**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)»**

**(СПбГЭТУ)**

**Факультет** компьютерных технологий и информатики

**Кафедра** АСОИУ

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ**

по дисциплине «Объектно-ориентированное программирование»

Тема работы: «Приложение-ассистент для настольной игры CLUEDO»

Выполнила Катенко Ю.В.

Группа №2363

Преподаватель: Егоров С.С.

Санкт-Петербург

2015 г.

# ВВЕДЕНИЕ

## 1 Правила игры

*«Сегодня вечером Сэмюэл Блэк был найден убитым в своем особняке! Детективы назвали шесть подозреваемых. В особняке были найдены шесть орудий убийства. Но детективы не смогли раскрыть дело, и найти убийцу предстоит вам!»*

CLUEDO — классическая детективная игра. Ее цель — с помощью дедукции выяснить ответы на три вопроса: кто, где и чем убил Блэка. Используются шесть видов орудий убийства, по одному на каждого, и, соответственно, в игре шесть подозреваемых. Само поле настольной игры состоит из девяти комнат, в четырёх из которых есть секретный переход, и игровых клеток для осуществления ходов. Количество шагов определяется числом выпавших очков на игральных кубиках.

Подозреваемые: Мистер Плам, Леди Уайт, Леди Скарлет, Мистер Грин, Мистер Мастард, Леди Пикок.

Орудия убийства: Веревка, Кинжал, Гаечный ключ, Револьвер, Подсвечник, Свинцовая труба.

Комнаты особняка: Внутренний двор, Бильярдная, Кабинет, Столовая, Гараж, Гостиная, Кухня, Спальня, Ванная комната.

Каждому подозреваемому, орудию убийства и комнате соответствует карта с уликой. Три карты кладутся в конверт, остальные распределяются между игроками. Если карты невозможно распределить поровну, «лишние» кладутся возле игрового поля.

Каждый игрок получает пустой список Улик, где должен отметить полученные карты. Затем во время своего хода игрок перемещается в одну из комнат, где высказывает предположение об убийце и орудии. После этого соперники, если у них есть названные улики, должны показать игроку по одной карте. Если названных улик нет, игрок может посмотреть одну из карт, лежащих возле игрового поля. В конце игры неотмеченными должны остаться три карты — они и лежат в конверте.

## 2 Цель и задачи

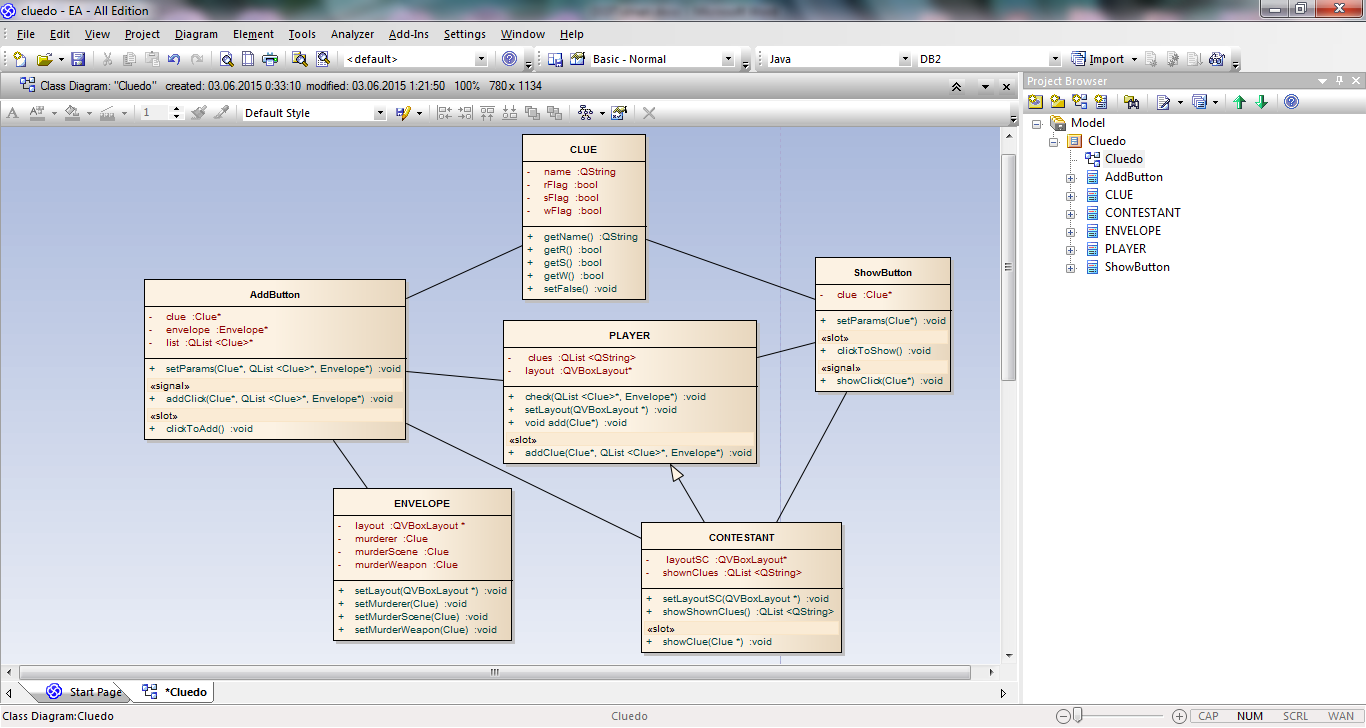
Целью данной работы является корректно-работающее приложение-ассистент для настольной игры CLUEDO. Приложение должно быть построено с учетом следующих условий:

