泡泡堂游戏的设计与实现

李俊良 段华琼
四川大学锦城学院计算机与软件学院 四川 成都 611371

【摘 要】目前已有的泡泡堂游戏主要基于 MFC、Windows GDI 开发。本文以 Microsoft Visual Studio 2017 为开发平台，基于 C++ 的轻量级 EasyX 库，设计开发了一款可玩性强、操作方便、画面精美的泡泡堂游戏。游戏对已有的泡泡堂游戏进行了改进，采用依据像素推导人物所在单元格的方法，解决了人物目标单元格位置判定错误的问题。游戏使用二维数组存放障碍物、道具的相关数据，运行稳定，可靠性强。

【关键词】游戏开发；C++；Microsoft Visual Studio；EasyX 库

引言

当今社会，游戏已经成为了人们日常生活中不可或缺的一部分，从而催生了种类繁多的游戏。其中采用 Visual Studio 作为开发平台的游戏数不胜数。论文采用 Microsoft Visual Studio 2017 为开发平台，在已有的 EasyX 图形库的基础上构建游戏架构。

1 开发环境及技术

1.1 系统开发环境

操作系统：Microsoft Windows 10
程序语言：C++
开发平台：Microsoft Visual Studio 2017
开发工具：EasyX 图形库

1.2 EasyX 图形库

EasyX 是针对 C++ 的一个轻量级的图形库，相关 API 较少，可以帮助初学者快速上手图形和游戏编程。比如，VC + EasyX 可以很快地用几何图形画一个房子或者一辆移动的小车，可以编写俄罗斯方块、贪吃蛇、黑白棋等小游戏，可以练习图形学的各种算法。虽然在图形界面上 EasyX 与 QT、MFC 相差很远，但其在简洁的基础上具有部分可以操作图像的基础函数。EasyX 是一款方便计算机图形学初学者使用的库。[1]

2 系统设计

2.1 泡泡堂游戏简介

泡泡堂游戏是一款双人竞技类游戏，由两名玩家来分别操控两位人物角色，通过障碍物的爆炸刷新出可以增强人物属性的道具。每个玩家的目标就是让另一名玩家被爆炸的泡泡波及，每一次被波及时人物的血量都会降低。当一方的血量降低至小于等于 0 时，另一方胜利。

泡泡堂游戏界面如图 1 所示：

图 1 游戏界面

2.2 系统设计

根据游戏规则，将作用相似或者可以在某一流程统一处理的函数归为一个功能模块，将游戏分为 4 个功能模块。

2.2.1 菜单模块

菜单具有首因效应的重要作用，是用户对于泡泡堂游戏的第一印象。菜单模块提供开始游戏、游戏说明、退出游戏三种功能按键，引导用户进行操作选择。

图 2 所示的类图是对菜单类的详细定义。menu 主菜单类中的变量 x、y 表示主菜单中按钮的绘制位置，变量 choose 代表当前鼠标选择的功能。

图 2 UML 静态结构图

2.2.2 地图模块

本模块完成地图的刷新和绘制，游戏刚开始时需要在地图的指定位置刷新出特定的障碍物，并且还需要在人物释放道具后“隐藏”人物所破坏的障碍物，以便于玩家正常进行游戏。

2.2.3 人物模块

本模块实现人物位置、移动速度和炸弹攻击力等相关信息的存储，提供公共函数接口供道具模块调用，完成对人物属性的修改。在人物模块中，使用变量 x、y 存储相关的人物坐标信息，towards 变量存储人物的朝向信息，blood 变量存储人物的血量，speed 变量存储人物的移动速度。为了满足面向对象中封装的特性，所有人物相关的变量都被放置在 private 作用域下，如果需要对人物相关属性进行改变，只需要调用相应的函数修改即可。如通过相应的 showX() 函数来展示 X，changeX() 函数用来修改该人物对象的 X 坐标。

