C语言程序设计大作业

所选题目名称

单 位： 计算机学院网络空间安全学院

班 级： 计算机科学与技术四班

学 号： 202205566927

姓 名： 吴畅

任课教师： 谢勇

湘 潭 大 学

2023年6月

## 文档信息

|  |  |  |  |
| --- | --- | --- | --- |
| 源代码 | 可执行文件 | 自评成绩 | 答辩PPT |
| 有 | 有 | 优 | 有 |

## 学术诚信承诺书

xxx，学号xxxxxxxxxx，系信息工程学院计算机系xxxx级xxxxx专业x班学生。本人承诺本课程设计提交的所有源代码和文档不存在抄袭、剽窃行为。如果被他人发现存在抄袭、剽窃、雷同等学术不端行为，愿意承担所带来的一切后果。

签名、时间照片图片

## 题目概述

1. 概述

简要介绍本题目的内容，目标。

1. 所要完成的任务说明

|  |  |  |
| --- | --- | --- |
| 序号 | 功能分类 | 任务 |
| 1 | 必做 | 地图滚动展示，方向控制，敌我子弹发射，碰撞检测等基本射击游戏元素 |
| 2 | 必做 | 游戏存盘和读盘功能 |
| 3 | 必做 | 成绩排行榜 |
| 4 | 选做 | 游戏音乐与音效 |
| 5 | 选做 | 敌人被击落的粒子特效 |

1. 完成情况

|  |  |
| --- | --- |
| 功能序号 | 完成情况说明 |
| 1 | 类太空侵略者的基本游戏元素已经完成 |
| 2 | 设计了UI来实现存盘读盘功能，能读取玩家的位置信息和玩家的生命值以及分数等信息 |
| 3 | 成绩排行榜功能以及完成，玩家生命值归为0之后会存储玩家的ID和分数，若在开始界面点击排行榜按钮，则会输出排行榜信息 |
| 4 | 完成了游戏背景音乐以及玩家射击，敌人被击败的音效 |
| 5 | 采用数学的方法模拟了敌人被击落后四散的粒子特效 |

1. 开发环境说明

开发环境：windows 11 64位,easyx

开发工具：visualstudio

简要介绍自己所用的开发环境，包括操作系统，开发工具等。

## 程序分析与设计

1. 系统分析

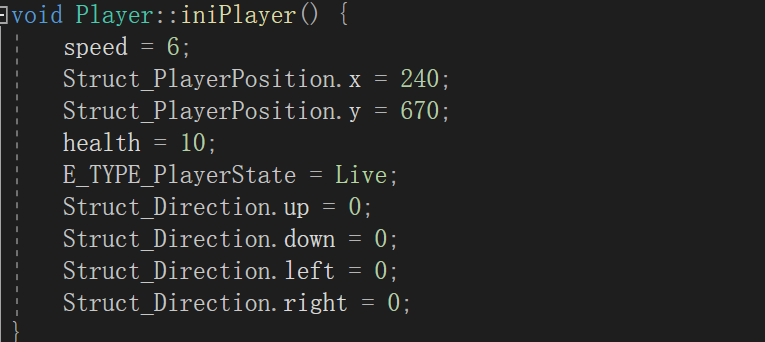
*以下任意要求在文档中缺少，不得分*

要求1：创建了哪些函数，对函数分别进行详细说明。 说明要包含：

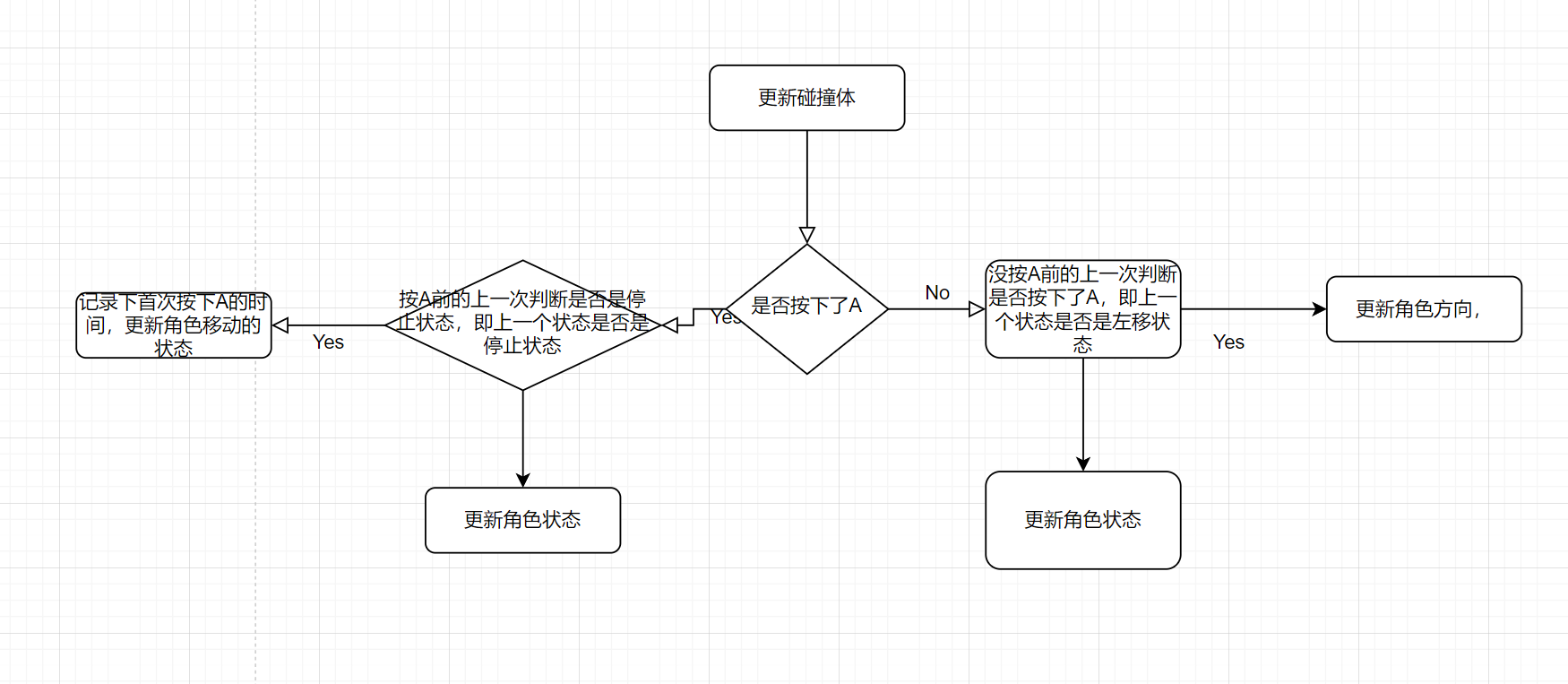
* 1. 函数原型；
  2. 函数的功能，如果有较为复杂的算法也需要说明；
  3. 各参数的意义；
  4. 函数的返回值类型

示例：

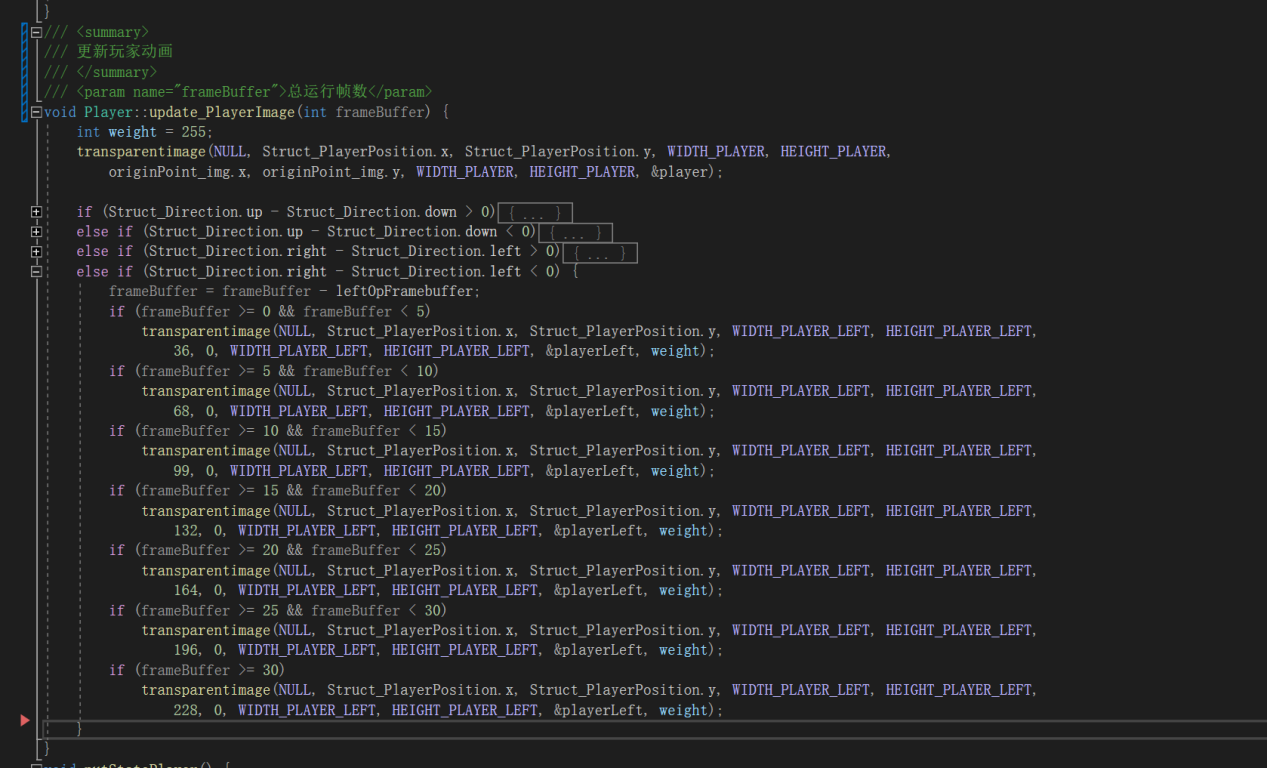
|  |  |
| --- | --- |
| 编号：001 | int max(int x, int y) |
| 函数功能 | 求两个数中大的值 |
| 参数说明 | x: int型，一个整数  y: int型，一个整数 |
| 函数返回值 | int型，两者中大的值 |
| 玩家类 |  |
| 编号：000 | void iniPlayer(); |
| 函数功能 | 初始化玩家类中的数据 |
| 参数说明 | void |
| 函数返回值 | void |

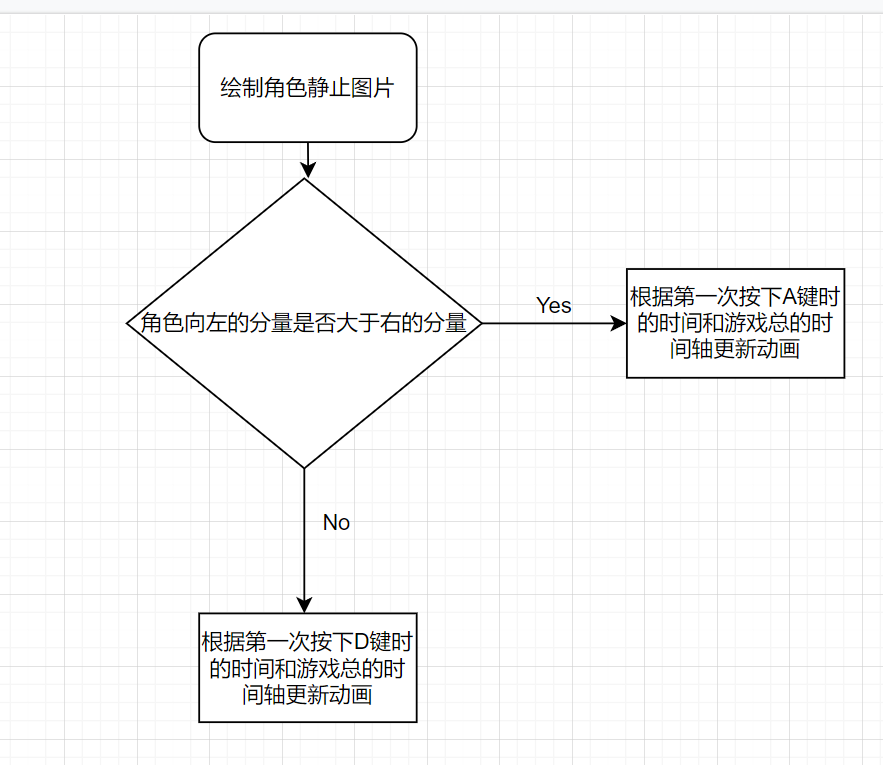


|  |  |
| --- | --- |
| 编号：001 | void Player::update\_PlayerPosition(int \_cmd, int framebuffer) |
| 函数功能 | 初始化玩家类中的数据 |
| 参数说明 | \_cmd:根据\_cmd位数上的0或者1来判断按下了哪些按键  framebuffer:游戏的总运行帧数,每运行10ms帧+1 |
| 函数返回值 | void |

