**大作业实现文档**

李畅 财税实验班 2018200047

# 一、负责工作与核心代码

我负责我们小组的基础场景搭建、按钮实现、关卡设置、调整图片素材、播放背景音乐、图片移动处理等工作。因为是组长，除统筹安排之外还在多方面帮助组员调试。

## （一）基础场景搭建

直接继承QWidget，核心代码如下：

namespace Ui {

Widget::**Widget**(QWidget \*parent) :

QWidget(parent),

ui(new Ui::Widget)

{

ui->setupUi(this);

this->setMouseTracking(true);

this->setWindowTitle("细胞大战细菌");

this->setWindowIcon(QIcon(":/Cells\_VS\_Virus.png"));

this->resize(1366,768);

// QPalette palete;

palete.setBrush(QPalette::Background,QBrush(QPixmap(":/start.jpg").scaled(this->size())));//图片和缩放模式

this->setPalette(palete);

}

Role::**Role**(QWidget \*parent):QWidget(parent)

{

//游戏背景布置

this->setMouseTracking(true);

this->setAutoFillBackground(true);

this->setWindowTitle("细胞大战细菌");

this->setWindowIcon(QIcon(":/Cells\_VS\_Virus.png"));

this->resize(1366,768);

QPalette palete;

palete.setBrush(QPalette::Background,QBrush(QPixmap(":/background\_cellVSvirus.jpg").scaled(this->size())));//图片和缩放模式

this->setPalette(palete);

}

## （二）按钮实现

利用槽函数构建startbtnClicked（）函数，调用QPushbutton库，连接widget界面和游戏界面role。

//第一关开始按钮

startBtn=new QPushButton(this);

startBtn->setIcon(QIcon(":/ClickToStart1.jpg"));

startBtn->setIconSize(QSize(525,43));

startBtn->setGeometry(QRect(414,635,545,55));

startBtn->setFlat(true);

connect(startBtn,SIGNAL(clicked(bool)),this,SLOT(startbtnClicked()));

//第二关开始按钮

startBtn2=new QPushButton(this);

startBtn2->setIcon(QIcon("://ClickToStart2.jpg"));

startBtn2->setIconSize(QSize(525,43));

startBtn2->setGeometry(QRect(414,700,545,55));

startBtn2->setFlat(true);

connect(startBtn2,SIGNAL(clicked(bool)),this,SLOT(startbtnClicked2()));

Role \*game;

Role2 \*game2;

void Widget::**startbtnClicked**()

{

game=new Role(this);

game->show();

}

void Widget::**startbtnClicked2**()

{

game2=new Role2(this);

game2->show();

}

//返回按钮封装

backBtn=new QPushButton(this);

backBtn->setIcon(QIcon(":/BACK.jpg"));

backBtn->setIconSize(QSize(100,63));

backBtn->setGeometry(QRect(1250,700,100,63));

backBtn->setFlat(true); connect(backBtn,SIGNAL(clicked(bool)),this,SLOT(backbtnClicked()));

Widget \*g;

void Role::**backbtnClicked**()

{

this->close();

QPalette palete;

palete.setBrush(QPalette::Background,QBrush(QPixmap(":/start.jpg").scaled(this->size())));//图片和缩放模式

this->setPalette(palete);

bgmusic->stop();

g=new Widget(this);

g->show();

}

## (三)关卡设置

将role作为虚基类，使role2继承第一关。在role2头文件中增设病毒数量。

class Role2:virtual public Role

{

Q\_OBJECT

public:

explicit Role2 (QWidget \*parent = 0);

~***Role2***();

public:

virus1 v1,v2,v3,v4,v5,v6,v8,v10,v13,v15,v18;

virus2 v7,v9,v14,v16;

virus3 v11,v12,v17;

