Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет

телекоммуникаций и информатики»

Кафедра прикладной математики и кибернетики

Практическая работа №7 по дисциплине

«Человеко-машинное взаимодействие»

Выполнил студент группы ИП-811

Адов А. C.

Проверил преподаватель

Мерзлякова Е. Ю.

Новосибирск, 2020

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №7**

Цель: Научиться работать с графикой в Qt

Требование: Задание по вариантам. Выполняется по подгруппам 1-2 человека.

Можно придумать свой вариант, предварительно согласовав с преподавателем.

Задание:

1. Создать графическую сцену.

2. Поместить на сцену различные элементы для составления картинки по

варианту. Обязательно использовать и геометрические фигуры, и картинки. Они

должны перемещаться с помощью мыши.

3. Ограничить края сцены «стенами» в виде каких-либо элементов.

4. Поместить на сцену движущийся элемент по заданию. Он должен

перемещаться с заданной скоростью, сталкиваться со «стенами» и фигурами на

сцене. Используйте таймер и функцию обнаружения столкновений.

**Вариант 10 Человек и движущийся телефон**

**Результат работы**

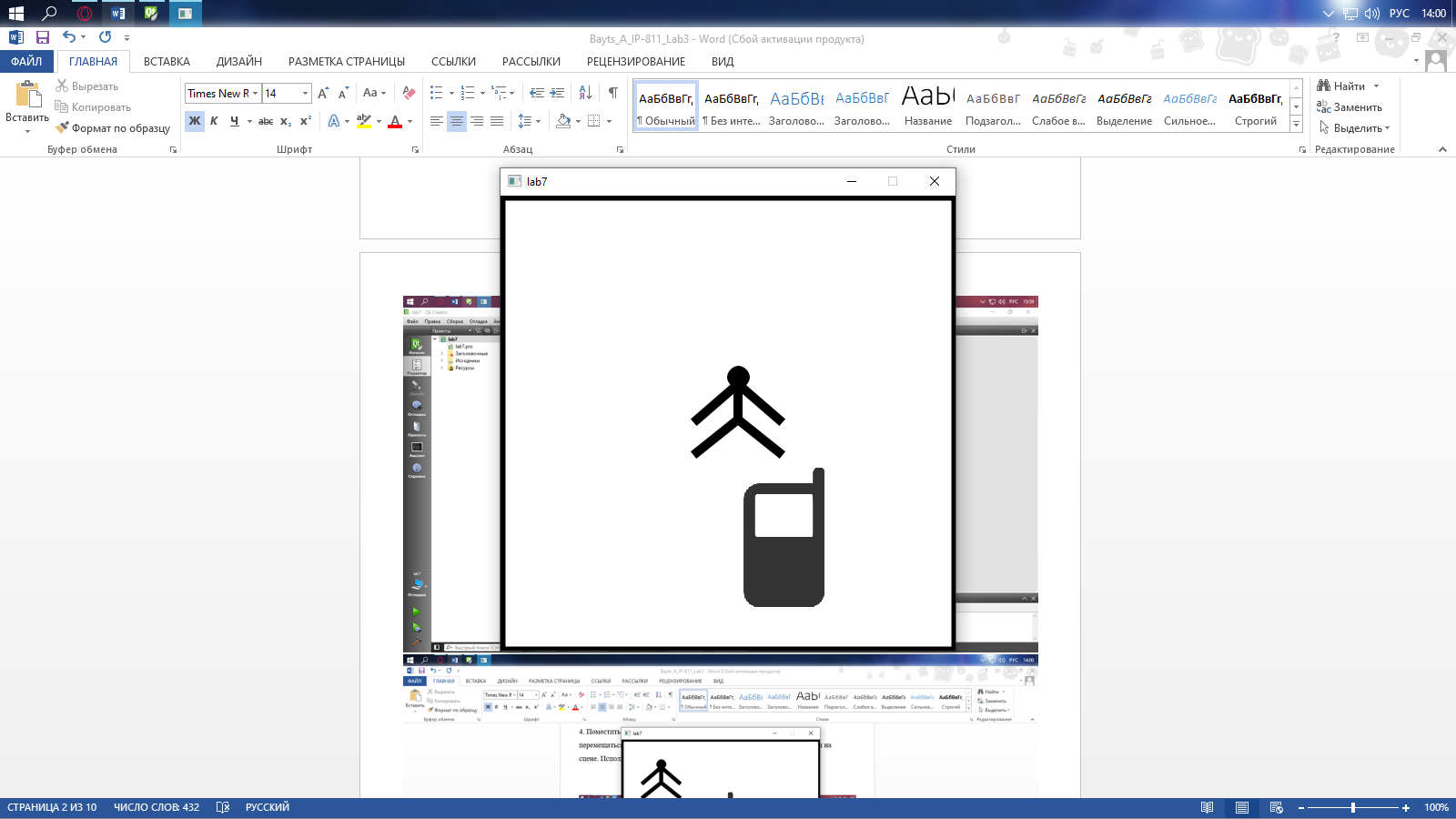
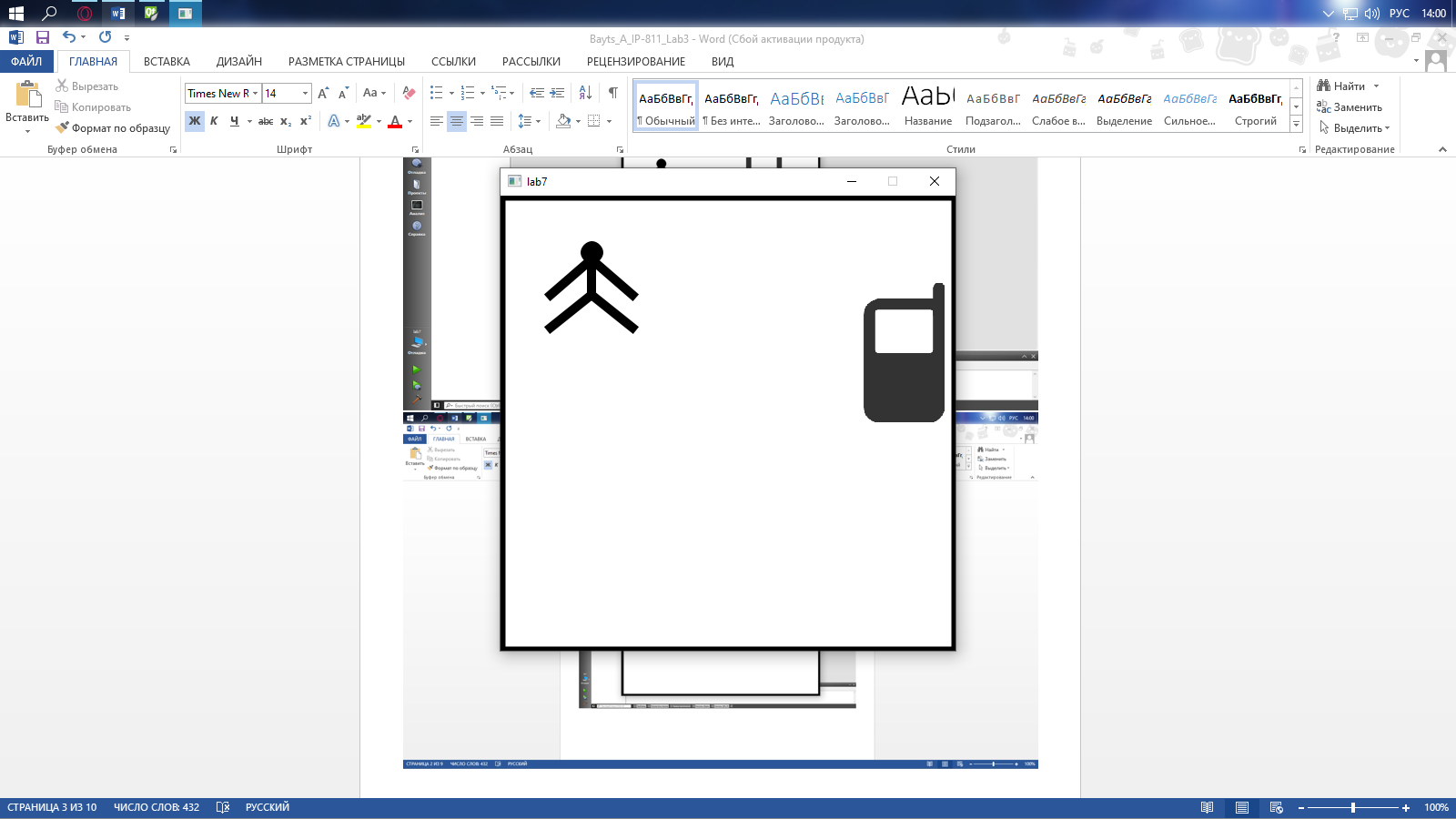
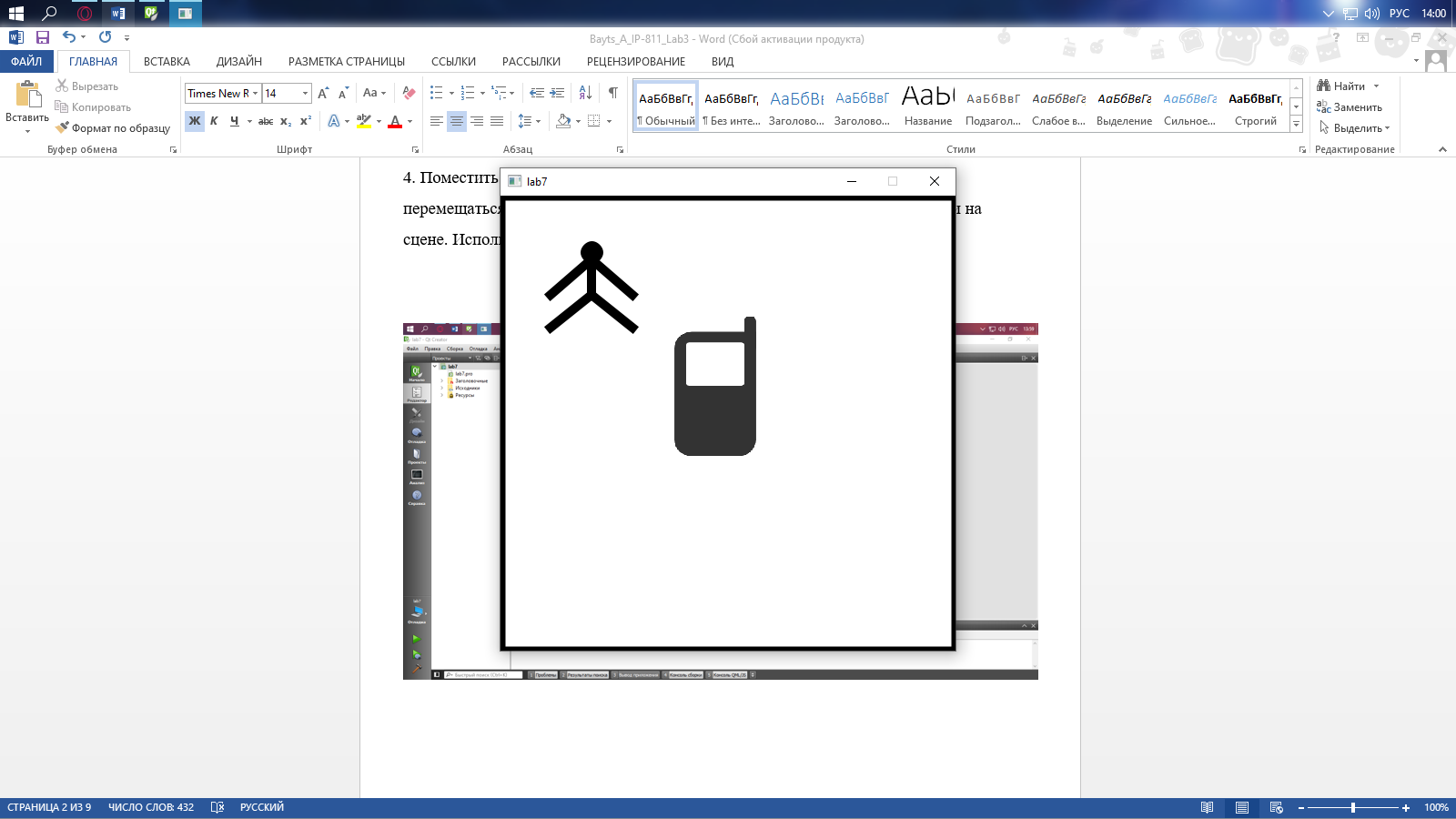
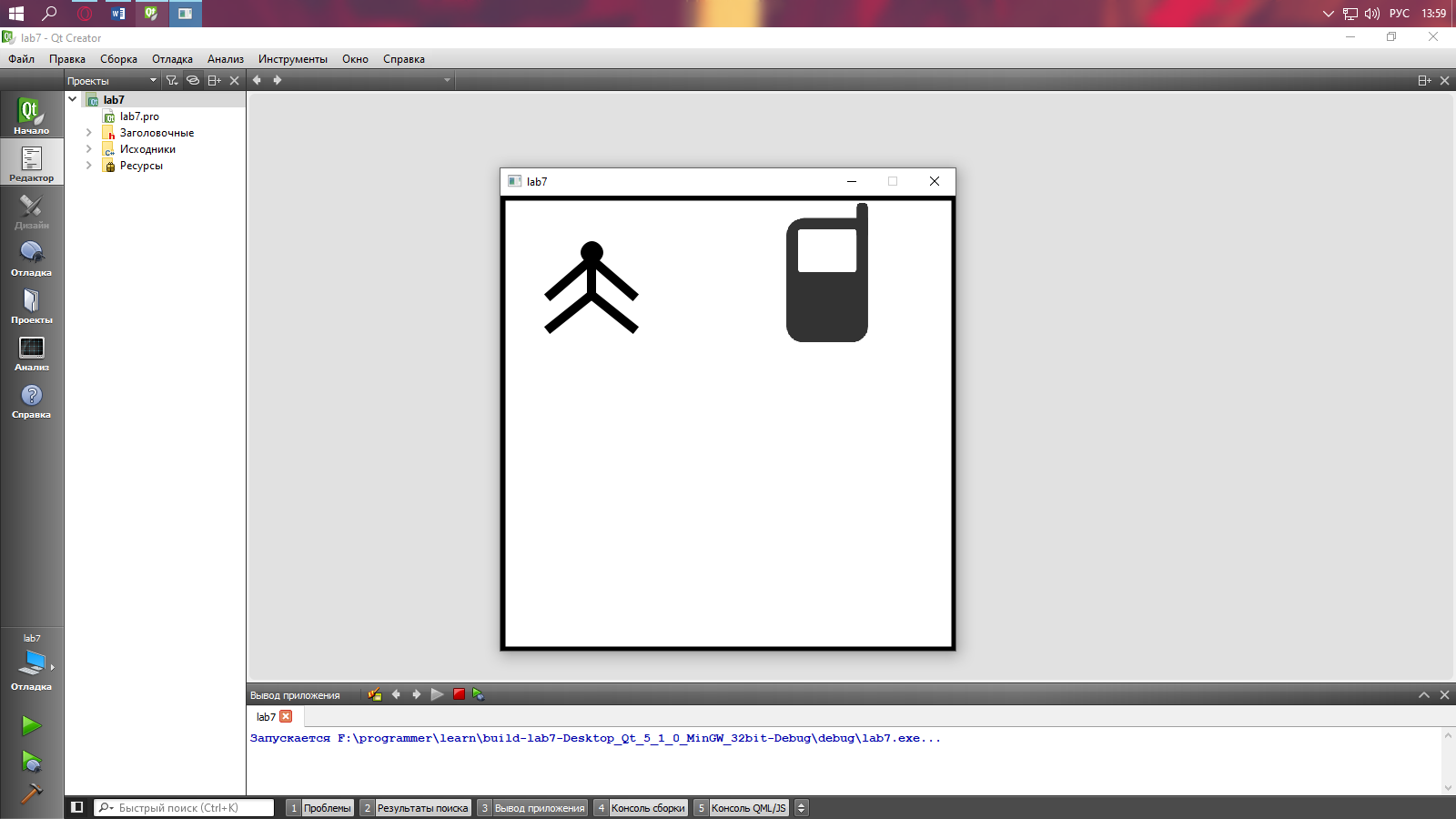