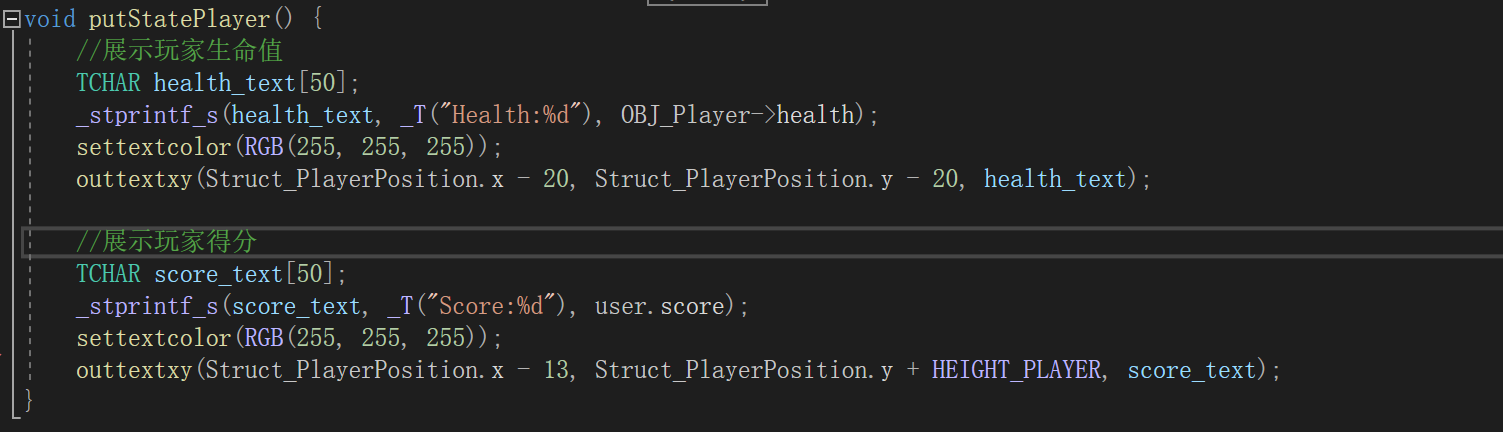


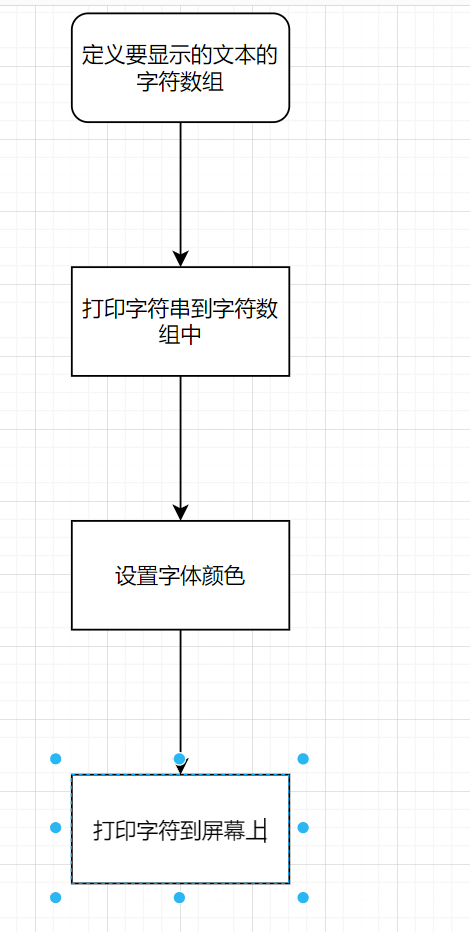
|  |  |
| --- | --- |
| 编号：003 | void Player::update\_PlayerImage(int frameBuffer) |
| 函数功能 | 更新玩家的动画 |
| 参数说明 | frameBuffer:游戏运行时的时间轴，每隔10ms+1 |
| 函数返回值 | void |





|  |  |
| --- | --- |
| 编号：004 | void putStatePlayer() |
| 函数功能 | 在游戏过程中展示玩家状态如何生命值，分数等 |
| 参数说明 | void |
| 函数返回值 | void |

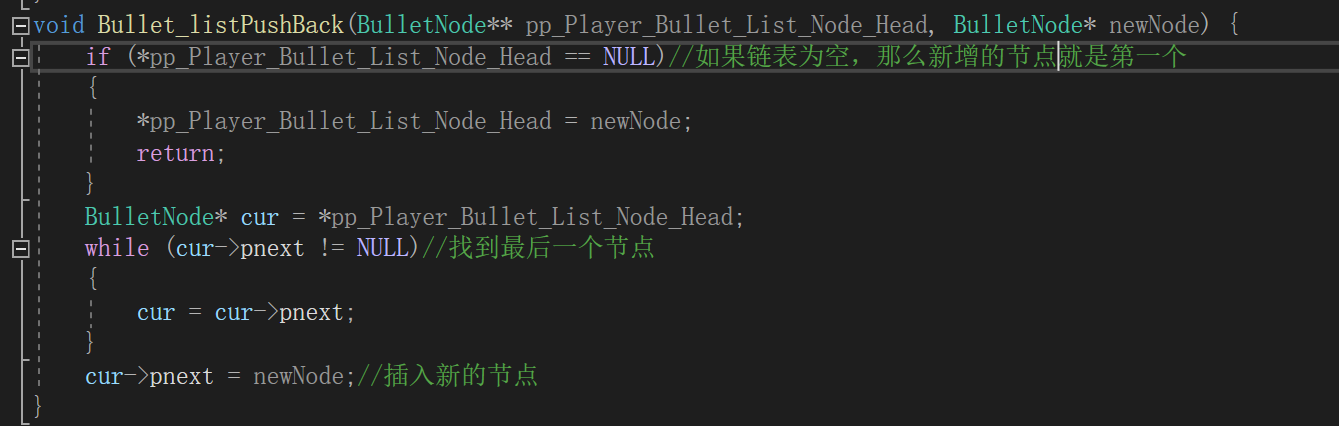


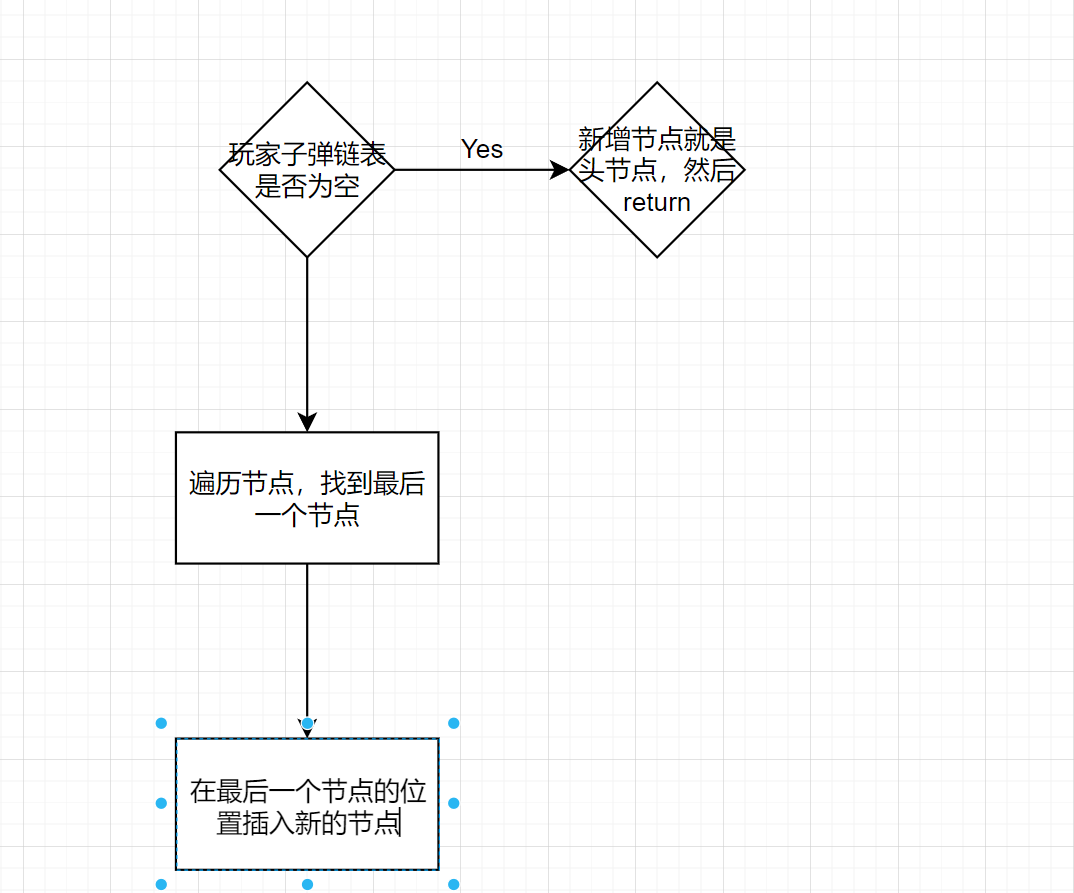


|  |  |
| --- | --- |
| 编号：004 | void Player::update\_Player(int n\_command, int framebuffer) |
| 函数功能 | 调用位置更新函数和图像更新函数 |
| 参数说明 | n\_command:获取按键操作  Framebuffer:时间轴 |
| 函数返回值 | void |

玩家子弹类:

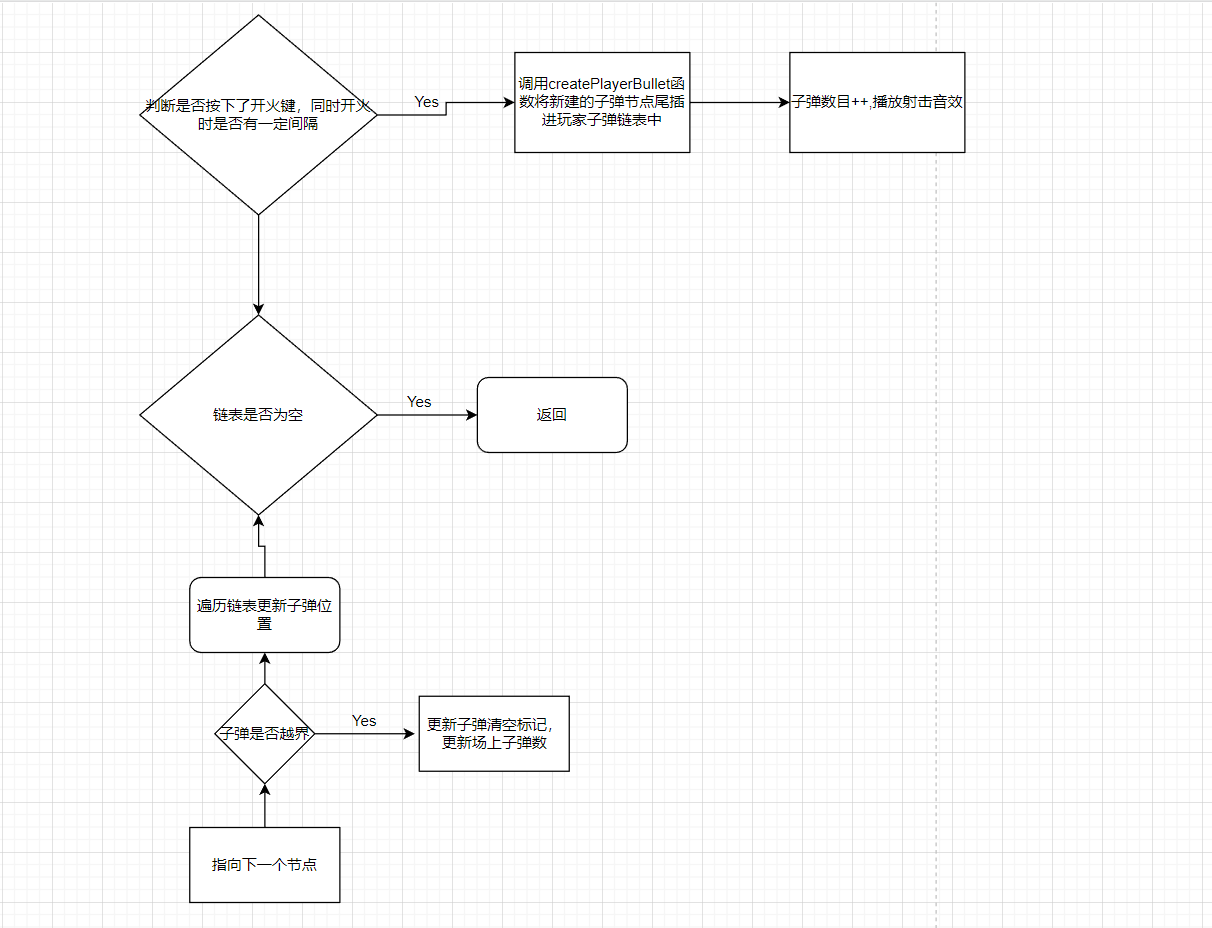
|  |  |
| --- | --- |
| 编号：000 | BulletNode\* creatPlayerBullet(int vx, int vy, int hitpoint)  BulletNode\* creatPlayerBullet(int x, int y, int vx, int vy, int hitpoint) |
| 函数功能 | 重载：一个用于创建一个玩家子弹类的节点，初始位置在角色正前方，一个用于读档时在任意位置创建子弹。 |
| 参数说明 | vx:子弹的x方向上的增量  Vy:子弹的y方向上的增量  x:子弹的初始位置  y:子弹的初始位置 |
| 函数返回值 | 指向BulletNode的一个指针 |
| 编号：001 | Void Bullet\_listPushBack(BulletNode\*\* pp\_Player\_Bullet\_List\_Node\_Head, BulletNode\* newNode) |
| 函数功能 | 重载：一个用于创建一个玩家子弹类的节点，初始位置在角色正前方，一个用于读档时在任意位置创建子弹。 |
| 参数说明 | vx:子弹的x方向上的增量  Vy:子弹的y方向上的增量  x:子弹的初始位置  y:子弹的初始位置 |
| 函数返回值 | 指向BulletNode的一个指针 |