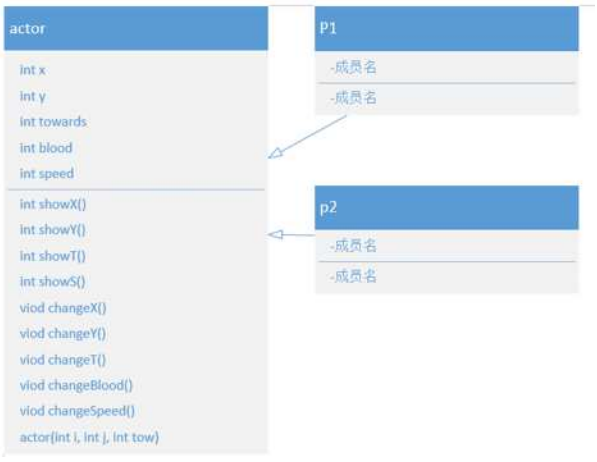

图 3 人物属性类图

2.2.4 道具模块

本模块是一个建立在人物模块和地图模块上的一个边缘模块，与这两个模块相互配合，通过检测人物所释放的炸弹爆炸是否生成道具，从而在地图上生成该道具。x、y 表示该道具放置的像素位置，通过 stateTrigger 来存储该道具在地图上的状态：true 表示已经释放，并且在下一次地图刷新时不显示该道具；false 表示未被触发，并且在地图刷新时通过修改地图数组来显示该道具。同时触发该道具时调用人物的修改属性函数，对人物的相关属性数值进行修改。类 bomb、increSpeed、increBlood 分别为道具的子类，实现相应的道具功能。

图 4 道具类图

3 系统实现

3.1 图像引擎的实现

首先初始化一个窗口，其次将所有 LoadImage（API 函数）都合并到一个初始化函数中，以增加代码的内聚性。当需要放置图片时直接调用 PutImage（API 函数），通过图像缩放将人物、炸弹、道具显示到指定的位置。

3.2 接口定义

函数命名时采用驼峰命名法，增强程序可读性的同时，增强了程序的规范性，并且将人物相关的变量隐藏起来，通过公开的函数访问变量，修改变量，增加了程序的安全性，代码的简洁性。

3.3 菜单模块的实现

菜单页面由背景图和“开始游戏”、“游戏背景”、“游戏帮助”、“退出游戏”4 个按钮组成。玩家可以通过单击按钮进入到相应的模块进行游戏或者阅读。

GetMouseMsg() 函数用来获取当前鼠标的信息并保存至 MOUSEMSG 类型的变量 m 中。通过对变量 m 的 x、y 值与 menu 类中用来指定标签位置的位置变量 x、y 的比对，确定鼠标的位置，继而实现鼠标操作。

将游戏所具有的四个模块分别封装到 4 个函数中，通过对鼠标位置的判断来选择相对应的功能。changemenuoff() 函数是“退出游戏”按键对应的函数实现，点击后关闭窗口并退出游戏。GameHelp() 函数用来显示游戏操作说明，内置一个箭头图标，通过点击该图

标即可返回上一级主菜单。GameProfiles() 函数用来展示游戏相关的背景，让用户具有代入感，增强用户体验。Welcomegame( ) 函数实现进入游戏界面，开始一场游戏。

以下代码是点击“开始游戏”按钮，进入游戏的代码。点击其他 3 个按钮的代码类似。

```
m = GetMouseMsg(); // 获取一条鼠标消息
    if (m.x >= 300 && m.x <= 500 && m.y >= 0 && m.y <= 150)
// 在这一区域内 开始游戏按钮会变色 {
    if (m.uMsg == WM_LBUTTONDOWN) // 点击进入游戏
    {
    changemenuon(2);
    Sleep(300);
    map();
    while (1) welcomegame();
        }
```

3.4 地图模块的实现

障碍物的状态会随着游戏进程的改变而改变，在地图模块中有专属的二维数组负责储存相关的地图障碍物的情况，用 1、2、3、4 这几个数来分别代表可以被破坏的障碍物、可以移动的空地、树和房子。再刷新地图时，通过这个数组对地图进行刷新，也即对这个地图现存的障碍物重新绘制。

3.5 人物模块的实现

将人物属性横坐标用 int 型变量 x 表示，纵坐标用 int 型变量 y 表示，朝向用 int 型变量 towards 表示，血量用 int 型变量 blood 表示，速度用 int 型变量 speed 表示，并封装在 private 作用域下。在 public 作用域下通过函数 show~() 来显示相关属性，函数 change~() 用于修改人物的相关属性。人物静止时调用绘制人物站立的函数 drawActorStand()。在控制人物移动的函数 KeyDown1() 中，通过按下按键触发调用绘制人物移动动画的函数 drawActor1Move()，实现人物的移动。

以下代码用于存储人物信息，修改人物属性：

```
class actor {
public:
    int showX() { return x; }
    int showY() { return y; }
    int showT() { return towards; }
    int showS() { return speed;}
    void changeX(int i) {          x += i; }
    void changeY(int i) {          y += i; }
    void changeT(int i) { towards = i; }
    actor(int i, int j, int tow)
    {
            x = i;
            y = j;
            towards = tow;
    }
    void ChangeBlood(int i) { blood = i; }
    void ChangeSpeed(int i) {     speed = i; }
private:
    int x;// 人物横坐标
    int y;// 人物纵坐标
    int towards;// 人物朝向
    int blood; // 人物血量
    int speed = 5;// 人物速度
};
```

