Санкт-Петербургский государственный политехнический университет

Факультет технической кибернетики

Кафедра компьютерных систем и программных технологий

Курсовая работа по курсу «Программирование»

Тема работы: «Разработка приложения с графическим интерфейсом на языке C++ с использованием библиотеки QT»

Работу выполнил:

студент группы 1081/3

Арсёнов Юрий

Преподаватель:

Доц. Пышкин Е. В.

Санкт - Петербург 2012

## Образ и границы проекта

Проект посвящен разработке программного обеспечения для игры в «черточки». Суть игры состоит в следующем:

А) На поле чертится квадрат из клеток размером N\*N клеток (в рассматриваемом далее примере N=5). Принципиально игра может быть вестись на клеточном поле любой формы, но в данном проекте мы ограничимся квадратным полем (см. рис.1.1).

Рис. 1.1. Начальное состояние игрового поля

Б) В игре участвуют два игрока. Они по по очереди ставит на поле черточки по стороне одной клетки. На рис. 1.2. приведен фрагмент процесса игры, где синие черточки соответствуют ходам одного игрока, а красные – другого.

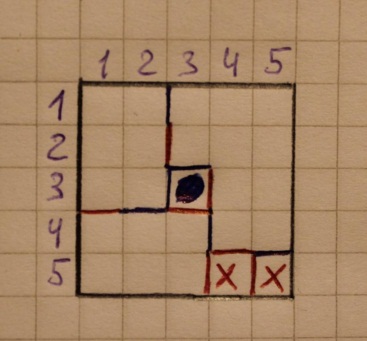
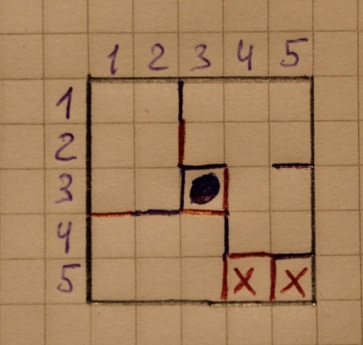
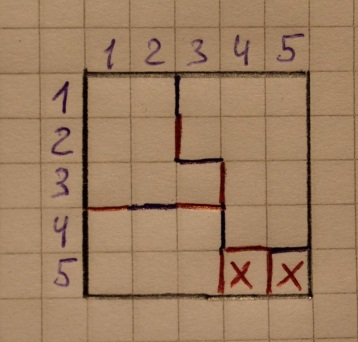
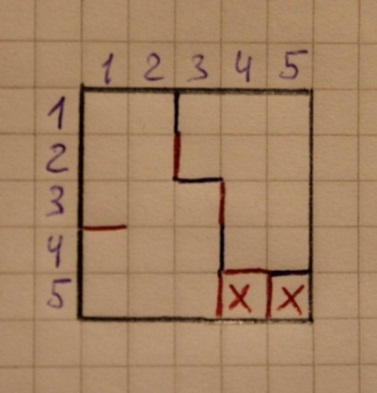
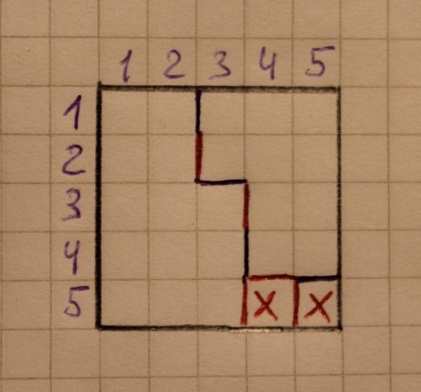
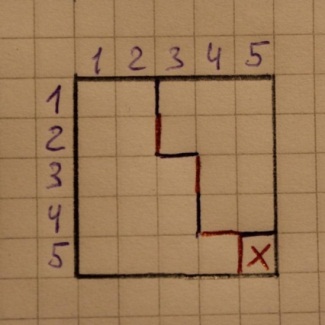
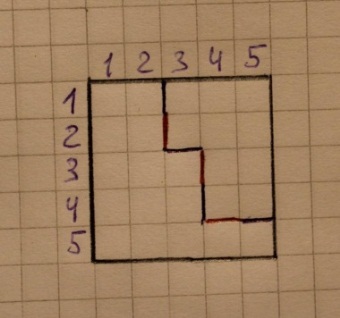
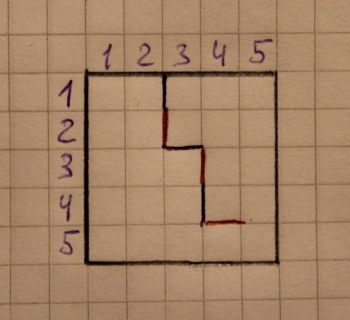
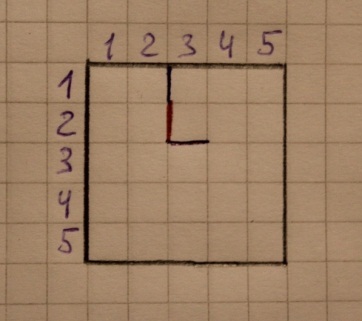
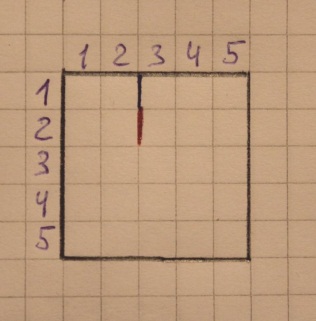
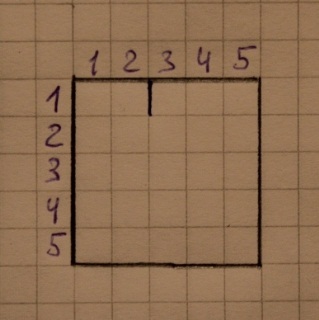
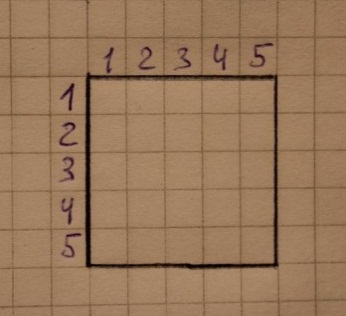
Рис. 1.2. Процесс игры