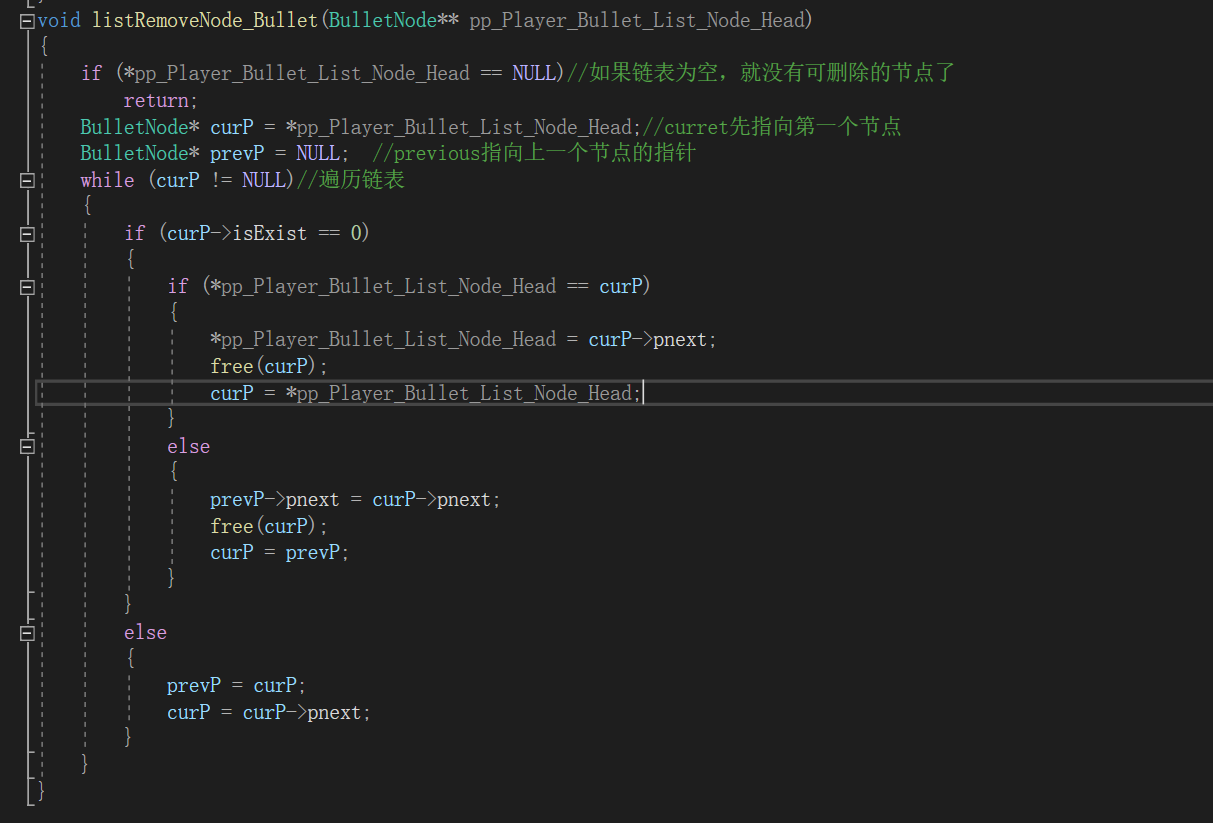


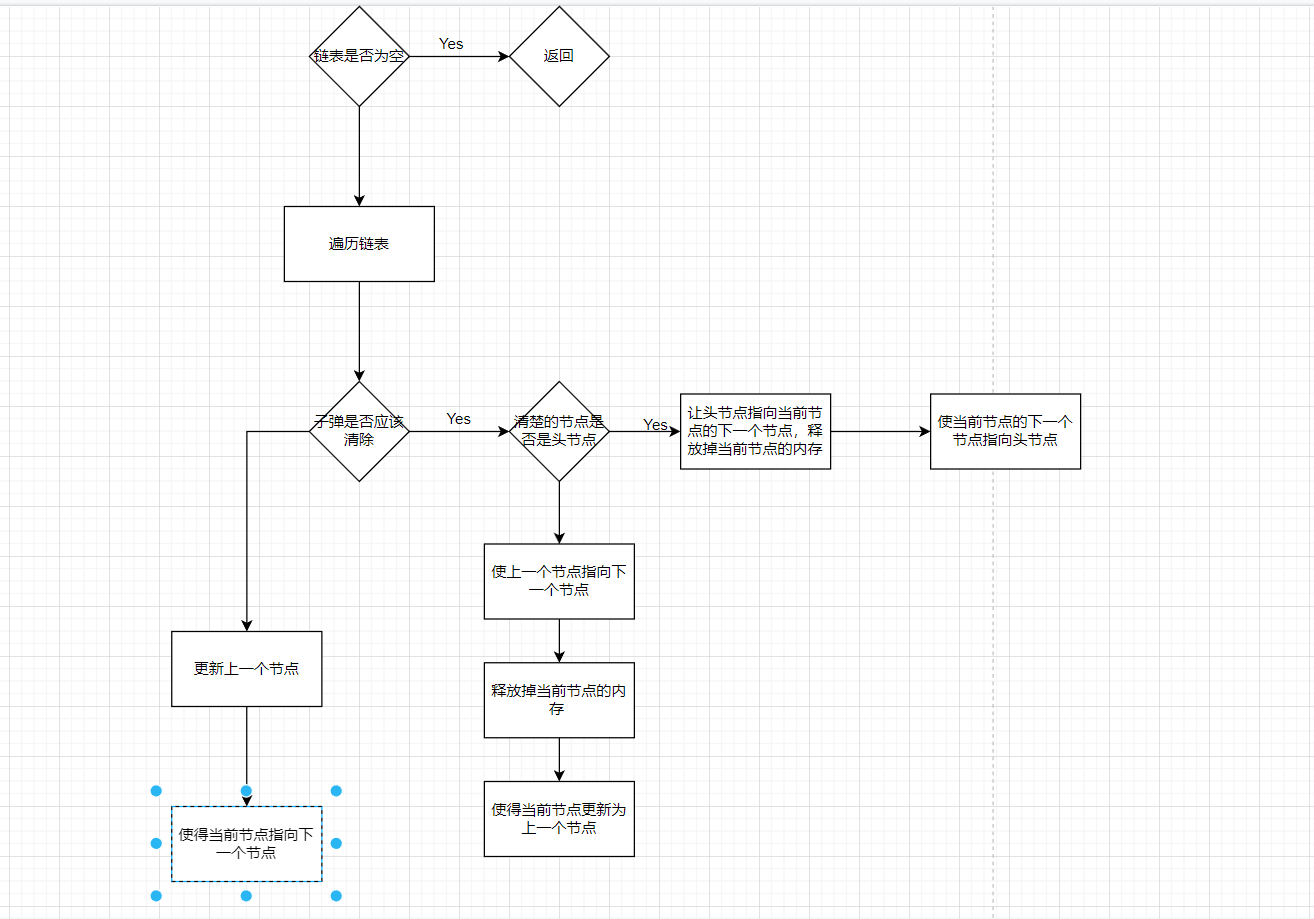
|  |  |
| --- | --- |
| 编号：002 | void update\_BulletPosition(BulletNode\*\* pp\_Player\_Bullet\_List\_Node\_Head, int command, int frameBuffer, int vx, int vy, int hitpoint) |
| 函数功能 | 遍历每个子弹节点，更新玩家子弹位置 |
| 参数说明 | pp\_Player\_Bullet\_List\_Node\_HeadVy:指向指向玩家子弹节点的指针的指针  Command:用于获取玩家按键操作  frameBuffer:游戏总的时间轴  Vx,vy:x和y方向上的增量  Hitpoint:玩家子弹的伤害值 |
| 函数返回值 | void |



