开源技术及应用

基于Linux平台下的小游戏俄罗斯方块

姓名 刘 厶 玮

学号 3014216068

专业 计算机科学与技术

班级 2014级 3班

天津大学计算机科学与技术学院

2016年12月6日

目录

一、问题描述

任意选择与本课程内容有关的需求，基于开源软件技术，开发一个具有特定功能的软件。

二、问题分析

由于基于开源软件技术，所以选择在操作系统Linux的环境下。通过对代码的学习，开发一款经典的俄罗斯方块小游戏，以C++为语言，使用QT平台。

三、开发平台与工具

开发平台：Ubuntu16.04

开发工具：QT5

四、前期准备

首先需具备Linux操作系统，安装16.04版本的Ubuntu。熟悉在Linux下的各种操作。

在Ubuntu下安装QT5 for Linux，能熟练地在QT下使用c++，了解并能够使用QT的信号槽等。

学习已经开源的俄罗斯方块，掌握基本框架。并对其有自己的理解，适当改进。

五、功能说明

俄罗斯方块游戏规则概述:移动、旋转游戏自动输出的各种方块，使之排列成完整的一行或多行，此时排列完整的方块可被消除得分。

开始：开始新的一局，积分清零，方块随机出现下落，并显示下一方块形状。

暂停：方块停止运动，再次按键游戏继续。

退出：退出游戏。

下一个：显示下一个方块形状。

分数：根据消除行数获得积分。

级别：根据积分升级，重新开始后，级数不变。

已满行数：已消除行数。

六、主要功能实现（代码）

6.1 设定键盘触发方块移动、旋转

```
void TetrixBoard::keyPressEvent(QKeyEvent *event)
{
    if (!isStarted || isPaused || curPiece.shape() == NoShape) {
        QFrame::keyPressEvent(event);
        return;
    }
    switch (event->key()) {
    case Qt::Key_Left://键盘向左，使方块左移
        tryMove(curPiece, curX - 1, curY);
        break;
    case Qt::Key_Right://键盘向右，使方块右移
        tryMove(curPiece, curX + 1, curY);
        break;
    case Qt::Key_Down://键盘向下，使方块顺时针旋转
        tryMove(curPiece.rotatedRight(), curX, curY);
        break;
```

```
    case Qt::Key_Up://键盘向上，使方块逆时针旋转
        tryMove(curPiece.rotatedLeft(), curX, curY);
        break;
    case Qt::Key_Space://空格键，使方块坠落
        dropDown();
        break;
    default:
        QFrame::keyPressEvent(event);
    }
}
```

6.2 设置方块形状

```
void TetrixPiece::setShape(TetrixShape shape)
{
    static const int coordsTable[8][4][2] = {
        { { 0, 0 },   { 0, 0 },   { 0, 0 },   { 0, 0 } },
        { { 0, -1 },  { 0, 0 },   { -1, 0 },  { -1, 1 } },
        { { 0, -1 },  { 0, 0 },   { 1, 0 },   { 1, 1 } },
        { { 0, -1 },  { 0, 0 },   { 0, 1 },   { 0, 2 } },
        { { -1, 0 },  { 0, 0 },   { 1, 0 },   { 0, 1 } },
        { { 0, 0 },   { 1, 0 },   { 0, 1 },   { 1, 1 } },
        { { -1, -1 }, { 0, -1 },  { 0, 0 },   { 0, 1 } },
        { { 1, -1 },  { 0, -1 },  { 0, 0 },   { 0, 1 } }
    };
    for (int i = 0; i < 4 ; i++) {
        for (int j = 0; j < 2; ++j)
            coords[i][j] = coordsTable[shape][i][j];
    }
    pieceShape = shape;
}
```

6.3 左右旋转时的变化

```
TetrixPiece TetrixPiece::rotatedLeft() const
{
    if (pieceShape == OShape)
        return *this;
    TetrixPiece result;
    result.pieceShape = pieceShape;
    for (int i = 0; i < 4; ++i) {
        result.setX(i, y(i));
        result.setY(i, -x(i));
    }
```

```
    return result;
}
TetrixPiece TetrixPiece::rotatedRight() const
{
    if (pieceShape == OShape)
        return *this;
    TetrixPiece result;
    result.pieceShape = pieceShape;
    for (int i = 0; i < 4; ++i) {
        result.setX(i, -y(i));
        result.setY(i, x(i));
    }
    return result;
}
```