В) Черточки можно ставить в любом порядке, но когда кто-то из игроков поставит сою черточку так:, что ход последующего игрока может захватить целую клетку, тоэтот игрок ходит и ставаит в захваченную клетку крестик. После этого следующий ход принадлежит этому же игроку, у него есть шанс ещё раз захватить клетку, уже расположенную слева от него, что он и делает. Больше игроку нечего захватывать, но ход принадлежит ему, следовательно он ставит свою черточку в любое свободное место:.

Далее повторяестя то же самое, но в этом случае повезло синему игроку, он и захватывает свободное поле:

Д)Игра продолжается до тех пор, пока не будут заняты все свободные клетки, после этого ведется подсчет количества захваченных клеток. Победил тот, у кого клеток больше.

## Примерная схема ходов игроков



## Техническое задание

Разработать реализацию игры, представленной в пункте 1, с использованием библиотеки Qt на языке C++.

### Основные требования

Главное меню приложения:

1. Начать новую игру.
2. Узнать порядок хода
3. Узнать победителя

### Основные сценарии работы приложения

Команды меню связаны с реализацией следующих сценариев работы:

### 1)Начало игры

А)Предложение создать новых игроков, если не найдется подходящего игрока или захочется создать других.

Б)Когда будут выбраны 2 игрока начнется игра, где каждый из пользователей по очереди будет ходить согласно установленным правилам в игре.

### 2)Определение победителя

### 3)Выход из игры.

### Требования к графическому интерфейсу

1) Возможность задания имен игроков с помощбю диалоговых окон.

