#ifndef SNAKE\_H

#define SNAKE\_H

#include <QMainWindow>

#include<qpainter.h>

#include<QVector>

#include<qstring.h>

namespace Ui {

class Snake;

}

class Snake : public QMainWindow

{

Q\_OBJECT

public:

explicit Snake(QWidget \*parent = 0);

~***Snake***();

void ***paintEvent***(QPaintEvent \*event);

void **InitSnake**();//初始化蛇

QRect **CreateRect**();

void **IsEat**();

void **IsHit**();

private slots:

void **Snake\_update**();//定义槽函数

private:

Ui::Snake \*ui;

QVector<QRect> vSnakeRect;//蛇

QRect SnakeHead;//蛇头

void ***keyPressEvent***(QKeyEvent \*key);//定义键盘事件

QTimer \*timer;//定时器

bool blsRun;//是否开始

bool blsOver;//是否结束

QString sDisplay;//开始结束提示

int nDirection;//蛇移动的方向

QRect food;//食物

int nScore;//得分

QString scoreLabel;//显示得分两个字

int speed;//移动速度

};

#endif // SNAKE\_H

#include "snake.h"

#include "ui\_snake.h"

#include<qrect.h>

#include<qtimer.h>

#include<QKeyEvent>

#include<QDebug>

#include<QFont>

#include<QPixmap>

Snake::**Snake**(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::Snake),blsRun(false),speed(500)

{

ui->setupUi(this);

this->setGeometry(QRect(600,300,290,310));

}

Snake::~***Snake***()

{

delete ui;

}

void Snake::***paintEvent***(QPaintEvent \*event){//update()或程序第一次启动时自动执行，用于画图

QPainter painter(this);

if(!blsRun){//只有blsRun为false时才需要初始化蛇

InitSnake();//初始化蛇，此时蛇还没有画，只是初始化了矩形，没有显示

}

//画游戏背景

//外墙

painter.setPen(Qt::black);//设置画笔颜色

painter.setBrush(Qt::gray);//设置填充颜色

painter.drawRect(15,15,260,260); //画矩形，相对于窗体的起点坐标（15，15），大小260\*260

//内墙

painter.setPen(Qt::black);//设置画笔颜色

painter.setBrush(Qt::black);//设置填充颜色

painter.drawRect(20,20,250,250); //画矩形，相对于窗体的起点坐标（20，20），大小250\*250

//画小格子，大小10\*10

painter.setPen(Qt::blue);//设置画笔颜色

for(int i=2;i<=27;i++){

painter.drawLine(20,i\*10,270,i\*10);//画横线，270是根据坐标换算出来的

painter.drawLine(i\*10,20,i\*10,270);//画竖线

}

//显示游戏开始、结束

QFont font1("Courier",24);

painter.setFont(font1);

painter.setPen(Qt::red);

painter.setBrush(Qt::red);

painter.drawText(40,150,sDisplay);

//得分显示

QFont font2("Courier",15);

painter.setFont(font2);

painter.setPen(Qt::blue);

painter.setBrush(Qt::blue);

painter.drawText(140,300,scoreLabel);

painter.drawText(230,300,QString::number(nScore));

//画蛇

painter.setPen(Qt::black);//设置画笔颜色

painter.setBrush(Qt::green);//设置填充颜色

painter.drawRect(food);

painter.drawRects(&vSnakeRect[0],vSnakeRect.size());//画n个小方块

if(blsOver)//游戏结束，停止计时器

timer->stop();

}

void Snake::**InitSnake**(){

nDirection=2;//默认向下运动

blsRun=true;

blsOver=false;

sDisplay="游戏开始";

scoreLabel="得分：";

nScore=0;

food=CreateRect();//创建食物

// QRect rect(100,70,10,10);//蛇的初始位置

// vSnakeRect=rect;//将矩形赋值给蛇

vSnakeRect.resize(5);//初始化snake的长度是5

//利用一个for循环初始化vector

for(int i=0;i<vSnakeRect.size();i++){

QRect rect(100,70+10\*i,10,10);//生成小方块

vSnakeRect[vSnakeRect.size()-1-i]=rect;//小方块赋值到容器

}

SnakeHead=vSnakeRect.first();//指定蛇头位置

timer=new QTimer(this);//设定计时器

timer->start(speed);//计时器时间是0.5s

connect(timer,SIGNAL(timeout()),SLOT(Snake\_update()));//信号连接槽，发出信号的是timer，信号是timeout，执行的槽是Snake\_update

}

void Snake::**Snake\_update**(){

sDisplay="";

SnakeHead=vSnakeRect.first();//获取到蛇头

IsEat();

IsHit();

//让数组移动，除了蛇头，后面的到前面的位置，因为蛇头是控制方向的

for(int j=0;j<vSnakeRect.size()-1;j++){

vSnakeRect[vSnakeRect.size()-1-j]=vSnakeRect[vSnakeRect.size()-2-j];

}

//下面我们要根据方向，来对蛇头小方块的移动方向进行调整

switch(nDirection)

{

case 1:

SnakeHead.setTop(SnakeHead.top()-10);

SnakeHead.setBottom(SnakeHead.bottom()-10);

break;

case 2:

SnakeHead.setTop(SnakeHead.top()+10);

SnakeHead.setBottom(SnakeHead.bottom()+10);

break;

case 3:

SnakeHead.setLeft(SnakeHead.left()-10);

SnakeHead.setRight(SnakeHead.right()-10);

break;

case 4:

SnakeHead.setLeft(SnakeHead.left()+10);

SnakeHead.setRight(SnakeHead.right()+10);

default:;

}

vSnakeRect[0]=SnakeHead;//第一个节点设置为蛇头位置

if(SnakeHead.left()<20||SnakeHead.right()>270||SnakeHead.top()<20 || SnakeHead.bottom()>270){

sDisplay="游戏结束";

blsOver=true;

}

update();//paintEvent更新

}

void Snake::***keyPressEvent***(QKeyEvent \*event){

QKeyEvent \*key=(QKeyEvent\*)event;

switch (key->key()) {

case Qt::Key\_Up:nDirection=1; //1代表上

break;

case Qt::Key\_Down:nDirection=2; //2代表下

break;

case Qt::Key\_Left:nDirection=3; //3代表左

break;

case Qt::Key\_Right:nDirection=4; //4代表右

break;

default:;

}

}

QRect Snake::**CreateRect**(){

int x,y;

x=qrand()%25;//生成小于25的整数

y=qrand()%25;

QRect rect(20+x\*10,20+y\*10,10,10);//食物小方块位置

return rect;

}

void Snake::**IsEat**(){

if(SnakeHead==food){//如果蛇头和食物重合，说明已经撞到食物了

SnakeHead=food;//吃到食物，食物就变为蛇头

vSnakeRect.push\_back(vSnakeRect.last());//将最后一个节点再次加到容器里，让蛇身变长

food=CreateRect();//吃到食物后再生成一个食物

nScore+=10;//得分

if(speed>50){

speed=speed-10;

timer->stop();//停止当前计时器

timer->start(speed);//设置新的计时器

}

}

}

void Snake::**IsHit**(){

for(int i=1;i<vSnakeRect.size();i++){//遍历蛇身

if(SnakeHead==vSnakeRect[i]){//蛇头和蛇身重合

sDisplay="游戏结束";

blsOver=true;

update();

}

}

}