drawActor1Move() 函数通过循环语句来逐帧显示人物动作以实现人物的移动动画。changeRow() 函数刷新出地图上放置现存的障碍物，最后通过函数 putimage() 显示人物的动作。moveRule1() 函数用来判断人物的移动是否合法，如不满足则不移动。

```
void drawActor1Move(int frameNum)//franmeNum 帧数
{
int CurRow = changePtoRowP1();
InitialFloor();
for (int i = 0; i < frameNum; ++i) {
changeRow();
putimage(p1.showX( ),p1.showY( ),40,40,&moveimg1Y,40*i,40*p1.showT( ),SRCAND);
putimage(p1.showX( ),p1.showY( ),40,40,&moveimg1,40*i,40*p1.showT( ),SRCPAINT);
putimage(p2.showX( ),p2.showY( ),40,40,&standimg2Y,40*p2.showT( ),0,SRCAND);
putimage(p2.showX( ),p2.showY( ),40,40,&standimg2,40*p2.showT( ),0,SRCPAINT);
moveRule1( );
Sleep(40);
}
}
```

drawActorStand() 是绘制人物站立图像的相关函数，通过传入人物对象的信息作为参数，绘制该人物的图像，并且同时根据他的不同朝向，使用函数 putimage() 画出人物不同的朝向。

```
void drawActorStand()
{
map();
```

```
putimage(p1.showX( ),p1.showY( ),40,40,&standimg1Y,
40*p1.showT( ),0, SRCAND);
putimage(p1.showX( ),p1.showY( ),40,40,&standimg1,
40*p1.showT( ),0, SRCPAINT);
putimage(p2.showX( ),p2.showY( ),40,40,&standimg2Y,
40*p2.showT( ),0,SRCAND);
putimage(p2.showX( ),p2.showY( ),40,40,&standimg2,
40*p2.showT( ),0,SRCPAINT);
}
```

函数 KeyDown1() 获取按键的输入，并且根据键盘的输入值，通过 switch 语句对该操作进行选择，分别通过 W、A、S、D 四个按键来分别控制人物 1 的上下左右移动的方向。在每一个移动方向中，首先调用对象的函数修改当前人物的朝向，然后再通过 drawActor1Move() 这个函数来绘制人物移动的图像，通过 SPACE 键释放炸弹，调用 showBomb1() 函数来绘制该炸弹。

```
void KeyDown1() // 角色 1 按键移动
{
    char Key1 = _getch(); // 获取用户按键信息
    Sleep(1);
    int CurRow = changePtoRowP1();
    switch (Key1) {
    case ‘ ‘: // 按空格放炸弹
            b1.changeX(p1.showX());// 第几个格子
            b1.changeY(p1.showY());
            showBomb1();
            break;
    }
case ‘A’:
    case ‘a’: // 向左移动
            p1.changeT(2) ;
            drawActor1Move(6);
            break;
    // 向右、向上、向下移动的代码类似
}
```

4 测试

采用白盒测试和手动测试的方法来对系统的运行进行测试。首先进行分模块测试，分别测试开始菜单模块、地图模块、人物模块、道具实现模块。

菜单模块：通过点按菜单目录下的每个按键是否能够稳定正常运行，来进行检测。经过测试后没有发现问题，每个按键点按能够正常交互。

地图模块：绘制出完整地图，但是由于贴图方式有待改进带来整个画面有时会有延迟的出现。

人物模块：通过控制人物上下左右移动的按键来移动人物，来检查人物移动方式，移动动画，攻击逻辑，攻击动画能否正常运行。检查完毕后，人物可以正常移动，有时炸弹的放置和动画会造成一定的偏移，并且有输入的延迟现象，有可能是输入到程序中很多命令没有及时的被执行所导致的缓冲区堵塞所造成的问题。

道具实现模块：通过控制人物释放炸弹破坏障碍物来使地图上随机刷新出道具，并且检查人物的属性能否通过道具来进行修改，通过上述方式检查后道具能够正常的刷新出来，并且拾取道具后能够成功的修改人物的属性，说明该模块能够正常完成目标功能。

5 结论

通过以上软件测试后，可得出以下的结论，本游戏在操作性上连贯流畅，画面精美，功能性上满足了刚开始的设计需求，性能方面足够胜任，内存占用以及 CPU 的使用情况都保持了非常不错的水平。符合游戏类别软件的设计要求，但并未设计远程多人对战的相关功能，有待进一步改进。

【参考文献】

[1] 为什么要用 EasyX？ [EB/OL]. [2009-02-22]. https://easyx.cn/readme/view.aspx?id=4.

[2] 武思齐 . 休闲游戏大白版泡泡堂的设计与实现 [J]. 电子技术与软件工程 ,2016(15):54-55.

[3] 甄冒发 , 李秀娟 , 洪斐 . 基于 VC 的游戏算法研究与设计 [J]. 机电工程技术 ,2012,41(04):74-78.