QString VirusStr[20] = {" ",v1.virusstr,v2.virusstr,v3.virusstr,v4.virusstr,v5.virusstr,v6.virusstr,v7.virusstr,

v8.virusstr,v9.virusstr,v10.virusstr,

v11.virusstr,v12.virusstr,v13.virusstr,v14.virusstr,v15.virusstr,v16.virusstr,v17.virusstr,v18.virusstr};

QString Attackstr[20] = {" ",v1.virusstr,v2.virusstr,v3.virusstr,v4.virusstr,v5.virusstr,v6.virusstr,v7.virusstr,

v8.virusstr,v9.virusstr,v10.virusstr,

v11.virusstr,v12.virusstr,v13.virusstr,v14.virusstr,v15.virusstr,v16.virusstr,v17.virusstr,v18.virusstr};

//病毒属性

int Virus\_x[20]={0,1200,1490,1500,1490,1553,2043,2102,2230,2133,3022,3290,3265,3270,3260,3290,3265,3270,3260}; //病毒初始横坐标

int Virus\_line[20] = {0,1,2,3,1,2,3,1,2,3,3,3,2,2,2,1,1,1,1}; //病毒所在

int VirusSIZE\_X[20]={0,v1.virussizex,v2.virussizex,v3.virussizex,v4.virussizex,v5.virussizex,v6.virussizex,v7.virussizex,v8.virussizex,v9.virussizex,v10.virussizex,v11.virussizex,v12.virussizex,v13.virussizex,v14.virussizex,v15.virussizex,v16.virussizex,v17.virussizex,v18.virussizex};

int VirusSIZE\_Y[20]={0,v1.virussizey,v2.virussizey,v3.virussizey,v4.virussizey,v5.virussizey,v6.virussizey,v7.virussizey,v8.virussizey,v9.virussizey,v10.virussizey,v11.virussizey,v12.virussizey,v13.virussizey,v14.virussizey,v15.virussizey,v16.virussizey,v17.virussizey,v18.virussizey};

int VirusSpeed[20]={0,v1.virusspeed,v2.virusspeed,v3.virusspeed,v4.virusspeed,v5.virusspeed,v6.virusspeed,v7.virusspeed,v8.virusspeed,v9.virusspeed,v10.virusspeed,v11.virusspeed,v12.virusspeed,v13.virusspeed,v14.virusspeed,v15.virusspeed,v16.virusspeed,v17.virusspeed,v18.virusspeed};

int Virus\_num=1; //病毒计数

int Virusblood[20]={0,v1.virusblood,v2.virusblood,v3.virusblood,v4.virusblood,v5.virusblood,v6.virusblood,v7.virusblood,v8.virusblood,v9.virusblood,v10.virusblood,v11.virusblood,v12.virusblood,v13.virusblood,v14.virusblood,v15.virusblood,v16.virusblood,v17.virusblood,v18.virusblood};

};

Role2::**Role2**(QWidget \*parent):Role(parent){

//换背景图

QPalette palete;

palete.setBrush(QPalette::Background,QBrush(QPixmap(":/background2\_cellVSvirus.png").scaled(this->size())));//图片和缩放模式

this->setPalette(palete);

}

## （四）调整图片素材

使用**PicsArt，PS**等软件进行抠图、调整图片大小。

## （五）播放背景音乐

在工程文件中添加QT += multimedia，利用QSound播放音乐。

//播放音乐

QSound \*bgmusic = new QSound(":/background\_music\_cellVSviru.wav",this);

bgmusic->play();

bgmusic->setLoops(-1);

## （六）图片移动

以QLabel类实现动态加载图片效果。以鼠标move事件来移动图片。

QLabel \*BCellPicture;

QLabel \*WhiteCellPicture;

QLabel \*B\_CellPicture[12];

QLabel \*White\_CellPicture[12];

