

设计报告

【问题描述】

怪物大战是基于 qt 平台开发的一款游戏。程序一共使用了 5700 行代码，全部为自己原创，编写了 30 个类。游戏核心玩法是通过控制飞碟，利用地图上的各种方块，躲避地刺，同时使用随机道具来击败所有怪物，获得游戏胜利。怪物大战有十分精致的贴图，细腻动画和生动的音效。支持玩家自己编写地图，保存地图文件和读取地图文件，也可以使用内置的默认地图进行游戏，整体游戏的完成度很高。

【系统设计说明】

(1) 数据结构说明：使用了数组，向量，向量的迭代器，指针

(2) 程序功能说明。包括：

各模块功能：游戏的模块有：游戏框架模块，玩家模块，怪物模块，地图模块，窗口模块，地图编辑与文件读写模块，按钮与音乐模块，特效模块，具体为：

游戏框架模块：

MSprite：精灵类，大多数对象的基类，主要控制贴图，碰撞检测和移动

MAnimationSprite：动画精灵，精灵的继承类，主要提供多种动画播放模式

MDefine：宏定义类，管理程序的各种宏定义

玩家模块：

MBullet：子弹类，控制子弹的飞行与撞击

MPlayerTop：玩家头部，控制子弹的发射方向和指令

MPlayerBottom：玩家地图，控制玩家的移动

MPlayer：玩家整体，控制玩家的碰撞检测

怪物模块：

MMonster：怪物类，控制怪物移动，攻击和记录各专项属性

MMonsterArray：用于统一管理怪物，提供默认怪物分布

地图模块：

MCube：方块的基类，主要控制方块的碰撞检测

MBrick：砖块类，控制子弹碰撞后的消失

MIce：冰块类，控制子弹碰撞后子弹和冰块的消失

MJelly：果冻类，控制子弹碰撞后的反弹和不同的果冻反弹动画

MSpikerock：地刺类，用于攻击触碰的玩家

MMap：地图类，用于管理所有方块和提供默认地图

窗口模块：

MStartWindow：用于创建开始菜单，提供菜单功能

MainWindow：游戏主窗口，用于运行处理各个类的功能，提供游戏界面

MEndWindow：结束窗口，由于提供游戏结束以后的菜单

地图编辑与文件读写模块：

MMapEditor：地图编辑类，提供窗口和按钮以供玩家自定义地图和保存地图

MFileReader：用于支持地图文件读取

按钮与音乐模块：

MMusicButton：使鼠标进入按钮时可以播放声音

MMoveButton：用于支持地图编辑中按钮的移动

MMusicPlayer: 用于管理音乐资源和提供接口

特效类:

MProp: 保存道具的各种属性

MPropArray: 用于统一管理道具, 控制道具的出现与信息传递

MMedicine: 血量包类, 用于在相应道具触发后增加玩家血量

MAccelerator: 加速器类, 用于在相应的道具触发后增加玩家速度

MFrozen: 冰冻类, 用于在相应道具触发后冰冻怪物

MBomb: 炸弹类, 用于在相应道具触发后产生爆炸特效

MSpecial: 特效类, 用于统一管理道具触发后的特效

算法描述: 核心算法主要有如下几个:

1. 角度算法, 通过诱导公式, 将窗口坐标系统一为标准平面坐标系, 然后利用解析几何的相关知识, 计算出目标两点连线与 y 负轴的夹角。
2. 直线移动算法, 给定角度, 利用两套坐标分别存储, 实现高精度的直线移动
3. 反弹算法, 在果冻类中, 根据目标点与碰撞物的相对位置和和目标点的飞行方向来计算反弹后的方向。

接口说明:

