

基于 Qt4 的扫雷游戏的设计与实现

柯胜¹, 孙进¹, 邓志良²

(1.江苏科技大学,江苏 镇江 2120031; 2.常州信息职业技术学院,江苏 常州 213164)

摘要: 在简要介绍扫雷游戏主界面设计的基础上,给出了一种基于 Qt 的扫雷游戏的设计与实现方法,并在 Qt4.3.2 和 Red Hat Enterprise Linux 4 操作系统下成功实现。经过实验测试,结果正确,各项功能达到游戏要求。该实现方法对其他基于 Qt 的游戏开发起到抛砖引玉的作用,同时对各专业人员借助 Qt 快速开发具备强大计算功能的专业应用软件具有重要的意义。

关键词: Qt;扫雷游戏;事件;信号与槽

中图分类号: TP311.1

文献标识码: A

文章编号: 1674-7720(2012)16-0006-03

Design and realization of mine-sweeping game based on Qt4

Ke Sheng¹, Sun Jin¹, Deng Zhiliang²

(1. Jiangsu University of Science and Technology, Zhenjiang 212003, China;

2. Changzhou College of Information Technology, Changzhou 213164, China)

Abstract: On the basis of briefly introducing the design of minesweeper game interface, a design and implementation of mine-sweeper game which is based on Qt is given, and realizes in Qt4.3.2 and Red Hat Enterprise Linux 4 operating system. After the experimental tests, the results are correct, and it can reach the various functions of the game requires. The realization plays a valuable role in other Qt-based game development, while it is of great significance to the professionals who rapidly develop professional application software with the powerful computing capabilities by Qt.

Key words: Qt; mine-sweeper game; event; signals and slots

最初以训练鼠标操作为目的而设计的扫雷游戏是一款经久不衰的 Windows 平台休闲游戏。尽管 Windows 操作系统经历数次换代,变得越来越庞大、复杂,但这个可爱的小游戏在任何版本的 Windows 操作系统里却依然保持着原貌,几乎每个电脑使用者都接触过^[1]。

Qt 是诺基亚开发的一个跨平台的 C++ 图形用户界面应用程序框架。它为应用程序开发者提供建立艺术级的图形用户界面所需的所有功能。Qt 是完全面向对象的,很容易扩展,并且允许真正地组件编程。自 1996 年,Qt 进入商业领域,它已经成为全世界范围内数千种成功的应用程序的基础。Qt 也是流行的 Linux 桌面环境 KDE 的基础。基本上,Qt 与 X Window 上的 Motif、Openwin、GTK 等图形界面库和 Windows 平台上的 MFC、OWL、VCL、ATL 属同类型,但 Qt 具有优良的跨平台特性、面向对象、丰富的 API、大量的开发文档等优点^[2]。

本课题是在 Linux 系统下设计并开发的,设计了一款基于 Qt 环境的扫雷游戏,使用了 C++ 语言程序。

1 扫雷游戏主界面的设计

游戏主界面由菜单、游戏区、按钮区、信息显示区等几部分构成,如图 1 所示。Qt 提供了一套完整的 GUI 模块,能够完成基本的 Windows 窗体应用程序,因此可以简单地地为扫雷程序制作出界面^[3]。图 1 主要通过子类化 QMainWindow 创建扫雷游戏应用程序用户界面。Qt 还提供了定时器,能够完成游戏的计时。

2 鼠标事件的处理

当点击鼠标左键时,设置 ok_flag_ 为 true,说明此方块进行了翻开操作。如果方块是地雷,发出一个 explode()信号;如果不是地雷,发出一个 safe()信号,同时显示数字。这动作应当在 ok_flag_ 无效且 mark_flag_ 也无效的前提下进行,因为如果 ok_flag_ 有效,则说明此方块已经翻开了,没有必要重做;如果 mark_flag_ 有效,则说明玩家标志此方块有雷,不应该去翻开,否则即为自取灭亡。

点击鼠标右键进行旗帜安插或者移除操作应该在 ok_flag_ 无效的前提下进行,因为对于一个已经翻开的



图 1 Linux (Red Hat Enterprise Linux 4) 环境下的程序界面

方块, 安插毫无意义。

类的定义如下:

```
class BlockArea:public QWidget
{
    Q_OBJECT
public:
    BlockArea(QWidget* parent=0);
private slots:
    void slotSafe();
    void slotExplode();
private:
    int calculateMines(int x,int y)const; //计算以(x,y)为中心
    的九宫格内的雷数
private:
    QGridLayout* mainLayout;
    int row_;
    int column_;
    int total_block_number_;
    int total_mine_number_;
    int ok_block_number_;
};
```

下面是代码实现部分:

```
void
Block::mousePressEvent(QMouseEvent* event)
{
    if(event->button()==Qt::LeftButton)
    {
        if(ok_flag_==false&&mark_flag_==false)
        {
            ok_flag_=true;
            if(mine_flag_==true)
            {
                QPixmap(QPixmap(":/images/mine.png"));
            }
        }
    }
}
```

8

```
update();
emit explode();
}else{
    QPixmap(QPixmap(":/images/mine_"+QString("%1").
        arg
        (number_)+".png"));
    update();
    emit safe();
}
}
}else
if(event->button()==Qt::RightButton){
    if(ok_flag_==false){
        if(mark_flag_==false){
            mark_flag_=true;
            QPixmap(QPixmap(":/images/flag.png"));
        }else{
            mark_flag_=false;
            QPixmap(QPixmap(":/images/normal.png"));
        }
        update();
    }
}
```