1. Все УЛИКИ отображаются в виде списка.
2. В процессе игры ИГРОК *отмечает* свои улики, улики СОПЕРНИКОВ и улики, не принадлежащие НИКОМУ.
3. Игрок отмечает улики, которые *показал* соперникам.
4. В окне программы отображаются списки улик игрока, соперников, ничьих улик, а также показанных соперникам улик.
5. Когда отмечены все улики определенного типа кроме одной, она автоматически помещается в КОНВЕРТ.
6. Содержимое конверта также отображается в виде списка.

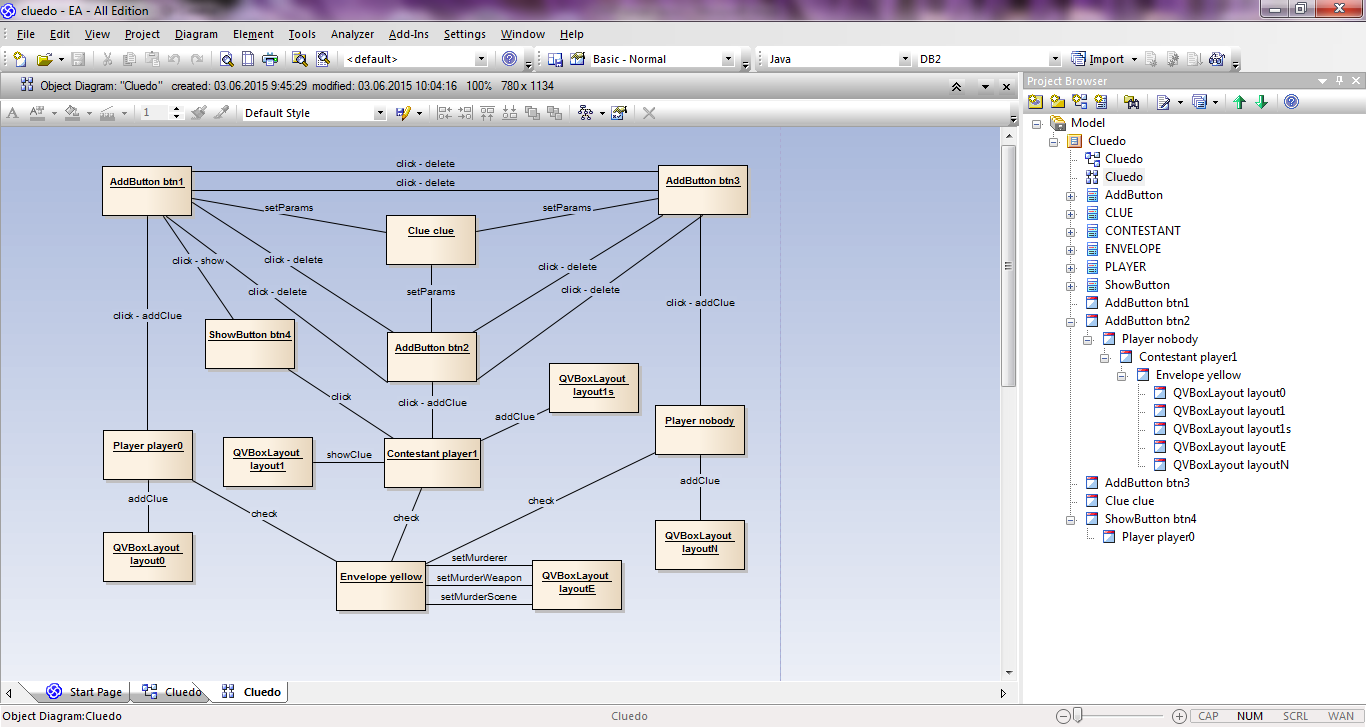
Для достижения цели необходимо решить следующие задачи:

