图、开始游戏或退出游戏。
* 在游戏场景中，两名玩家可以通过键盘操作进行移动、跳跃、射击等操作，并且能够看到对方玩家的实时动作。
* 地图的切换效果平滑，玩家在不同地图之间切换时不会出现明显的卡顿或异常。
* 子弹的发射、移动和击中效果符合预期，玩家的生命值能够正确地根据被击中的次数减少。
* 当玩家生命值为零或掉出游戏区域时，游戏能够正确地结束，并且显示获胜玩家的动画和提示信息。

**2. 代码结构与可维护性**

* 项目代码采用了面向对象的编程思想，通过类的继承和多态实现了功能的扩展和复用。例如，地图类（Map）作为基类，Map1和Map2作为派生类，继承了基类的通用方法，并且实现了各自特有的进入、退出、更新和渲染逻辑。
* 代码的结构清晰，每个类和函数都有明确的职责，便于阅读和维护。例如，玩家类（player）中包含了玩家的所有属性和方法，如移动、射击、碰撞检测等，使得玩家相关的逻辑集中在一个地方，方便后续的修改和扩展。
* 在代码中添加了详细的注释，对每个类、函数以及关键的代码段进行了说明，帮助其他开发者更好地理解代码的实现逻辑。

**六、收获与体会**

**1. 面向对象编程的应用**

* 通过本次项目，深入理解了面向对象编程的核心概念，如类的继承、多态、封装等，并且学会了如何在实际项目中运用这些概念来设计和实现功能。例如，通过定义抽象的基类Map和Scene，以及派生类Map1、Map2、MenuScene和GameScene，实现了不同地图和场景的切换和管理，提高了代码的复用性和可扩展性。
* 学会了如何使用虚函数和虚析构函数来实现多态，使得可以通过基类指针调用派生类的方法，从而实现不同的功能。例如，在地图管理器中，通过基类指针currentMap调用不同地图的on\_enter、on\_exit等方法，实现了地图的切换逻辑。

**2. 游戏开发的基本流程**

* 了解了游戏开发的基本流程，包括游戏设计、资源准备、代码实现、测试和优化等阶段。在项目中，首先根据需求设计了游戏的玩法和功能，然后准备了所需的图片、音频等资源，接着通过编写代码实现了游戏的逻辑和功能，最后进行了测试和优化，确保游戏能够正常运行并且具有良好的用户体验。
* 学会了如何使用图形库（如EasyX）来实现游戏的绘制和交互功能，包括加载和显示图片、绘制图形、处理键盘和鼠标输入等。通过实践，掌握了图形库的基本使用方法和一些常用的图形绘制技巧。