Федеральное государственное бюджетное образовательное учреждение «Сибирский государственный университет телекоммуникаций и информатики»

кафедра прикладной математики и кибернетики

Лабораторная работа №7 по дисциплине «Визуальное программирование и человеко-машинное взаимодействие»

По теме «Графика»

**Выполнил(а):** студентка 3 курса

Колосова Яна Владимировна

**Проверил(а)**: Преподаватель кафедры ПМиК

Мерзлякова Екатерина Юрьевна

Новосибирск 2020г.

**Цель работы:**

Научиться работать с графикой в Qt.

**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.

По варианту необходимо:

Машина и движущийся мяч

**Листинг программы:**

**ball\_kolosova.h:**

#ifndef BALL\_H

#define BALL\_H

#include <QObject>

#include <QGraphicsItem>

#include <QPainter>

#include <QGraphicsSceneMouseEvent>

#include <QDebug>

#include <QCursor>

class ball : public QObject, public QGraphicsItem

{

Q\_OBJECT

public:

explicit ball(QObject \*parent = 0);

~***ball***();

signals:

public slots:

void **slotGameTimer**();

protected:

QRectF ***boundingRect***() const;

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

private:

bool LEFT;

bool RIGHT;

bool UP;

bool DOWN;

QRect rect;

QTimer \*timer;

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

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

};

#endif // BALL\_H

**ball\_kolosova.cpp:**

#include "ball\_kolosova.h"

ball::**ball**(QObject \*parent) :

QObject(parent), QGraphicsItem()

{

UP = true;

DOWN = false;

LEFT = true;

RIGHT = false;

}

ball::~***ball***()

{

}

QRectF ball::***boundingRect***() const

{

return QRectF(-30,-30,30,30);

}

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

{

painter->setBrush(Qt::red);

painter->drawEllipse(-30, -30, 30, 30);

Q\_UNUSED(option);

Q\_UNUSED(widget);

}

void ball::***mouseMoveEvent***(QGraphicsSceneMouseEvent \*event)

{

this->setPos(mapToScene(event->pos()));

}

void ball::***mousePressEvent***(QGraphicsSceneMouseEvent \*event)

{

this->setCursor(QCursor(Qt::ClosedHandCursor));

Q\_UNUSED(event);

}

void ball::***mouseReleaseEvent***(QGraphicsSceneMouseEvent \*event)

{

this->setCursor(QCursor(Qt::ArrowCursor));

Q\_UNUSED(event);

}

void ball::**slotGameTimer**()

{

if(LEFT == true){

setPos(mapToParent(-5, 0));

}

if(RIGHT == true){

setPos(mapToParent(5, 0));

}

if(UP == true){

setPos(mapToParent(0, -5));

}

if(DOWN == true){

setPos(mapToParent(0, 5

}

if(this->x() - 10 < -250){

this->setX(-240);// слева

LEFT = false;

RIGHT = true;

}

if(this->x() + 10 > 250){

this->setX(240); // справа

RIGHT = false;

LEFT = true;

}

if(this->y() - 10 < -250){

this->setY(-240); // сверху

UP = false;

DOWN = true;

}

if(this->y() + 10 > 250){

this->setY(240); // снизу

UP = true;

DOWN = false;

}

}

**car\_kolosova.h:**

#ifndef car\_kolosova\_H

#define car\_kolosova\_H

#include <QObject>

#include <QGraphicsItem>

#include <QPainter>

#include <QGraphicsSceneMouseEvent>

#include <QDebug>

#include <QCursor>

#include <QPixmap>

#include <windows.h>

class car\_kolosova : public QObject, public QGraphicsItem

{

Q\_OBJECT

public:

explicit car\_kolosova(QObject \*parent = 0);

~***car\_kolosova***();

protected:

QRectF ***boundingRect***() const;

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

private:

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

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

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

};

#endif // car\_kolosova\_H

**car\_kolosova.cpp:**

#include "car\_kolosova.h"

car\_kolosova::**car\_kolosova**(QObject \*parent) :

QObject(parent), QGraphicsItem()

{

}

car\_kolosova::~***car\_kolosova***()

{

}

QRectF car\_kolosova::***boundingRect***() const

{

QImage source;

source.load("C:\\Users\\yava06\\Desktop\\lab7\_kolosova\_v2\\car.png");

QRectF tmp(source.rect());

return tmp; /// Ограничиваем область, в которой лежит наша машинка

}

void car\_kolosova::***paint***(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

{

QImage source;

source.load("C:\\Users\\yava06\\Desktop\\lab7\_kolosova\_v2\\car.png

QRect rect(source.rect());

painter->drawImage(rect, source);

Q\_UNUSED(option);

Q\_UNUSED(widget);

}

void car\_kolosova::***mouseMoveEvent***(QGraphicsSceneMouseEvent \*event)