//图片移动处理

int BCell\_i=1,WhiteCell\_i=1; //每一种细胞图片的个数

int Picture\_x=-76,Picture\_y=-114; //透明图片移出窗口

int flag[3]={0}; //每一种细胞图片的标记

int move1=0; //判断哪一个细胞图片随鼠标移动

int PictureDark[3]={0}; //细胞图片阻止变为彩色

int BCellY[12]={0},WhiteCellY[12]={0}; //针对每一种细胞图片，刷新图片尺寸

int BCell\_pocx=65,WhiteCell\_pocx=65; //加载图片时由于尺寸增加导致的Y方向的下降的初始位置及其变化为差值

int BCell\_clock=0,WhiteCell\_clock=0;

void Role::**WhiteCellCardupload**(QString &str, int m)

{

WhiteCell\_clock=1;

White\_CellPicture[m]->setGeometry(QRect(8+117,WhiteCell\_pocx,76,WhiteCellY[m]));

Img->load(str);

White\_CellPicture[m]->setPixmap(QPixmap::fromImage(\*Img));

}

void Role::**BCellCardupload**(QString &str, int m)

{

BCell\_clock=1;

B\_CellPicture[m]->setGeometry(QRect(84+117,BCell\_pocx,76,BCellY[m]));

Img->load(str);

B\_CellPicture[m]->setPixmap(QPixmap::fromImage(\*Img));

}

void Role::***paintEvent***(QPaintEvent \*)

{

QString s1[2] = {":/whitecellBoard.jpg",":/B cell board.jpg"}; //board

QString s2[2] = {":/whitecellBoard.jpg",":/B cell board.jpg"};

QString s3[2] = {":/whitecellBoard.jpg",":/B cell board.jpg"};

paint=new QPainter;

//绘制木板

paint->begin(this);

QPixmap mapw(":/woodboard.jpg");

paint->drawPixmap(QRect(117,0,170,130),mapw);

//绘制血量木板

QPixmap mapsb(":/bloodborad.jpg");

paint->drawPixmap(QRect(0,0,117,125),mapsb);

//绘制卡片

QString str;

if(flag[1] == 0)

str = s1[0];

else if(flag[1] == 1)

str = s2[0];

else if(flag[1]==2)

{

str = s3[0];

WhiteCellCardupload(s1[0],WhiteCell\_i);

}

QPixmap map1(str);

paint->drawPixmap(QRect(8+117,8,76,114),map1);

if(flag[2]==0)

str = s1[1];

else if(flag[2]==1)

str = s2[1];

else if(flag[2]==2)

{

str = s3[1];

BCellCardupload(s1[1],BCell\_i);

}

QPixmap map2(str);

paint->drawPixmap(QRect(84+117,8,76,114),map2);

paint->end();

update();

}

bool Role::**CellLocalconclude**(int x, int y)

{

if(x>=0&&x<=245||y>=0&&y<=232||x>=953||y>=622)

return false;

else

{

for(int j=0;j<3;j++)

if(y>=(232+j\*131)&&y<=(232+(j+1)\*131))

{

for(int k=0;k<9;k++)

{

if(x>=(245+k\*79)&&x<=(245+(k+1)\*79))

{

if(local[j+1][k+1]==1)

return false;

}

}

}

}

return true;

}

int Role::**cardLocal**(int x, int y)

{

if(y>=8&&y<=121)

{

if(x>=8+117&&x<=84+117)

return 1;

else if(x>=84+117&&x<=160+117)

return 2;

}

else

return 0;

}

void Role::***mouseMoveEvent***(QMouseEvent \*event)

{

pos1 = cursor().pos();

QPoint pos = mapFromGlobal(pos1);

if(pos.rx()>=(8+117)&&pos.rx()<=(692+117)&&pos.ry()>=8&&pos.ry()<=121)

setCursor(Qt::OpenHandCursor);

else

setCursor(Qt::ArrowCursor);

if(pos.rx()>=(117+8)&&pos.rx()<=(84+117)&&pos.ry()>=8&&pos.ry()<=121&&flag[1]!=2)

{

flag[1]=1;

if(music[1]==0)

QSound::play("://mousemove~1.wav");

music[1]++;

}

else

{

if(PictureDark[1]==0)

flag[1]=0;

music[1]=0;