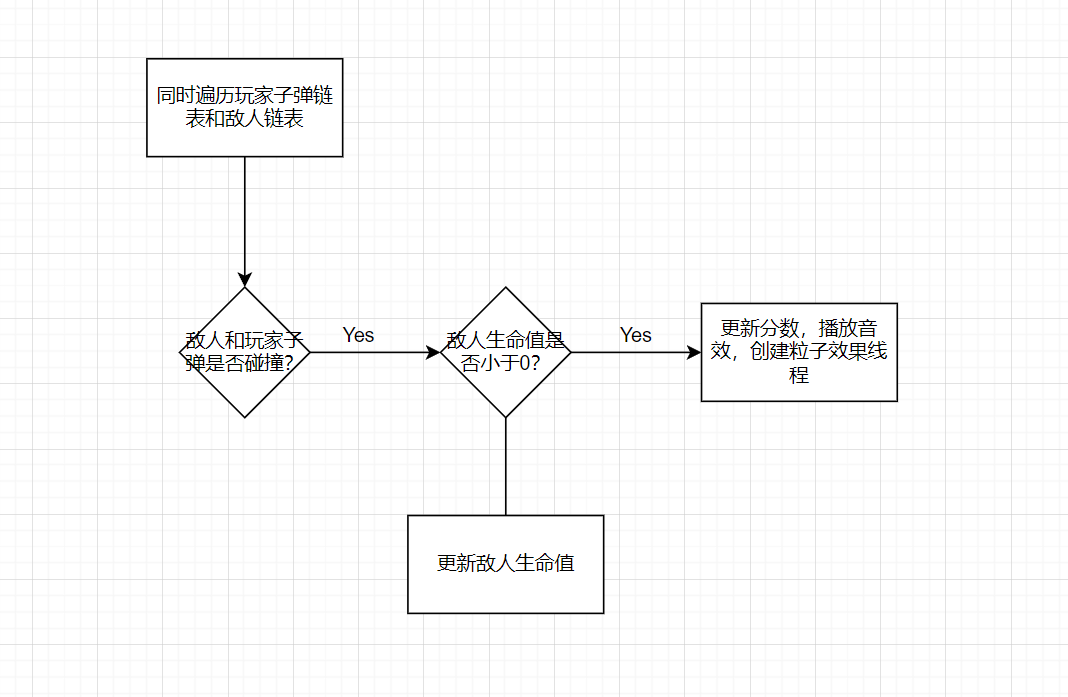
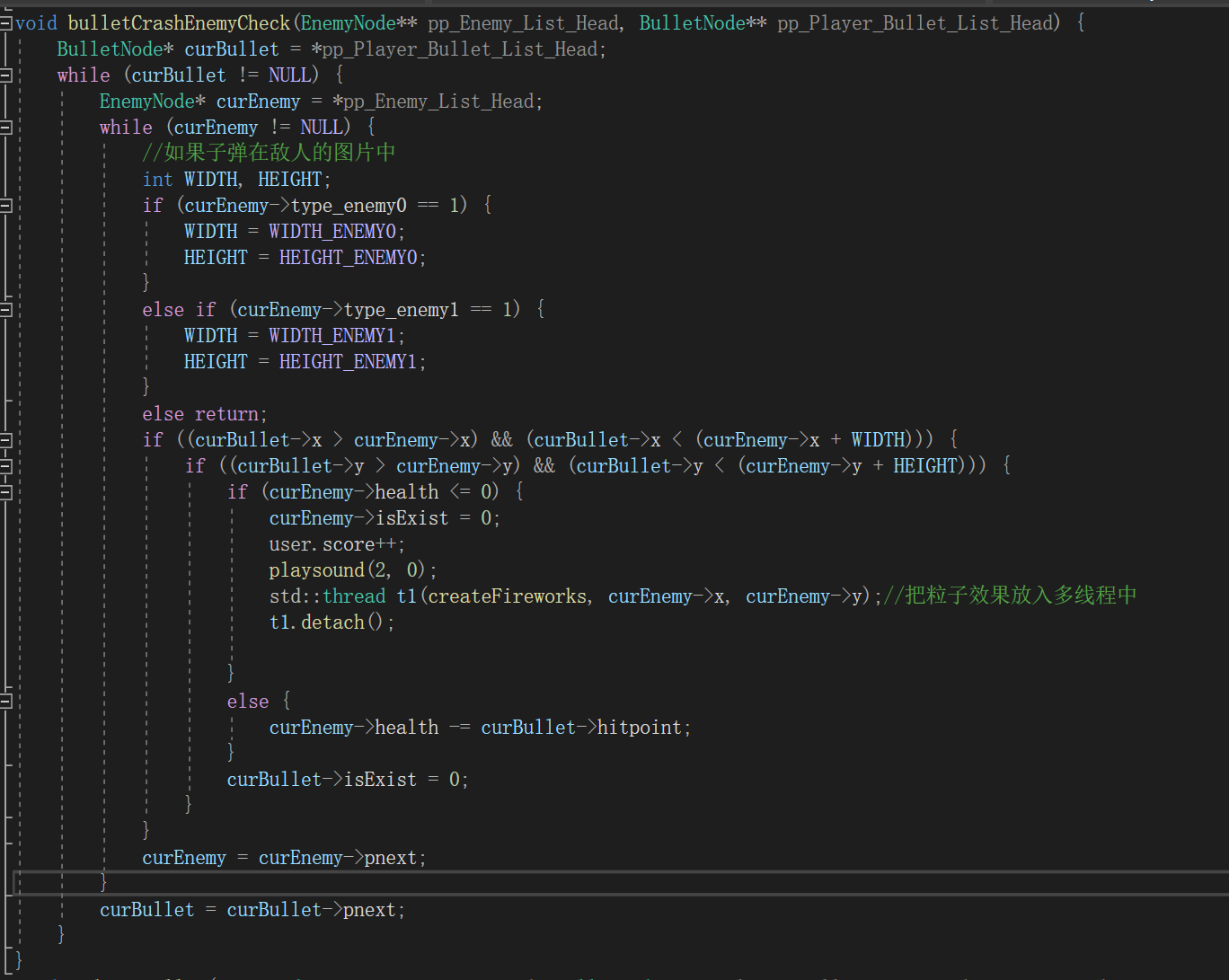


|  |  |
| --- | --- |
| 编号：003 | void listRemoveNode\_Bullet(BulletNode\*\* pp\_Player\_Bullet\_List\_Node\_Head) |
| 函数功能 | 遍历每个子弹节点，清楚应该清空的子弹节点 |
| 参数说明 | pp\_Player\_Bullet\_List\_Node\_HeadVy:指向指向玩家子弹节点的指针的指针 |
| 函数返回值 | void |





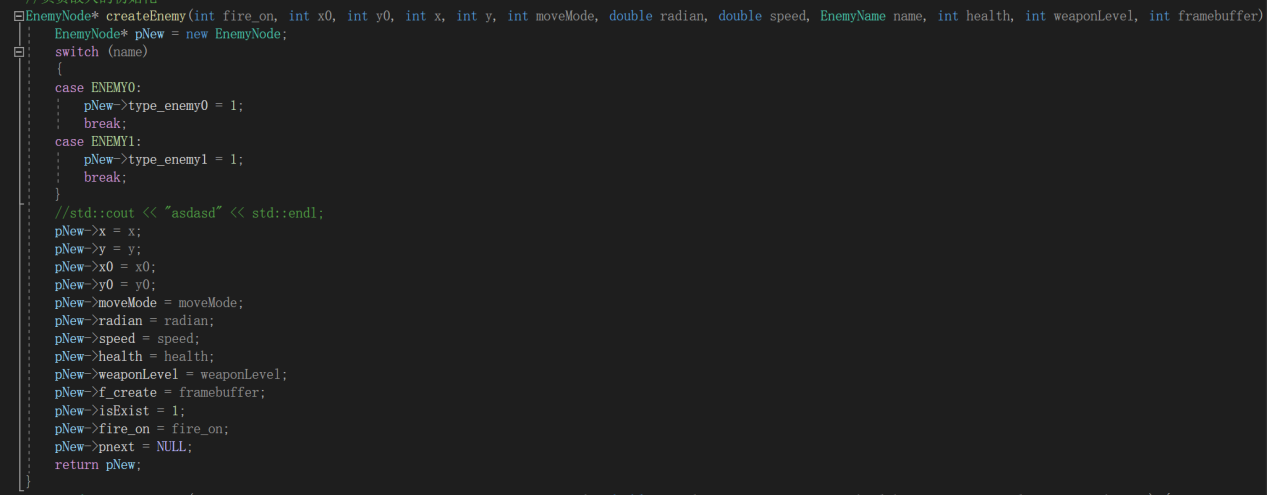
|  |  |
| --- | --- |
| 编号：004 | void update\_BulletImage(BulletNode\*\* pp\_Player\_Bullet\_List) |
| 函数功能 | 根据位置更新每个子弹的Sprite |
| 参数说明 | pp\_Player\_Bullet\_List\_Node\_HeadVy:指向指向玩家子弹节点的指针的指针 |
| 函数返回值 | void |
| 编号：005 | void bulletCrashEnemyCheck(EnemyNode\*\* pp\_Enemy\_List\_Head, BulletNode\*\* pp\_Player\_Bullet\_List\_Head) |
| 函数功能 | 根据位置更新每个子弹的Sprite |
| 参数说明 | pp\_Enemy\_List\_Head::指向指向敌人节点的指针的指针  pp\_Player\_Bullet\_List\_Node\_HeadVy:指向指向玩家子弹节点的指针的指针 |
| 函数返回值 | void |

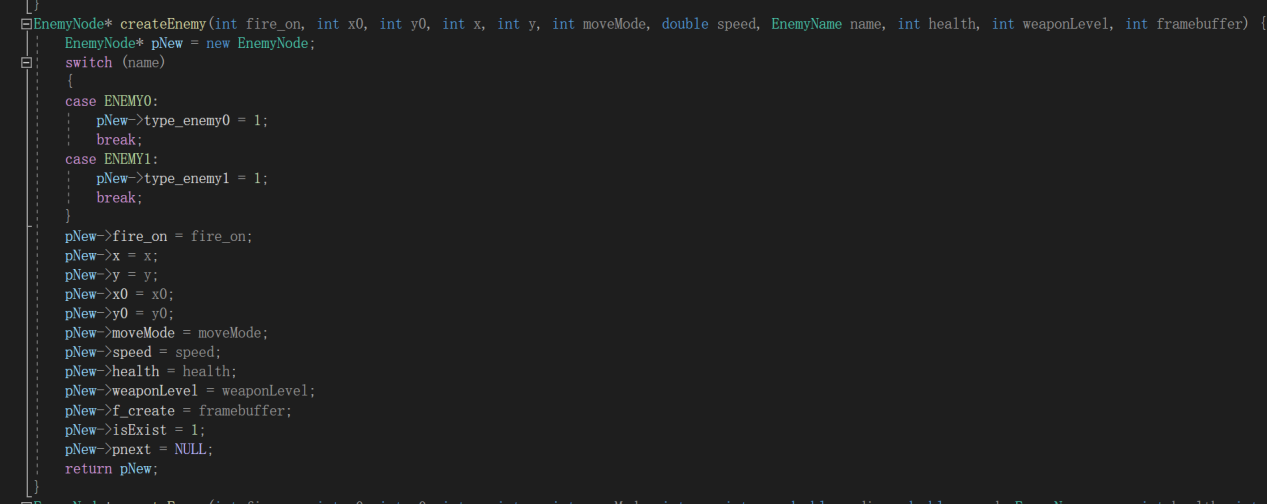


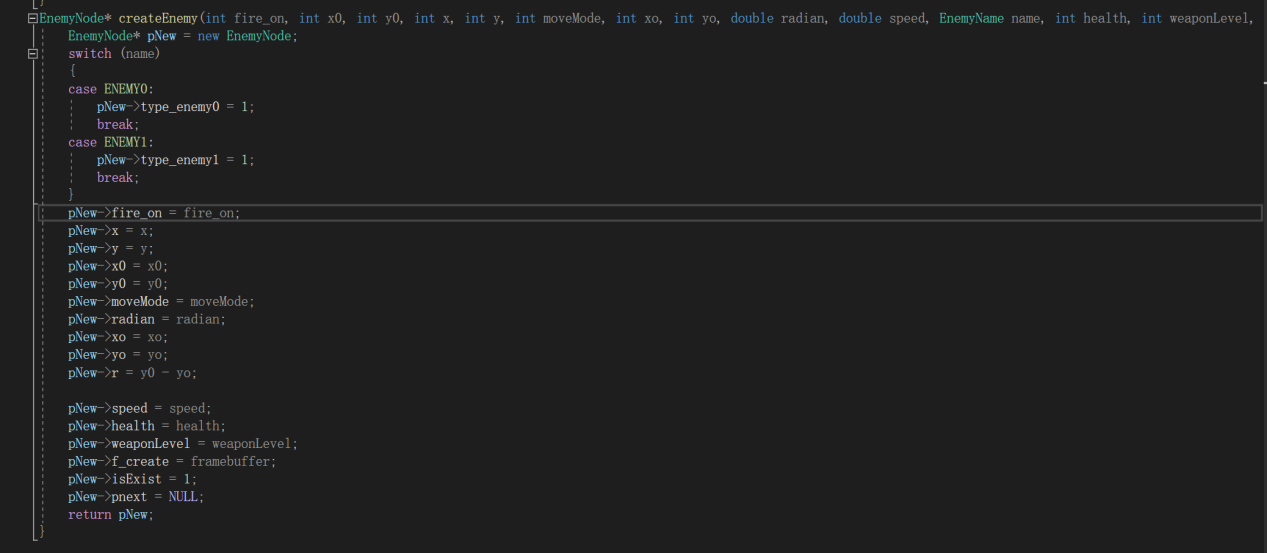
|  |  |
| --- | --- |
| 编号：006 | void update\_Bullet(EnemyNode\*\* pp\_Enemy\_List\_Head, BulletNode\*\* pp\_Player\_Bullet\_List\_Head, int command, int frameBuffer, int vx, int vy, int hitpoint) |
| 函数功能 | 调用update\_BulletPosition  bulletCrashEnemyCheck  listRemoveNode\_Bullet函数 |
| 参数说明 | 略 |
| 函数返回值 | void |