Рисунок 1-4 Разные положения телефона и человека, телефон движется и отталкивается от стен и человека.

Код программы

Main.cpp

#include <QApplication>

#include "viewexample.hpp"

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

viewExample v;

v.show();

return a.exec();

}

Viewexample.hpp

#ifndef VIEWEXAMPLE\_HPP

#define VIEWEXAMPLE\_HPP

#include <QGraphicsView>

#include <QGraphicsScene>

#include "man.hpp"

#include "telephone.hpp"

#include "walls.hpp"

class viewExample : public QGraphicsView

{

public:

viewExample();

private:

void \_init\_view\_elements();

QGraphicsScene nScene;

man \*g;

telephone \*tel;

walls \*wal1;

walls \*wal2;

walls \*wal3;

walls \*wal4;

QTimer mTimer;

private slots:

void speed();

};

#endif // VIEWEXAMPLE\_HPP

Viewexample.cpp

#include "viewexample.hpp"

#include <QDebug>

viewExample::viewExample()

{

setRenderHint(QPainter::Antialiasing);

setCacheMode(QGraphicsView::CacheNone);

setVerticalScrollBarPolicy(Qt::ScrollBarAlwaysOff);

setHorizontalScrollBarPolicy(Qt::ScrollBarAlwaysOff);

setFrameStyle(0);

setSceneRect(0, 0, 500, 500);

setSizePolicy(QSizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed));

setFixedSize(500, 500);

nScene.setItemIndexMethod(QGraphicsScene::NoIndex);

setScene(&nScene);

\_init\_view\_elements();

connect(&mTimer, &QTimer::timeout, this, &viewExample::speed);

mTimer.start(1);

}

void viewExample::\_init\_view\_elements(){

wal1 = new walls();

wal2 = new walls();

wal3 = new walls();

wal4 = new walls();

wal1->setPos(0,0);

wal1->setGeometry(5, 500);

wal2->setPos(0,0);

wal2->setGeometry(500, 5);

wal3->setPos(0,496);

wal3->setGeometry(500, 5);

wal4->setPos(496,0);

wal4->setGeometry(5, 500);

nScene.addItem(wal1);

nScene.addItem(wal2);

nScene.addItem(wal3);

nScene.addItem(wal4);

g = new man();

g->setPos(50, 50);

g->setGeometry(100, 100);

nScene.addItem(g);

tel = new telephone();

tel->setPos(300, 300);

tel->setImage(":/pics/C:/Users/User/Desktop/w256h2561390849252cellphone256.png");

nScene.addItem(tel);

}

void viewExample::speed(){

if(tel->mMove == false){

tel->setPos(tel->pos().x()+0.1\*tel->x, tel->pos().y()+0.1\*tel->y);

if(tel->*collidesWithItem*(wal2) || tel->*collidesWithItem*(wal3))

{

tel->y \*= -1;

}

if(tel->*collidesWithItem*(wal4) || tel->*collidesWithItem*(wal1))