}

if(pos.rx()>=84+117&&pos.rx()<=117+160&&pos.ry()>=8&&pos.ry()<=121&&flag[2]!=2)

{

flag[2]=1;

if(music[2]==0)

QSound::play("://mousemove~1.wav");

music[2]++;

}

else

{

if(PictureDark[2]==0)

flag[2]=0;

music[2]=0;

}

QPoint s = event->pos();

if(move1==1)

{

PictureDark[1]=1;

WhiteCellPicture->move(s);

}

if(move1==2)

{

PictureDark[2]=1;

BCellPicture->move(s);

}

}

void Role::***mousePressEvent***(QMouseEvent \*event)

{

pos1 = cursor().pos();

QPoint pos = mapFromGlobal(pos1);

//改变鼠标图标指令

if(pos.rx()>=8+117&&pos.rx()<=692+117&&pos.ry()>=8&&pos.ry()<=121&&move1==0)

{ setCursor(Qt::ClosedHandCursor);}

//点击图片的指令

if(pos.rx()>=8+117&&pos.rx()<=84+117&&pos.ry()>=8&&pos.ry()<=121&&move1==0&&flag[1]!=2&&BloodNum>=100)

{

if(event->button()==Qt::LeftButton)

move1=1;

}

if(pos.rx()>=84+117&&pos.rx()<=160+117&&pos.ry()>=8&&pos.ry()<=121&&move1==0&&flag[2]!=2&&BloodNum>=150)

{

if(event->button()==Qt::LeftButton)

move1=2;

}

//种植植物的鼠标指令

if(move1!=0&&CellLocalconclude(pos.rx(),pos.ry()))

{

if(event->button()==Qt::LeftButton)

{

if(move1==1&&BloodNum>=100)

{

draw(p[w],Cellstr[1],pos.rx(),pos.ry(),CellSIZE\_X[1],CellSIZE\_Y[1]);

flag[1]=2;

WhiteCellPicture->move(Picture\_x,Picture\_y);

WhiteCellnum++;

w++;

}

else if(move1==2&&BloodNum>=150)

{

draw(p[w],Cellstr[2],pos.rx(),pos.ry(),CellSIZE\_X[2],CellSIZE\_Y[2]);

flag[2]=2;

BCellPicture->move(Picture\_x,Picture\_y);

BCellnum++;

w++;

}

move1=0;

}

else if(event->button()==Qt::RightButton)

{

if(move1==1)

{

WhiteCellPicture->move(Picture\_x,Picture\_y);

flag[1]=0;

PictureDark[1]=0;

}

else if(move1==2)

{

BCellPicture->move(Picture\_x,Picture\_y);

flag[2]=0;

PictureDark[2]=0;

}

move1=0;

}

}

}

void Role::**BCellPicupload**()

{

B\_CellPicture[BCell\_i]->move(0,BCell\_pocx);

if(BCellY[BCell\_i]<114&&BCell\_clock==1)

{

BCell\_pocx-=1;

BCellY[BCell\_i]+=2;

}

if(BCellY[BCell\_i]==114)

{

B\_CellPicture[BCell\_i]->move(-76,-114);

BCell\_i++;

BCell\_pocx=65;

flag[2]=0;

PictureDark[2]=0;

}

BCell\_clock=0;

}

void Role::**WhiteCellPicupload**()

{

White\_CellPicture[WhiteCell\_i]->move(0,WhiteCell\_pocx);

if(WhiteCellY[WhiteCell\_i]<114&&WhiteCell\_clock==1)

{

WhiteCell\_pocx-=1;

WhiteCellY[WhiteCell\_i]+=2;

}

if(WhiteCellY[WhiteCell\_i]==114)

{

White\_CellPicture[WhiteCell\_i]->move(-76,-114);

WhiteCell\_i++;

WhiteCell\_pocx=65;

flag[1]=0;

PictureDark[1]=0;

}

WhiteCell\_clock=0;

}

# 二、UML类图