敌人类：

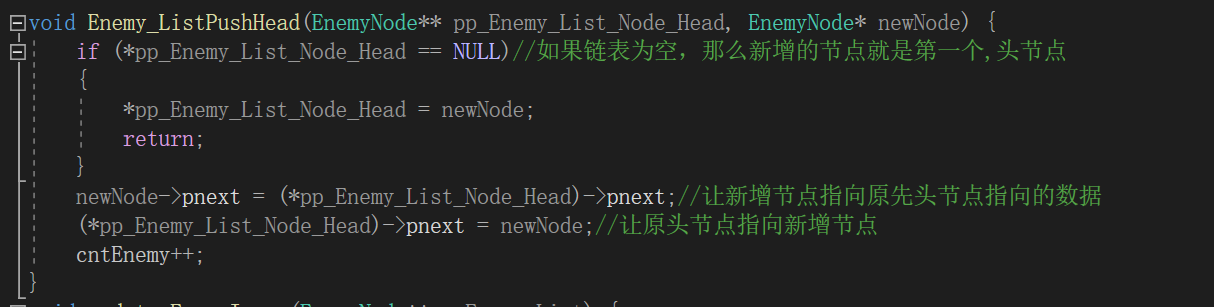
|  |  |
| --- | --- |
| 编号：000 | EnemyNode\* createEnemy(int fire\_on, int x0, int y0, int x, int y, int moveMode, double radian, double speed, EnemyName name, int health, int weaponLevel, int framebuffer)  EnemyNode\* createEnemy(int fire\_on, int x0, int y0, int x, int y, int moveMode, double speed, EnemyName name, int health, int weaponLevel, int framebuffer)  EnemyNode\* createEnemy(int fire\_on, int x0, int y0, int x, int y, int moveMode, int xo, int yo, double radian, double speed, EnemyName name, int health, int weaponLevel, int framebuffer) |
| 函数功能 | 创建敌人节点 |
| 参数说明 | fire\_on:敌人每隔多少帧开一次火  x0,y0:敌人的初始位置  x,y:敌人的实时位置  movemode:敌人的移动模式  radian:偏转角,沿着y方向为0,逆时针旋转偏转角增大  speed:移动速度  name:枚举类型,敌人的种类  health:敌人初始生命值  framebuffer:游戏的时间轴，总共进行了多少帧 |
| 函数返回值 | EnemyNode\* |

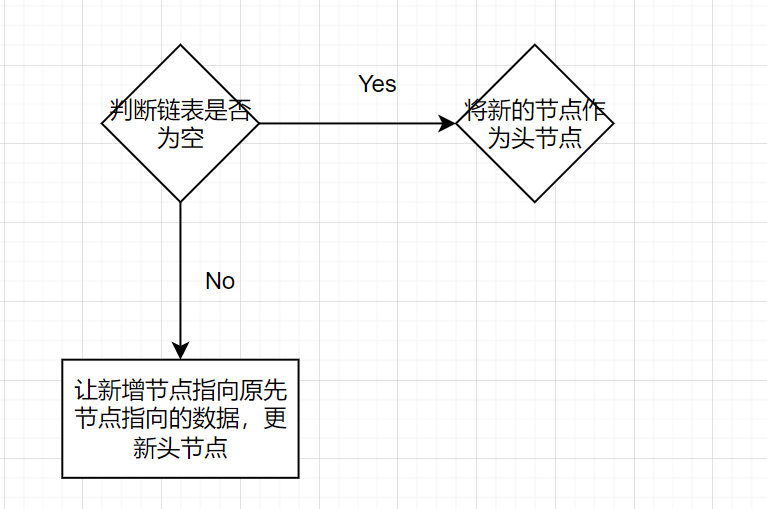




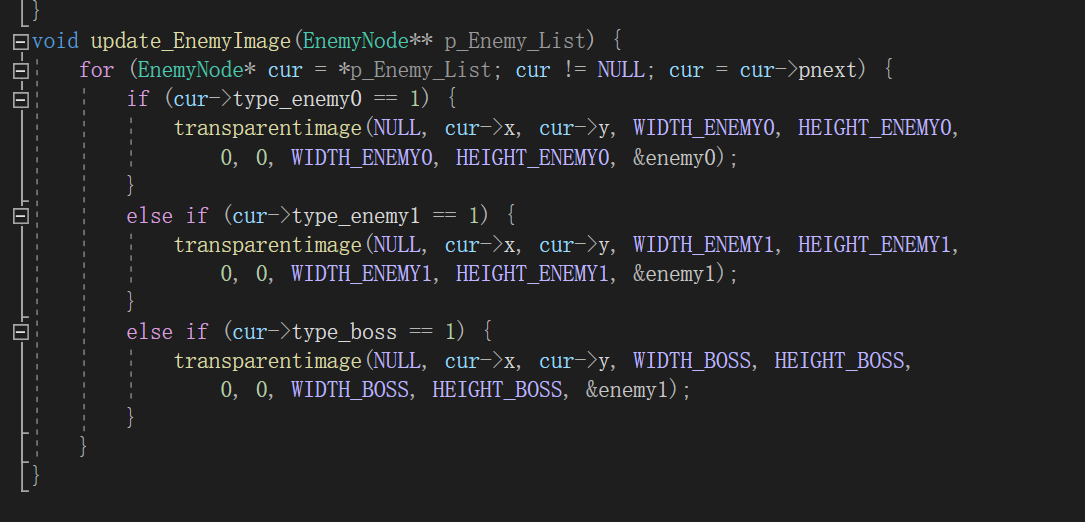


|  |  |
| --- | --- |
| 编号：001 | void Enemy\_ListPushHead(EnemyNode\*\* pp\_Enemy\_List\_Node\_Head, EnemyNode\* newNode) |
| 函数功能 | 将新增节点头插到敌人链表中 |
| 参数说明 | 略 |
| 函数返回值 | void |

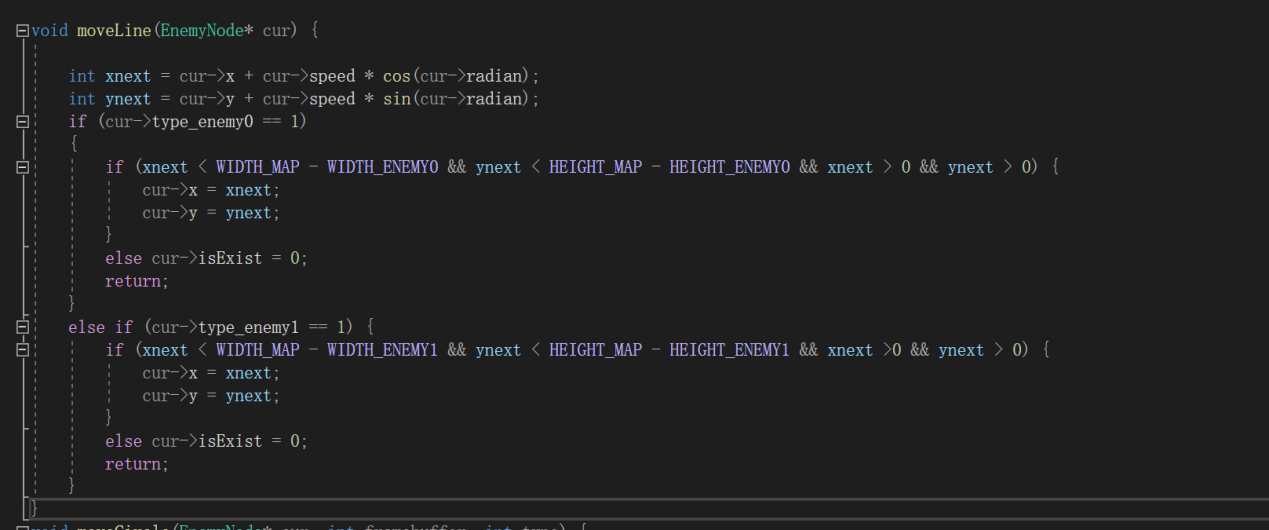


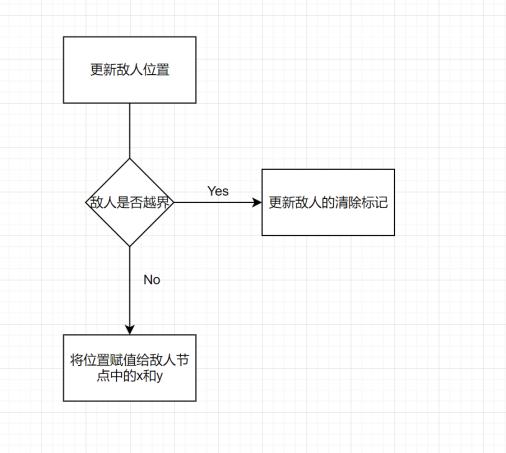


|  |  |
| --- | --- |
| 编号：002 | void update\_EnemyImage(EnemyNode\*\* p\_Enemy\_List) |
| 函数功能 | 更新敌人动画 |
| 参数说明 | 指向敌人链表的二级指针 |
| 函数返回值 | void |

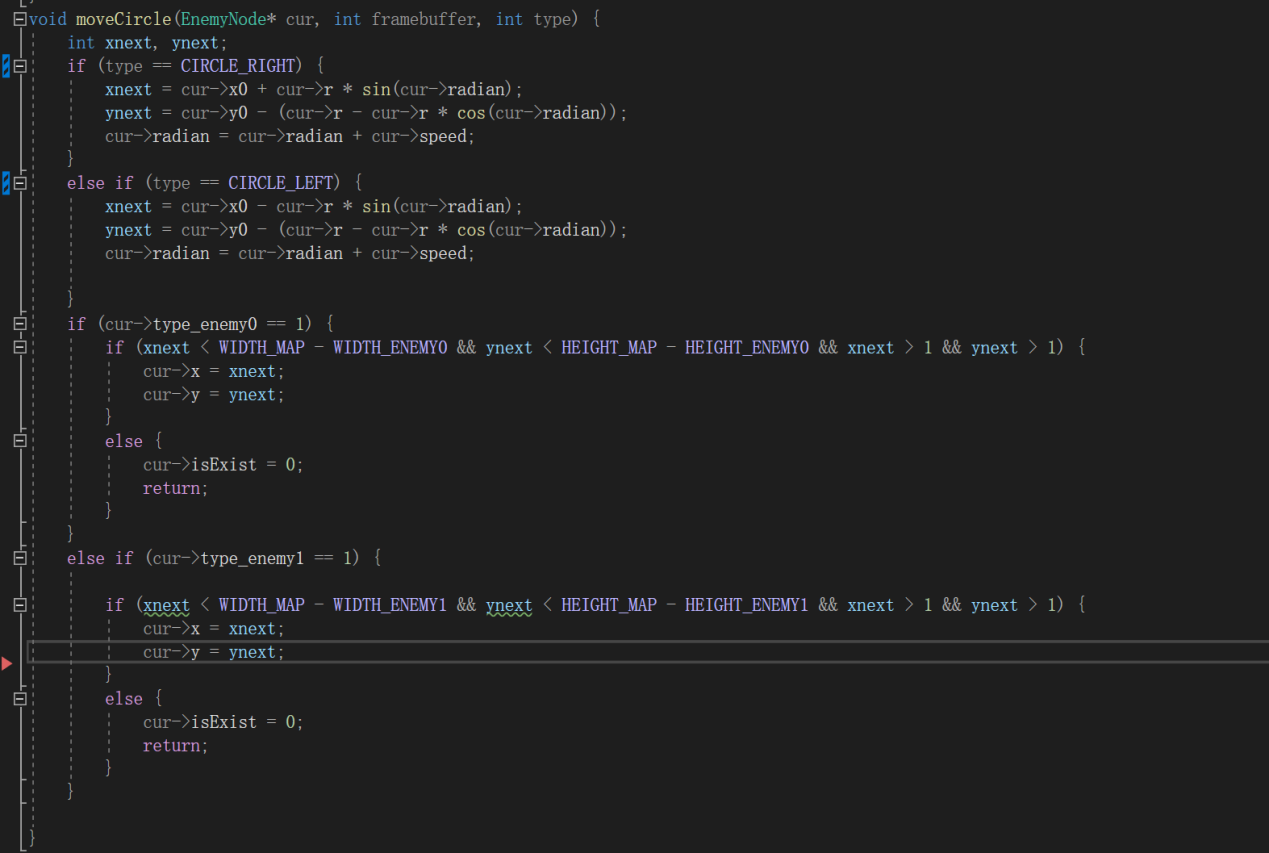


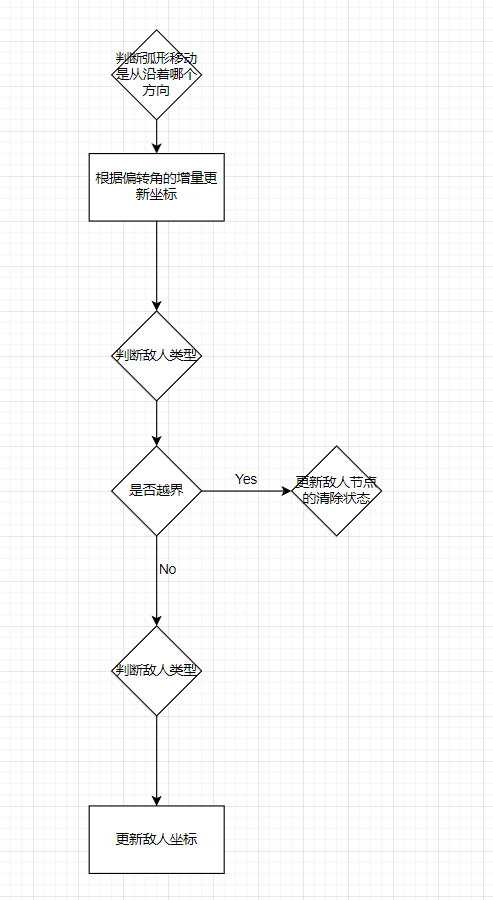
|  |  |
| --- | --- |
| 编号：002 | void moveLine(EnemyNode\* cur) |
| 函数功能 | 使敌人线性的走 |
| 参数说明 | cur:指向敌人链表中的某一节点 |
| 函数返回值 | void |