{

tel->x \*= -1;

}

if(tel->*collidesWithItem*(g)){

if((tel->pos().y()+tel->mHeight - g->pos().y())<=1 && (tel->pos().x() +tel->mWidth - g->pos().x()) > 1){

tel->y \*= -1;

qWarning() <<"1";

}

if((tel->pos().x()+tel->mWidth - g->pos().x())<=1 && (tel->pos().y()+tel->mHeight - g->pos().y()) > 1){

tel->x \*= -1;

qWarning() <<"4";

}

if(g->pos().x()+g->mWidth - tel->pos().x() <= 1 && tel->pos().y() + tel->mHeight - g->pos().y() > 1){

tel->x \*= -1;

}

if(g->pos().y()+g->mHeight - tel->pos().y() <=1 && tel->pos().x() + tel->mWidth - g->pos().x() > 1){

tel->y \*= -1;

}

}

}

}

Walls.hpp

#ifndef WALLS\_HPP

#define WALLS\_HPP

#include <QGraphicsItem>

class walls : public QGraphicsItem

{

public:

walls();

QRectF *boundingRect*() const override;

void *paint*(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) override;

void setGeometry(int, int);

private:

int mWidth{0};

int mHeight{0};

};

#endif // WALLS\_HPP

Walls.cpp

#include "walls.hpp"

#include <QPainter>

walls::walls()

{

}

QRectF walls::*boundingRect*() const{

return QRectF(0,0, mWidth, mHeight);

}

void walls::*paint*(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) {

painter->setBrush(Qt::black);

painter->drawRect(0, 0, mWidth, mHeight);

}

void walls::setGeometry(int aWidth, int aHeigth){

mWidth = aWidth;

mHeight = aHeigth;

}

Man.hpp

#ifndef MAN\_HPP

#define MAN\_HPP

#include <QGraphicsItem>

#include <QPointF>

class man : public QGraphicsItem

{

public:

man();

QRectF *boundingRect*() const override;

void *paint*(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) override;

void setGeometry(int, int);

void *hoverEnterEvent*(QGraphicsSceneHoverEvent \*event) override;

void *hoverLeaveEvent*(QGraphicsSceneHoverEvent \*event) override;

void *mousePressEvent*(QGraphicsSceneMouseEvent \*event) override;

void *mouseMoveEvent*(QGraphicsSceneMouseEvent \*event) override;

void *mouseReleaseEvent*(QGraphicsSceneMouseEvent \*event) override;

int mWidth{0};

int mHeight{0};

private:

bool mHover{false};

QPointF mStartMovePos;

};

#endif // MAN\_HPP

Man.cpp

#include "man.hpp"

#include <QPainter>

#include <QDebug>

#include <QPointF>

#include <QGraphicsSceneMouseEvent>

#include <QApplication>

man::man()

{

setAcceptHoverEvents(true);

setAcceptedMouseButtons(Qt::LeftButton);

}

QRectF man::*boundingRect*() const{

return QRectF(0,0, mWidth, mHeight);

}

void man::*paint*(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) {

QPen pen(Qt::black, 10, Qt::SolidLine);

painter->setPen(pen);

painter->drawLine(5, mHeight-5, mWidth/2, mHeight\*0.6);

painter->drawLine(mWidth-5, mHeight-5, mWidth/2, mHeight\*0.6);

painter->drawLine(mWidth/2, mHeight\*0.6, mWidth/2, mHeight\*0.2);

painter->drawLine(mWidth/2, mHeight\*0.2, 5, mHeight\*0.6);

painter->drawLine(mWidth/2, mHeight\*0.2, mWidth-5, mHeight\*0.6);

painter->setBrush(Qt::black);

painter->drawEllipse((mWidth/2)-mHeight\*0.1+4, 5, mHeight\*0.2-5, mHeight\*0.2-5);

}

void man::setGeometry(int aWidth, int aHeigth){

mWidth = aWidth;

mHeight = aHeigth;