设计的实现比较简单,需要说明的是本文模拟的鼠标事件并不是单击而是按下,这对于扫雷已经足够了。同时,会发现两个信号函数没有实现,这个工作会由 moc 自动完成,因此不必在.cpp 中实现,并且它们永远不会有返回值(即 void)。

3 初始化

3.1 雷区/非雷区的产生

下面是 BlockArea 的构造函数:

```
BlockArea::BlockArea(QWidget* parent)
:QWidget(parent)
{
    //下面 5 行来初始化 BlockArea 的信息,自定义行数为
    10、列数为 10、总格数为 100、雷数为 10。
    row_=10;
    column_=10;
    total_block_number_=row_*column_;
    total_mine_number_=10;
    ok_block_number_=0;
```

//下面 6 行生成一个具有 total_block_number_ 个元素的 bool 类型的随机序列,用来布雷(即确定哪些方块放雷,哪些不放),因为 QtAlgorithms 中没有包含打乱序列的算法,故采用 C++的 STL 里的 random_shuffle。

```
bool mine_flag[total_block_number_];
for(int i=0;i<total_mine_number_;i++)
    mine_flag[i]=true;
```

《微型机与应用》2012 年 第 31 卷 第 16 期

```
for(int i=total_mine_number_;i<total_block_number_;i++)
mine_flag[i]=false;
std::random_shuffle (mine_flag,mine_flag +total_block_number_);
```

```
//下面 4 行将方块放进布局
mainLayout=new QGridLayout(this);
for(int i=0;i<row_;i++)
for(int j=0;j<column_;j++)
mainLayout->addWidget(new Block (mine_flag [i*column_ +
j]),i,j);
```

//下面 6 行设置每个方块的周围雷数,并且将信号 safe()与槽 slotSafe()相联,信号 explode()与槽 slotExplode()相联。Block* current_block=static_cast<Block*>(mainLayout->itemAtPosition(i,j)->widget());这句话是获取布局中位于(i,j)位置处的 widget*, 将其转换为 Block*,static_cast 是 C++关键字,这里执行一个下行转换动作。

```
for(int i=0;i<row_;i++){
for(int j=0;j<column_;j++){
Block* current_block=static_cast<Block*>(mainLayout->
itemAtPosition(i,j)->widget());
current_block->setNumber(calculateMines(i,j));
connect(current_block,SIGNAL(safe()),this,SLOT(slotSafe
()));
connect(current_block,SIGNAL(explode()),this,SLOT(slot
Explode()));
}
}
```

3.2 周边雷数的计算

要计算(x,y)位置周边地雷的个数,首先需计算出一个点,然后判断此点是否落在 BlockArea 中,如果落在 BlockArea 中,再判断是否是雷,如果是,则计数器加 1。

```
int BlockArea::calculateMines(int x,int y)const
{
int number=0;
for(int i=0;i<3;i++)
for(int j=0;j<3;j++)
if( (x-1+i>=0) && (x-1+i<row_) && (y-1+j>=0)
&& (y-1+j<column_)
```

```
if(static_cast<Block*>(mainLayout->itemAtPosition(x-
1+i,y-1+j)->widget()->isMine())
++number;
return number;
}
```

该扫雷游戏在 Qt4 和 Red Hat Enterprise Linux 4 操作系统上成功实现,能在 Windows 和 Linux 平台下运行。除能实现基本的左键打开、右键标记的扫雷功能以外,还能实现计时、自定义游戏难度、作弊、语音提示等扩展功能。经过试验测试,界面美观,结果正确,各项功能达到游戏要求。该实现方法对其他基于 Qt 的游戏开发起到了抛砖引玉的作用,其中的键盘、鼠标等功能的实现方法可用于其他 Qt 编程,同时对各专业人员借助 Qt 快速开发具备强大计算机功能的专业应用软件具有重要的意义。

参考文献

- [1] 陈子为.基于 Matlab GUI 扫雷游戏的设计与实现[J].现代电子技术,2008(24):85-88.
- [2] BLANCHETTE J,SUMMERFIELD M.C++ GUI QT4 编程(第二版)[M].北京:电子工业出版社,2008.
- [3] 张建强,张秀梅.扫雷游戏策略初探[J].数学教学,2004(6):32-33.
- [4] 成洁,卢紫毅.Linux 窗口程序设计——Qt 精彩实例分析[M].北京:清华大学出版社,2008.
- [5] 赵大伟,肖周芳,张艳.从扫雷游戏浅谈一些算法问题[J].科技信息,2008(29):69.
- [6] 钱会敏,于守秋.自动扫雷算法浅谈[J].科技创新导报,2009(31):250.
- [7] 刘艳青,苏桂莲.基于 Qt4 的图形用户界面程序的设计与实现[J].现代计算机(专业版),2009(3):170-172.

(收稿日期:2012-04-21)

作者简介:

柯胜,男,1987 年生,研究生,主要研究方向:嵌入式应用开发。

孙进,男,1988 年生,研究生,主要研究方向:嵌入式应用开发。

邓志良,男,1962 年生,博士,教授,硕士生导师,主要研究方向:控制理论与控制工程。