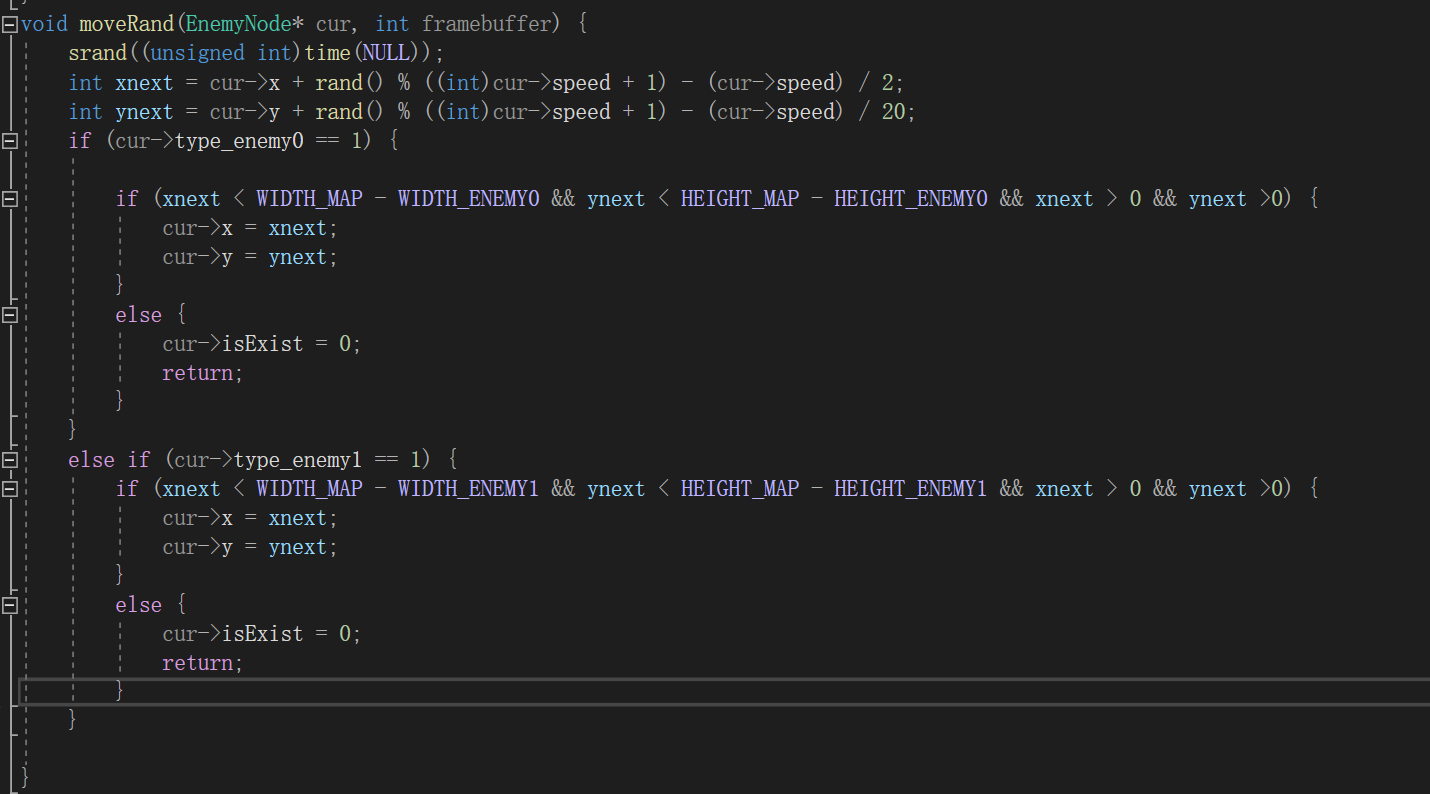


|  |  |
| --- | --- |
| 编号：003 | void moveCircle(EnemyNode\* cur, int framebuffer, int type) |
| 函数功能 | 使敌人沿着圆弧走 |
| 参数说明 | cur:指向敌人链表中的某一节点  framebuffer:游戏总经过了多少帧  type:选择圆弧向左还是向右走 |
| 函数返回值 | void |

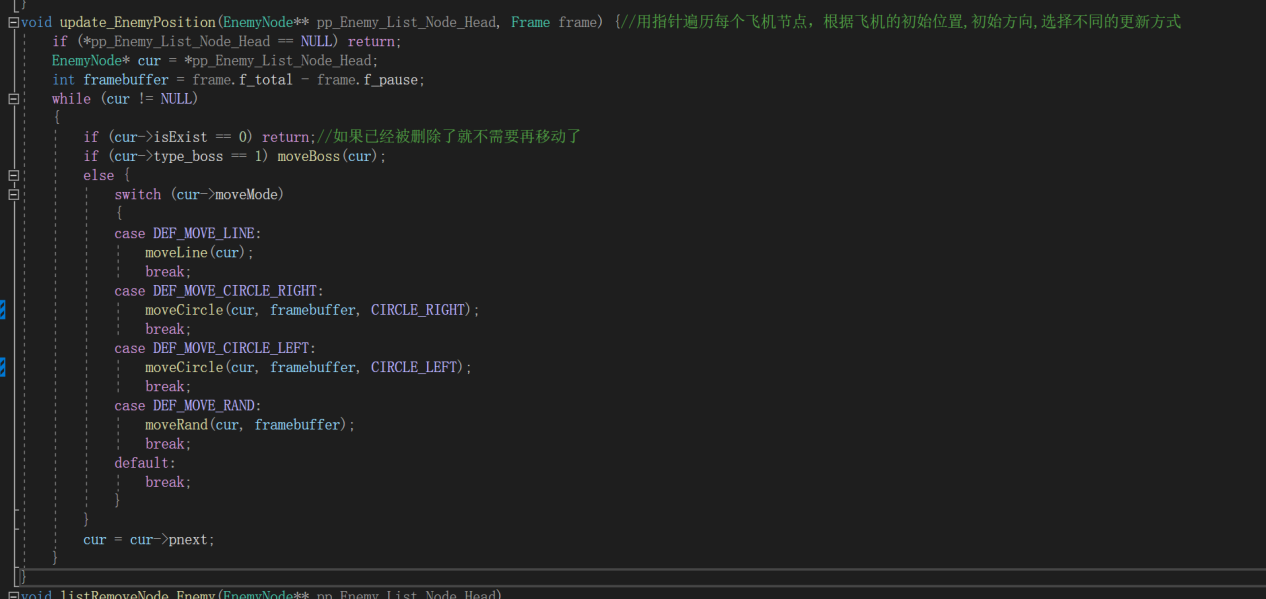




|  |  |
| --- | --- |
| 编号：004 | void moveRand(EnemyNode\* cur, int framebuffer) |
| 函数功能 | 使敌人随机移动 |
| 参数说明 | cur:指向敌人链表中的某一节点  framebuffer:游戏总经过了多少帧 |
| 函数返回值 | void |