{

this->setPos(mapToScene(event->pos()));

}

void car\_kolosova::***mousePressEvent***(QGraphicsSceneMouseEvent \*event)

{

this->setCursor(QCursor(Qt::ClosedHandCursor));

Q\_UNUSED(event);

}

void car\_kolosova::***mouseReleaseEvent***(QGraphicsSceneMouseEvent \*event)

{

this->setCursor(QCursor(Qt::ArrowCursor));

Q\_UNUSED(event);

}

**widget\_kolosova.h:**

#ifndef WIDGET\_H

#define WIDGET\_H

#include <QWidget>

#include "ball\_kolosova.h"

#include "car\_kolosova.h"

#include <QGraphicsScene>

#include <QShortcut>

#include <QTimer>

QT\_BEGIN\_NAMESPACE

namespace Ui { class widget; }

QT\_END\_NAMESPACE

class widget : public QWidget

{

Q\_OBJECT

public:

explicit widget(QWidget \*parent = 0);

~***widget***();

private:

Ui::widget \*ui;

QGraphicsScene \*scene;

ball \*Ball;

car\_kolosova \*Car;

QTimer \*timer;

};

#endif // WIDGET\_H

**widget\_kolosova.cpp:**

#include "widget\_kolosova.h"

#include "ui\_widget.h"

widget::**widget**(QWidget \*parent)

: QWidget(parent)

, ui(new Ui::widget)

{

ui->setupUi(this);

this->resize(800,600);

this->setFixedSize(800,600);

scene = new QGraphicsScene();

Ball = new ball();

Car = new car\_kolosova();

ui->graphicsView->setScene(scene);

ui->graphicsView->setRenderHint(QPainter::Antialiasing);

ui->graphicsView->setVerticalScrollBarPolicy(Qt::ScrollBarAlwaysOff);

ui->graphicsView->setHorizontalScrollBarPolicy(Qt::ScrollBarAlwaysOff);

scene->setSceneRect(-250,-250,500,500);

scene->addLine(-250,-250, 250,-250, QPen(Qt::black));

scene->addLine(-250, 250, 250, 250, QPen(Qt::black));

scene->addLine(-250,-250,-250, 250, QPen(Qt::black));

scene->addLine( 250,-250, 250, 250, QPen(Qt::black));

scene->addItem(Ball);

scene->addItem(Car);

Ball->setPos(200,0);

Car->setPos(0, 0);

timer = new QTimer();

connect(timer, &QTimer::timeout, Ball, &ball::slotGameTimer);

timer->start(1000 / 50);

}

widget::~***widget***()

{

delete ui;

}

**main.cpp:**

#include "widget\_kolosova.h"

#include <QApplication>

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

{

QApplication a(argc, argv);

widget w;

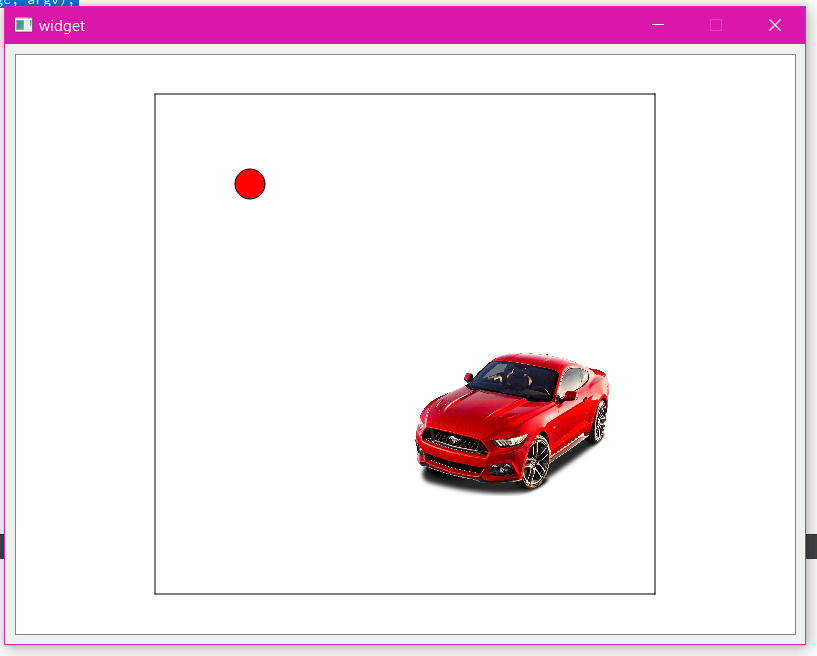
w.show();

return a.exec();

}

**Результат работы программы:**

1. Начало работы  
   На скриншоте виден мяч, картинка машины и стены, от которых отталкивается мяч



1. Передвигаем машину, мяч движется сам только в области, ограниченной стенами