2)Экран игры с разлинованным игровым полем с возможностью с помощью мышки ставить черточки и крестики с ноликами по очереди каждому игроку.

3)Виджет, отвечающий за очередность хода.

4)Виджет отвечающий за определение победителя.

### Основной сценарий функционирования программы

1)Запуск программы с возможностью выбора имен 2-х игроков

2)Появление игрового поля

3)Начало самой игры в возможностью поочередно кликать на линии и на внутренние клетки

4)Определение победителя и выход из игры

### 3. Системная архитектура

### Классы графического интерфейса

Класс, отвечающий за линии

class MyLine : public QObject, public QGraphicsLineItem

{

Q\_OBJECT

public:

MyLine(int x1, int x2, int x3, int x4);

bool isChecked;

signals:

void mySignal();

protected:

void *mousePressEvent*(QGraphicsSceneMouseEvent\* event);

};

Класс, отвечающий за текст в клетках

class MyText : public QGraphicsTextItem

{

Q\_OBJECT

public:

MyText(const QString q);

protected:

void *mousePressEvent*(QGraphicsSceneMouseEvent\* event);

void *mouseDoubleClickEvent*( QGraphicsSceneMouseEvent \* event );

void *keyPressEvent*(QKeyEvent \*event);

// void mouseReleaseEvent ( QMouseEvent \* event );

signals:

void textSignal();

};

Класс, отвечающий за определение победителя

class WinText : public QGraphicsTextItem

{

Q\_OBJECT

public:

WinText(const QString q);

signals:

void winSignal();

protected:

void *mousePressEvent*(QGraphicsSceneMouseEvent\* event);

};

Класс, отвечающий за игровое поле в общем

class GameFieldA : public QObject

{

Q\_OBJECT

public:

// QPen PenBlack;

QPen PenGray;

QGraphicsScene\* GameScene;

std::vector<std::vector<MyLine\*> > VerticalLine;

std::vector<std::vector<MyLine\*> > HorizontalLine;

WinText\* finished;

TurnText \*tu;

QGraphicsTextItem\* hello;

std::vector<std::vector<MyText\*> > CellText;

// MyText\* CellText[10][10];

GameFieldA(int height, int width, QString str1, QString str2);

QString string1;

QString string2;

int h;

int w;

~*GameFieldA*();

void InitializationField();

int CheckField();

int FieldCounting();

void ShowField();

protected:

void keyPressEvent(QKeyEvent \*event);

public slots:

void mySlot()

{

if(tu->toPlainText() == string1)

{

delete tu;

tu=new TurnText(string2);

tu->setPos(420, 120);

tu->setDefaultTextColor(Qt::blue);

GameScene->addItem(tu);

}

else

{

delete tu;

tu=new TurnText(string1);

tu->setPos(420, 120);

tu->setDefaultTextColor(Qt::red);

GameScene->addItem(tu);

qDebug()<<"Text1111";

}

}

void winSlot()

{

delete finished;

int red = 0;

int blue = 0;

for(int i = 0; i < w; i++)

{

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

{

if(CellText[i][j]->toPlainText() == "x")

red++;

// else

// blue++;

if(CellText[i][j]->toPlainText() == "o")

blue++;

}

}

if(red > blue)

{

finished = new WinText(string1);

finished->setPos(380, 150);

finished->setDefaultTextColor(Qt::red);

GameScene->addItem(finished);

qDebug()<<"Hello";

}

if(red < blue)

{

finished = new WinText(string2);

finished->setPos(380, 150);

finished->setDefaultTextColor(Qt::blue);

GameScene->addItem(finished);

qDebug()<<"Hello";

}

if(red == blue)

{

finished = new WinText("dead heat");

finished->setPos(380, 150);

finished->setDefaultTextColor(Qt::blue);

GameScene->addItem(finished);