6.4 方块的消除

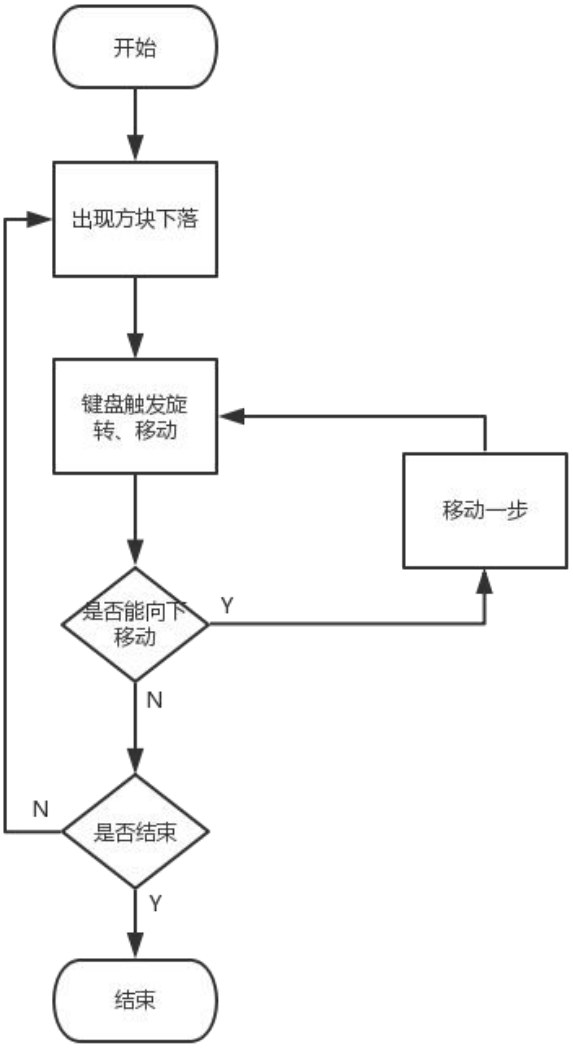
```
void TetrixBoard::removeFullLines()
{
    int numFullLines = 0;
    for (int i = BoardHeight - 1; i >= 0; --i) {
        bool lineIsFull = true;
        for (int j = 0; j < BoardWidth; ++j) {
            if (shapeAt(j, i) == NoShape) {
                lineIsFull = false;
                break;
            }
        }
          if (lineIsFull) {
            ++numFullLines;
            for (int k = i; k < BoardHeight - 1; ++k) {
                for (int j = 0; j < BoardWidth; ++j)
                    shapeAt(j, k) = shapeAt(j, k + 1);
            }
            for (int j = 0; j < BoardWidth; ++j)
                shapeAt(j, BoardHeight - 1) = NoShape;
        }//若仍有满行则增加满行的个数，同时销毁掉这些行数，将其上的各
行下移
    }
        if (numFullLines > 0) {
        numLinesRemoved += numFullLines;
        score += 10 * numFullLines;//每消去一行分数加上 10
        emit linesRemovedChanged(numLinesRemoved);
        emit scoreChanged(score);
```

```
        timer.start(500, this);
        isWaitingAfterLine = true;
        curPiece.setShape(NoShape);
        update();
    }
}
```

6.5 设置下一方块

```
void TetrixBoard::showNextPiece()
{
    if (!nextPieceLabel)
        return;
    int dx = nextPiece.maxX() - nextPiece.minX() + 1;
    int dy = nextPiece.maxY() - nextPiece.minY() + 1;
    //用来设置下一个方块的坐标范围
    QPixmap pixmap(dx * squareWidth(), dy * squareHeight());
    QPainter painter(&pixmap);
    painter.fillRect(pixmap.rect(),
nextPieceLabel->palette().background());
    for (int i = 0; i < 4; ++i) {
        int x = nextPiece.x(i) - nextPiece.minX();
        int y = nextPiece.y(i) - nextPiece.minY();
        drawSquare(painter, x * squareWidth(), y * squareHeight(),
                   nextPiece.shape());
    }
    nextPieceLabel->setPixmap(pixmap);
}
```

七、流程图



八、软件截图

8.1 初始状态

8.2 消除前后


我的俄罗斯方块
下一个
分数
10
级别
已满行数
1
1
开始
退出
暂停

8.3 暂停


我的俄罗斯方块
下一个
分数
10
级别
已满行数
暂停
1
1
开始
退出
暂停

8.4 重新开始

九、总结

通过此次作业,不仅使我对编程方面取得了进步,更让我对Linux操作系统有了进一步的了解。让我在Linux下有了再一次实践的机会。此次程序的成功是建立在虚拟机成功安装之上的，幸而在学期初就开始着手Ubuntu的安装与使用，使得我在此次大作业中节省了许多时间。

在对开源软件的代码理解过程中，遇到过许多困难，由于QT并不是我经常接触到的软件，对于其中一些语句需要逐句上网搜索含义。自己对这个软件重写时也遇到各种各样的错误，但逐一修改后，发现自己对程序的理解，语言的运用更深了一步。

临近期末，课程渐进尾声，我对开源有了初步的认识，对Linux系统也产生了浓厚的兴趣。希望在接下来的时间能够对他们有更深入的理解和交流。