1. 开始菜单提供三个按钮接口: 创建新地图, 读取地图存档, 默认地图。点击读取地图存档以后, 有文件名输入接口, 和确认与返回按钮。
2. 在地图编辑窗口, 左侧有六个方块按钮, 用于创建方块。右侧有文件名输入接口, 保存, 开始, 返回, 重置四个按钮。
3. 在游戏界面, 左侧有主菜单, 暂停, 重置三个按钮, 在游戏地图, 可以读取相应的鼠标和键盘信息, **WASD** 控制玩家移动, **R** 发射子弹, 鼠标控制发射方向。
4. 在游戏结束窗口, 有重置和退出两个按钮。
5. 每一个类中的数据成员都有对应的接口函数, 同时还有一些功能函数可供使用, 由于程序的类太多, 就不做展开介绍了。

关键变量说明。

变量太多, 只能着重说几个列子

1. **MMapEditor** 中的 **MMoveButton** 的指针 **moveButton**, 该指针用于指向鼠标正在移动的方块。
2. **MainWindow** 中的 **int** 型的 **gameMode** 变量, 用于记录游戏的打开模式, 以保证游戏可以被正常地重新开始。
3. **MainWindow** 中的 **musicPlayer** (**MMusicPlayer** 实例化), **player** (**MPlayer** 实例化), **gameMap** (**MMap** 实例化), **monsterArray** (**MMonsterArray** 实例化), **startWindow** (**MStartWindow** 实例化), **mapEditor** (**MMapEditor** 实例化), **endWindow** (**MEndWindow** 实例化), **fileReader** (**MFileReader** 实例化), **propArray** (**MPropArray** 实例化), **special** (**MSpecial** 实例化)

【实现环境】

- (1) 设计平台要求: 装有 **Qt5**, **Visual Studio**
- (2) 运行环境要求: **window** 系统, 显示器 **1920*1080** 分辨率最佳

【设计实现及分析】

程序设计和调试过程出现的问题及解决方法，程序测试方法和数据等

1. 问题：直线运动时产生较大误差。解决方法：由于是因为坐标是 `int`，而每一帧对象移动的距离较小，若直接舍弃小数部分会使得后续移动的误差较大。于是采用两套变量来记录坐标，一套用 `int`，一套有 `double`，每一帧用 `double` 来记录结果，再转化为整型，这样可以后续的移动误差在一个像素以内。
2. 问题：用户加载的地图文件可能不是合法的地图文件。解决方法：在写入地图文件时在文件开头写入一段密码，读取文件时检查密码，如果密码不符合则弹出警告。
3. 问题：编辑地图中移动方块时，由于方块一直位于鼠标之下，鼠标事件被传递给子控件导致方块移动不流畅。解决方法：在子控件中重写鼠标事件
4. 程序测试方法：debug，由于是游戏，没有直接的数据记录

【讨论】

程序的评价，包括程序结构、时空效率、调试方法等

游戏的整体完成度很高，贴图和动画都很完整，同时配有很完整的音效。同时游戏的扩展性良好，模块分类详细，接口函数齐全，后期新增功能十分便捷。为了保证良好的拓展性，程序的结构不够精简，时空效率不够高，值得后期逐步优化。由于是第一次使用 Qt 进行这种较大项目的编程，程序没有很好地体现 Qt 编程的特性，信号和槽的使用还不够流畅，后期学习和理解更深一步以后可以逐步改良。

【系统使用说明】

由于是使用 Qt 编程，需要很多 Qt 提供的文件，所以请在 **Release** 文件夹中运行 **exe** 文件。同时，游戏的声音文件也被放置在了 **Release** 文件夹。

如果要重新编译文件，请确保电脑已安装 **Qt5**，否则无法顺利编译。请确保 **UIDocuments** 文件夹和 **Music** 文件夹在项目文件夹之内，否则程序将没有贴图和声音。

小结

这次大作业耗时较久，前期主要用来学习 Qt 和理解 Qt 编程的逻辑。在实际编程过程中，也遇到了很多困难，通过不断地查找资料和反复调试，终是都被解决。这一次编程，是自己第一次独立编写这个大的项目，很好地锻炼了自己的编程能力。