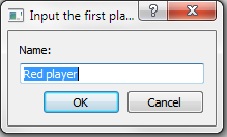
}

}

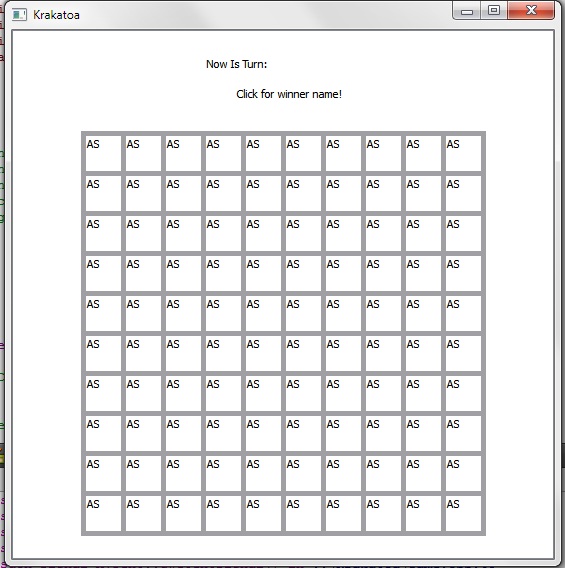
};

### Тестирование приложения

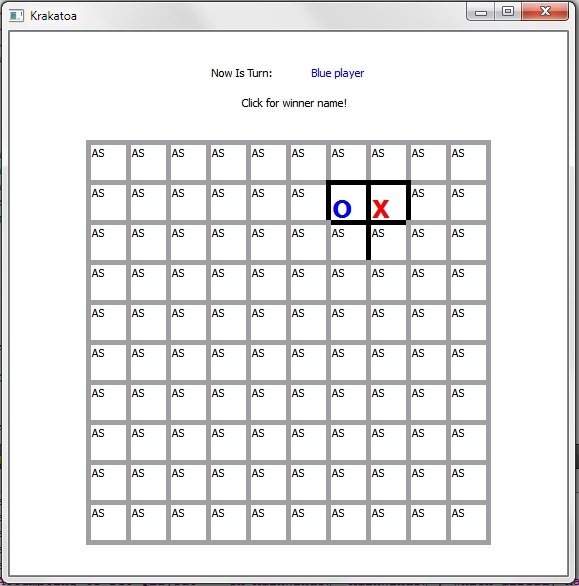
1. Запускаем приложение



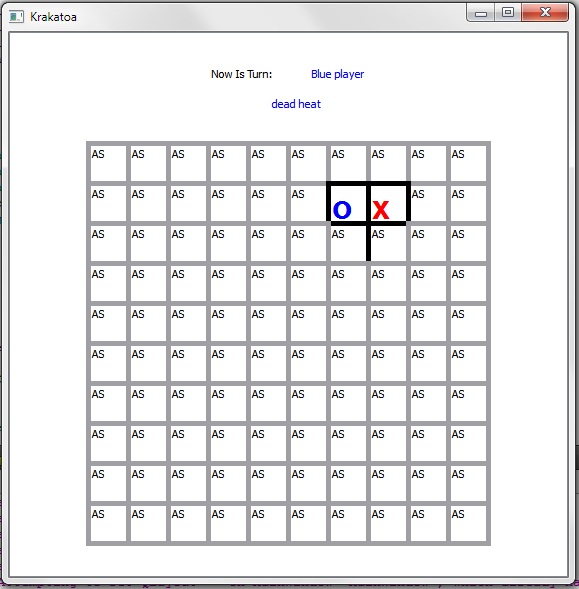
1. Вводим имена игроков и выводится игровое поле



1. Ставим линии, и крестики с ноликами, при этом отображается очередность хода



4)Определние победителя и конец игры



Видим, что процесс теста прошел достаточно удачно. Однако, клики на линии регистрируются неточно, это связано с определением классов линий в библиотеках Qt.

В целом, считаю, что задача о создании игры «Krakatoa» мною выполнена.