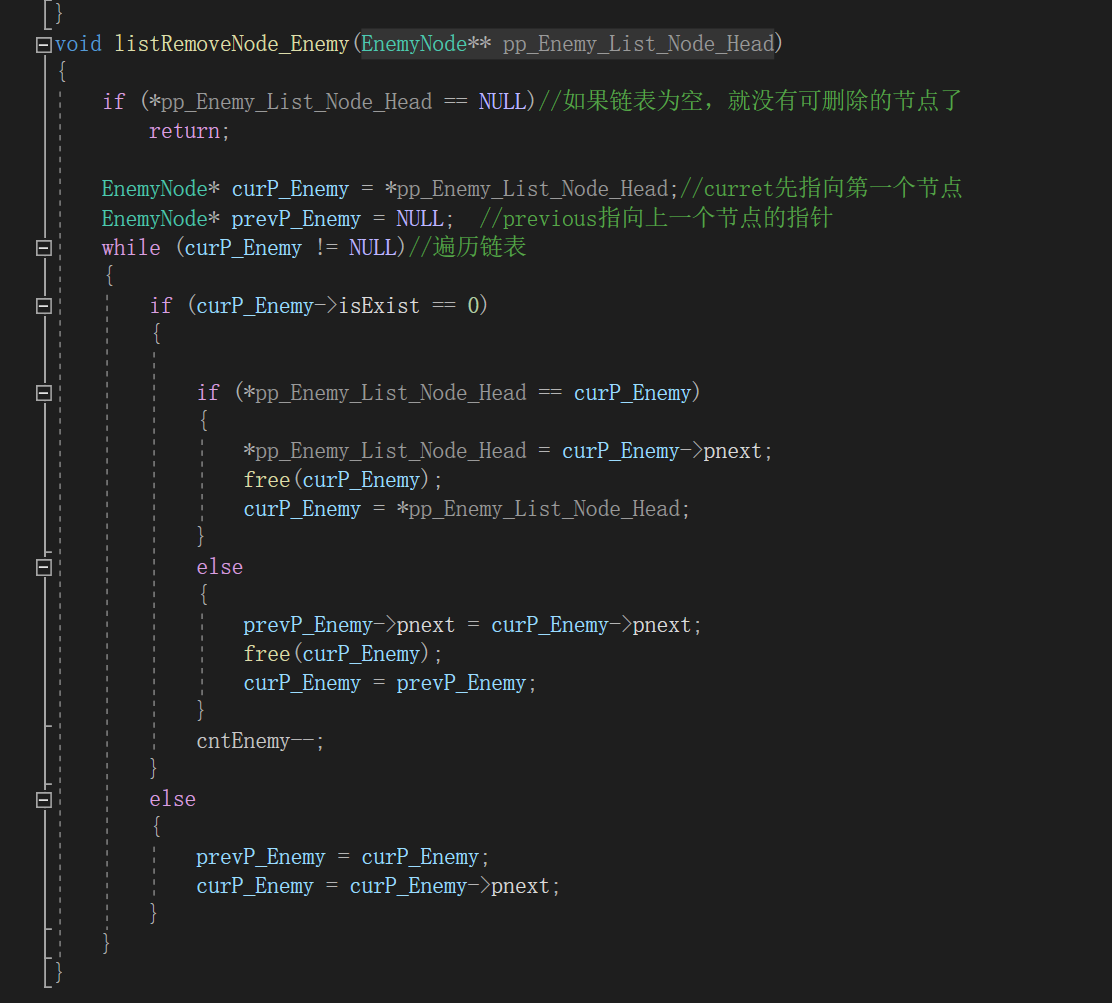


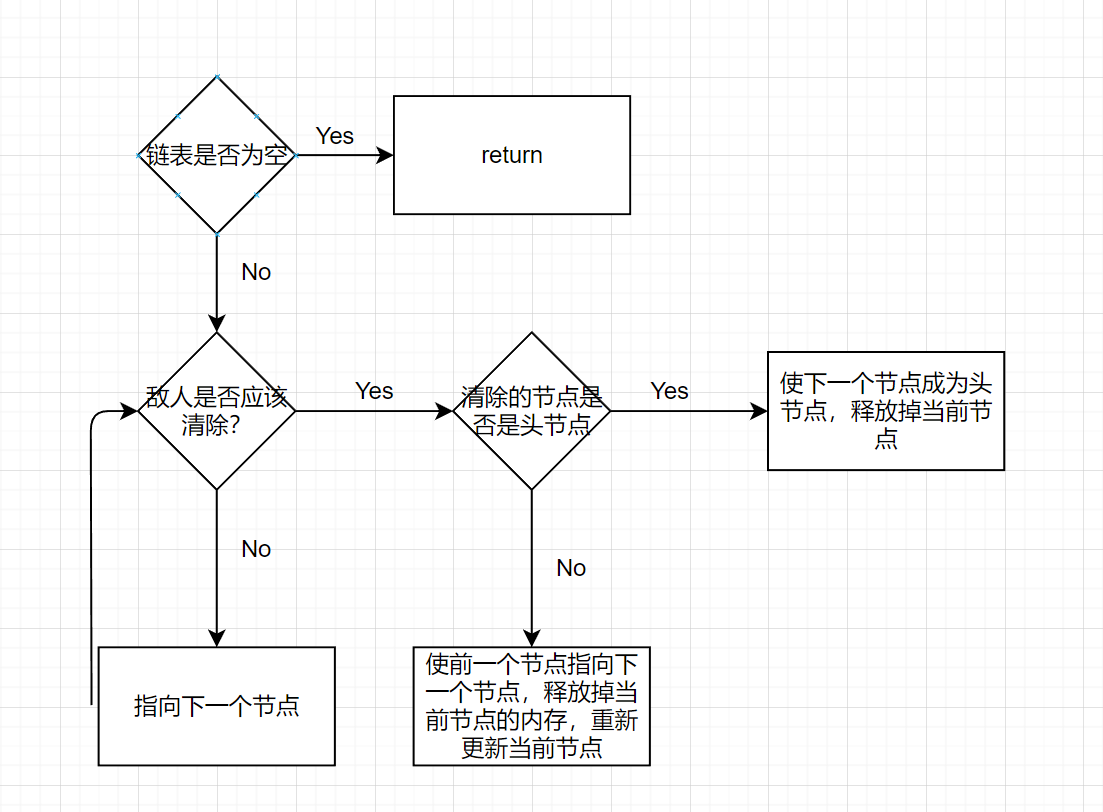
|  |  |
| --- | --- |
| 编号：005 | void update\_EnemyPosition(EnemyNode\*\* pp\_Enemy\_List\_Node\_Head, Frame frame) |
| 函数功能 | 更新敌人Sprite |
| 参数说明 | pp\_Enemy\_List\_Node\_Head:指向敌人链表的的二级指针  frame:控制游戏的时间轴的结构体 |
| 函数返回值 | void |





|  |  |
| --- | --- |
| 编号：006 | void listRemoveNode\_Enemy(EnemyNode\*\* pp\_Enemy\_List\_Node\_Head) |
| 函数功能 | 从链表中删除越界或者生命值小于等于0的敌人 |
| 参数说明 | pp\_Enemy\_List\_Node\_Head:指向敌人链表的的二级指针 |
| 函数返回值 | void |



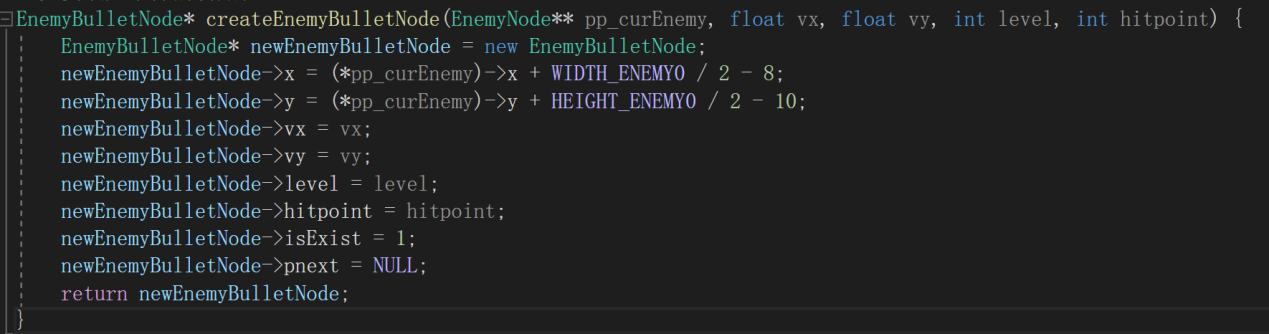


|  |  |
| --- | --- |
| 编号：007 | void update\_Enemy(EnemyNode\*\* pp\_Enemy\_List\_Node\_Head, Frame frame) |
| 函数功能 | 调用更新敌人位置和清除敌人函数 |
| 参数说明 | pp\_Enemy\_List\_Node\_Head:指向敌人链表的的二级指针  frame:控制游戏时间轴的类 |
| 函数返回值 | void |

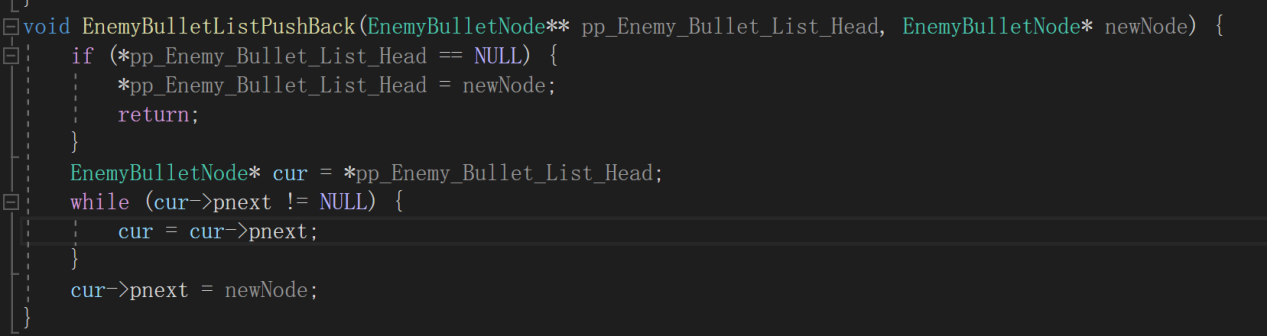
#### 

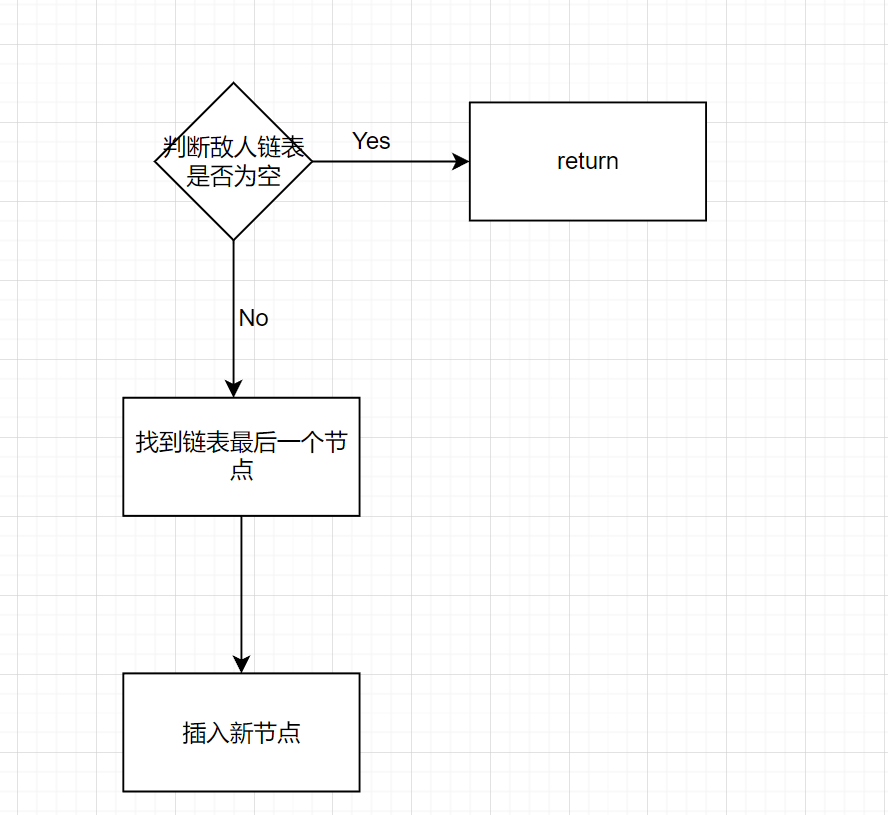
敌人子弹类：

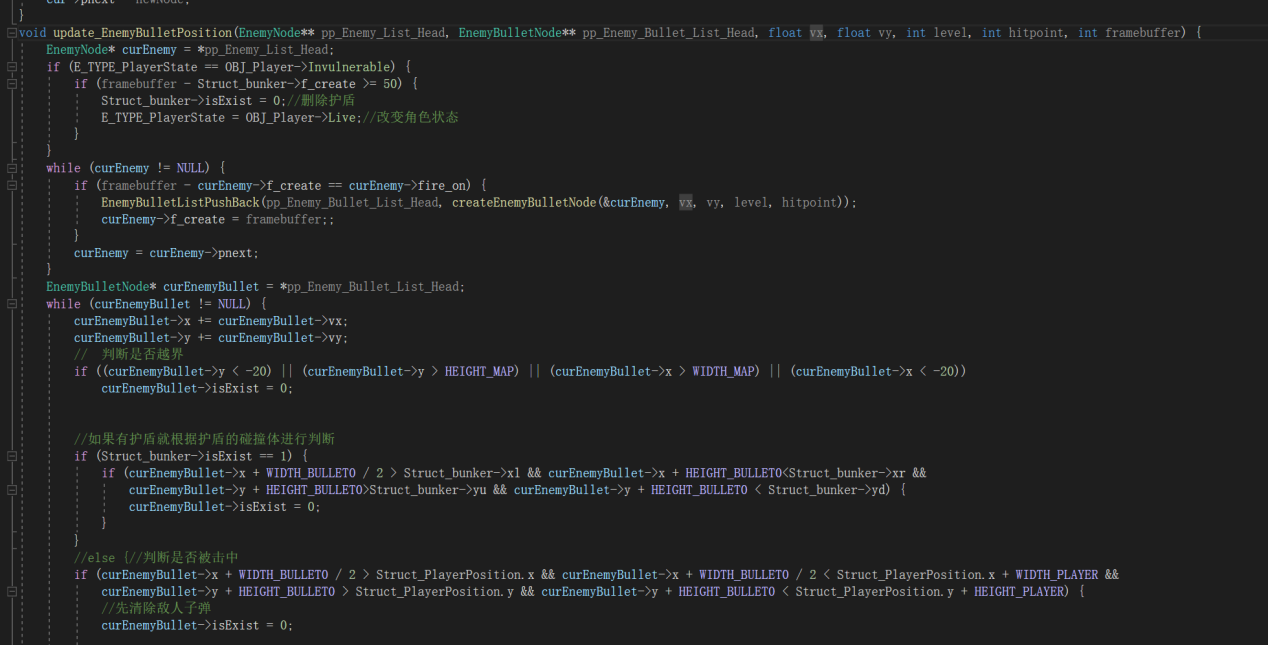
|  |  |
| --- | --- |
| 编号：000 | EnemyBulletNode\* createEnemyBulletNode(EnemyNode\*\* pp\_curEnemy, float vx, float vy, int level, int hitpoint) |
| 函数功能 | 根据敌人位置创建敌人子弹节点 |
| 参数说明 | pp\_curEnemy:指向敌人链表的其中一个节点的的二级指针  vx,vy:子弹在x和y方向上的增量  hitpoint:敌人子弹的伤害 |
| 函数返回值 | EnemyBulletNode\* |