}

void man::*hoverEnterEvent*(QGraphicsSceneHoverEvent \*event){

qWarning() << "Enter";

mHover = true;

}

void man::*hoverLeaveEvent*(QGraphicsSceneHoverEvent \*event){

qWarning() << "Leave";

mHover = false;

}

void man::*mousePressEvent*(QGraphicsSceneMouseEvent \*event){

mStartMovePos = event->pos();

}

void man::*mouseMoveEvent*(QGraphicsSceneMouseEvent \*event){

int distance = (event->pos() - mStartMovePos).manhattanLength();

if(distance > QApplication::startDragDistance()){

QPointF np = mapToScene(event->pos() - mStartMovePos);

qWarning() <<np;

if( (np.x() < 6) || (np.y() < 6) ){

qWarning() << "b";

return;

}

if( (np.x()+mWidth > 495) || (np.y()+mHeight > 495) ){

qWarning() <<"v";

return;

}

this->setPos(np);

QGraphicsItem::update();

}

}

void man::*mouseReleaseEvent*(QGraphicsSceneMouseEvent \*event){

}

Telephone.hpp

#ifndef TELEPHONE\_HPP

#define TELEPHONE\_HPP

#include <QGraphicsItem>

#include <QPointF>

#include <QPixmap>

#include <QTimer>

#include <QObject>

class telephone : public QObject, public QGraphicsItem

{

//Q\_OBJECT

public:

telephone();

QRectF *boundingRect*() const override;

void *paint*(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) override;

void *hoverEnterEvent*(QGraphicsSceneHoverEvent \*event) override;

void *hoverLeaveEvent*(QGraphicsSceneHoverEvent \*event) override;

void *mousePressEvent*(QGraphicsSceneMouseEvent \*event) override;

void *mouseMoveEvent*(QGraphicsSceneMouseEvent \*event) override;

void *mouseReleaseEvent*(QGraphicsSceneMouseEvent \*event) override;

void setImage(QString);

int x{1};

int y{1};

bool mMove{false};

int mWidth{0};

int mHeight{0};

private:

QPointF mStartMovePos;

QPixmap mPixMap;

};

#endif // TELEPHONE\_HPP

Telephone.cpp

#include "telephone.hpp"

#include <QPainter>

#include <QDebug>

#include <QPointF>

#include <QGraphicsSceneMouseEvent>

#include <QApplication>

#include <QGraphicsScene>

telephone::telephone()

{

setAcceptHoverEvents(true);

setAcceptedMouseButtons(Qt::LeftButton);

}

QRectF telephone::*boundingRect*() const{

return QRectF(0,0, mWidth, mHeight);

}

void telephone::*paint*(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) {

painter->drawPixmap(0,0, mWidth, mHeight, mPixMap);

}

void telephone::*hoverEnterEvent*(QGraphicsSceneHoverEvent \*event){

qWarning() << "Enter";

}

void telephone::*hoverLeaveEvent*(QGraphicsSceneHoverEvent \*event){

qWarning() << "Leave";

}

void telephone::*mousePressEvent*(QGraphicsSceneMouseEvent \*event){

mStartMovePos = event->pos();

}

void telephone::*mouseMoveEvent*(QGraphicsSceneMouseEvent \*event){

int distance = (event->pos() - mStartMovePos).manhattanLength();

if(distance > QApplication::startDragDistance()){

mMove = true;

QPointF np = mapToScene(event->pos() - mStartMovePos);

qWarning() <<np;

if( (np.x() < 0) || (np.y() < 0) ){

qWarning() << "b";

return;

}

if( (np.x()+mWidth > 500) || (np.y()+mHeight > 500) ){

qWarning() <<"v";

return;

}

this->setPos(np);

}

}

void telephone::*mouseReleaseEvent*(QGraphicsSceneMouseEvent \*event){

mMove = false;

}

void telephone::setImage(QString aPath)

{

mPixMap.load(aPath);

mWidth = mPixMap.width()-50;

mHeight = mPixMap.height()-100;

}