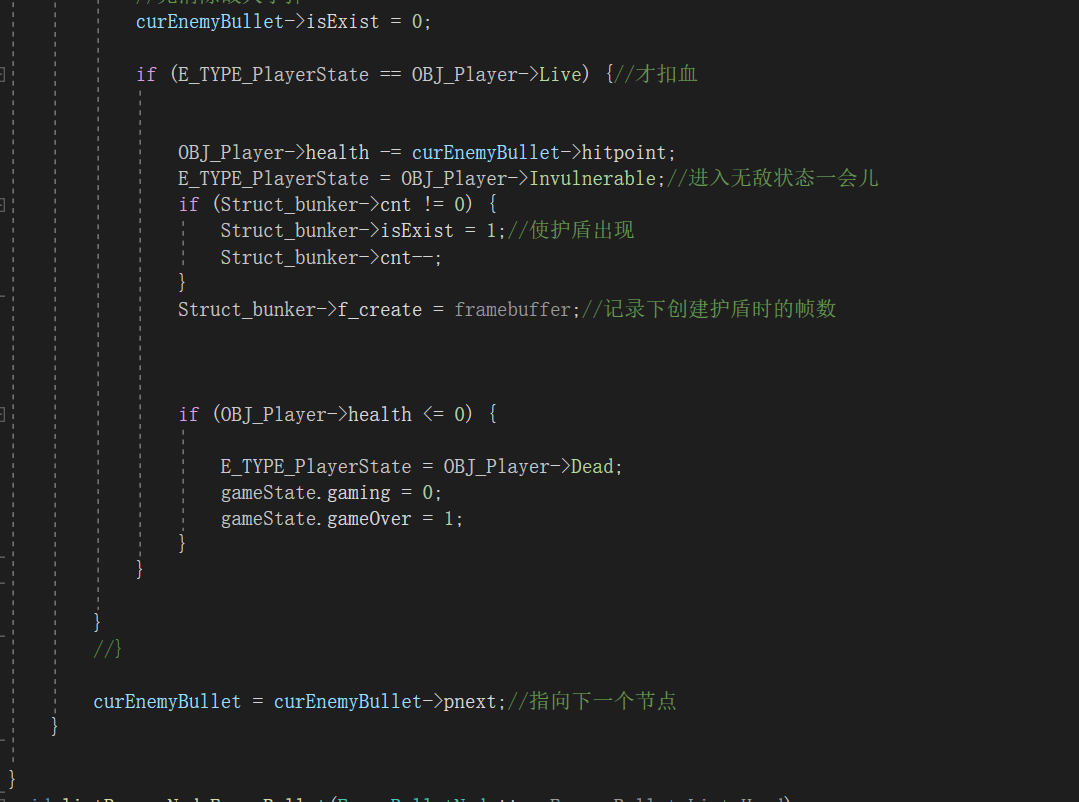


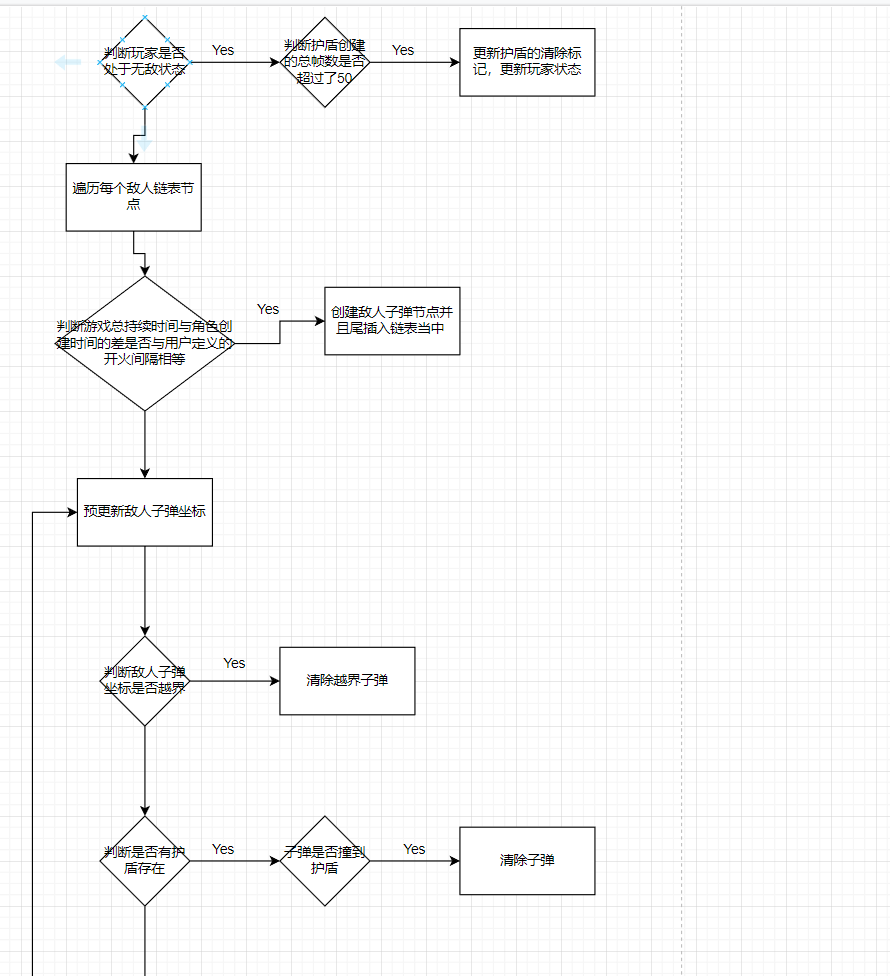
|  |  |
| --- | --- |
| 编号：001 | void EnemyBulletListPushBack(EnemyBulletNode\*\* pp\_Enemy\_Bullet\_List\_Head, EnemyBulletNode\* newNode) |
| 函数功能 | 将新增的子弹节点尾插到链表中 |
| 参数说明 | pp\_Enemy\_List\_Node\_Head:指向敌人链表的的二级指针  newNode:指向新建节点 |
| 函数返回值 | void |

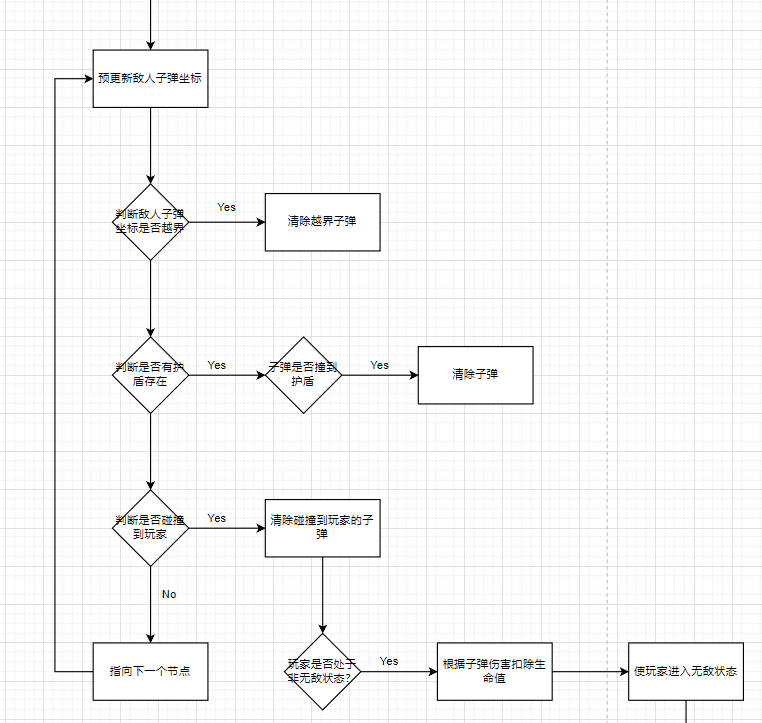


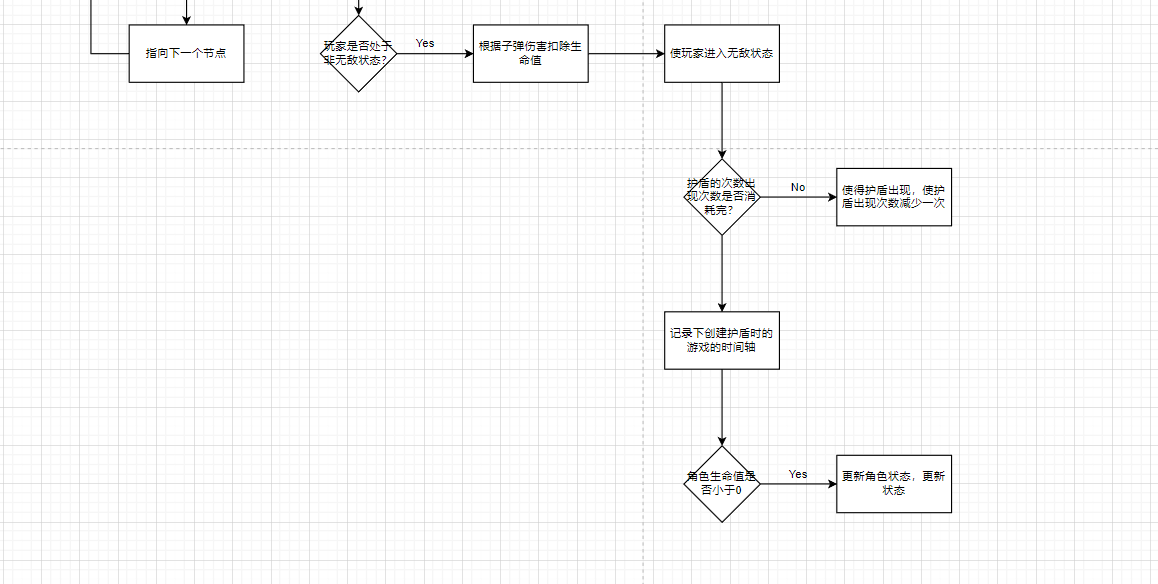










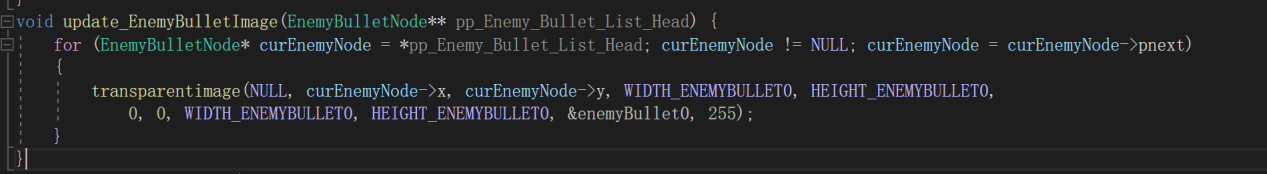


|  |  |
| --- | --- |
| 编号：003 | void listRemoveNodeEnemyBullet(EnemyBulletNode\*\* p\_Enemy\_Bullet\_List\_Head) |
| 函数功能 | 清除越界的敌人子弹 |
| 参数说明 | pp\_Enemy\_Bullet\_List\_Head:指向敌人子弹链表的的二级指针 |
| 函数返回值 | void |

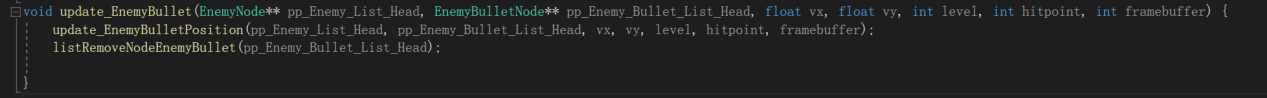


流程图同玩家子弹函数

|  |  |
| --- | --- |
| 编号：003 | void update\_EnemyBulletImage(EnemyBulletNode\*\* pp\_Enemy\_Bullet\_List\_Head) |
| 函数功能 | 更新敌人子弹的Sprite |
| 参数说明 | pp\_Enemy\_Bullet\_List\_Head:指向敌人子弹链表的二级指针 |
| 函数返回值 | void |



|  |  |
| --- | --- |
| 编号：004 | void update\_EnemyBullet(EnemyNode\*\* pp\_Enemy\_List\_Head, EnemyBulletNode\*\* pp\_Enemy\_Bullet\_List\_Head, float vx, float vy, int level, int hitpoint, int framebuffer) |
| 函数功能 | 调用子弹位置更新函数和子弹清除函数 |
| 参数说明 | pp\_Enemy\_Bullet\_List\_Head:指向敌人子弹链表的二级指针  pp\_Enemy\_List\_Head:指向敌人链表的二级指针  vx,vy:敌人子弹x方向和y方向上的增量  level:子弹的等级  hitpoint:子弹的伤害  framebuffer:游戏的时间轴 |
| 函数返回值 | void |



粒子类

|  |  |
| --- | --- |
| 编号：000 | ParticleSwarm::ParticleSwarm(int x, int y, float colorh = float(rand() % 256)) |
| 函数功能 | 初始化单个粒子群 |
| 参数说明 | x,y:粒子群的初始坐标  colorh:粒子群的初始颜色 |
| 函数返回值 | void |

要求2：函数之间的调用关系图

1. 函数设计

要求：

在每个3级标题下，对创建的每个函数分别画出流程图（推荐使用visio或者draw.io等专业绘图工具），贴出对应代码和注释，无流程图、无代码注释、注释与流程图对应不上不得分。

## 测试分析

1. 测试与调试
2. 测试结果分析
3. 结论

## 系统演示

1. 系统个功能运行截图与说明

## 设计体会

1. 设计过程中的问题及其解决方案
2. 设计过程中的错误及其原因分析
3. 体会和收获

参考教程：[opengl制作烟花教程-百度经验 (baidu.com)](https://jingyan.baidu.com/article/597a06430119c8312b524304.html)

[(65条消息) 基于OpenGL的烟花粒子系统\_linuxheik的博客-CSDN博客](https://blog.csdn.net/linuxheik/article/details/71109751)