1. Создать классы УЛИКА, ИГРОК, СОПЕРНИК, КОНВЕРТ
2. Создать методы, позволяющие:
   1. Отмечать улики
   2. Показывать улики соперникам
   3. Вычислять, какие улики находятся в конверте
   4. Отображать списки улик каждого из игроков, списки показанных улик, а также содержимое конверта

ДИАГРАММА КЛАССОВ



# ДИАГРАММА ОБЪЕКТОВ



# ЗАГОЛОВОЧНЫЕ ФАЙЛЫ КЛАССОВ

Листинг 1 clue.h

#include <QString>

class Clue

{

QString name;

bool sFlag, wFlag, rFlag; //suspect, weapon, room

public:

Clue();

Clue(char\*, bool, bool, bool);

QString getName() { return name; }

bool getS() { return sFlag; }

bool getW() { return wFlag; }

bool getR() { return rFlag; }

void setFalse (){ sFlag = 0;

wFlag = 0;

rFlag = 0; }

};

Листинг 2 player.h

#include <QList>

#include <QLabel>

#include "clue.h"

#include <QObject>

#include <QVBoxLayout>

#include "envelope.h"

class Player : public QObject

{

Q\_OBJECT

QList <QString> clues;

QVBoxLayout\* layout;

public:

Player();

~*Player*();

void setLayout(QVBoxLayout \*l) { layout = l; }

void add(Clue\*);

void check(QList <Clue>\*, Envelope\*);

public slots:

void addClue(Clue\*, QList <Clue>\*, Envelope\*);

};

Листинг 3 contestant.h

#include "player.h"

class Contestant : public Player

{

Q\_OBJECT

QList <QString> shownClues;

QVBoxLayout\* layoutSC; //shown clues

public:

Contestant();

~*Contestant*();

QList <QString> showShownClues(){ return shownClues; }

void setLayoutSC(QVBoxLayout \*l) { layoutSC = l; }

public slots:

void showClue(Clue \*c);

};

Листинг 4 envelope.h

#include "clue.h"

#include <QString>

#include <QVBoxLayout>

#include <QLabel>

class Envelope

{

Clue murderer;

Clue murderWeapon;

Clue murderScene;

QVBoxLayout \*layout;

public:

Envelope();

void setLayout(QVBoxLayout \*l);

void setMurderer(Clue m);

void setMurderWeapon(Clue w);

void setMurderScene(Clue r);

};

Листинг 5 addbutton.h

#include <QPushButton>

#include "envelope.h"

#include "contestant.h"

class AddButton : public QPushButton

{

Q\_OBJECT

Clue\* clue;

QList <Clue>\* list;

Envelope\* envelope;

public:

AddButton(const QString & text, QWidget \* parent = 0);

void setParams (Clue\*, QList <Clue>\*, Envelope\*);

signals:

void addClick(Clue\*, QList <Clue>\*, Envelope\*);

public slots:

void clickToAdd();

};

Листинг 6 showbutton.h

#include <QPushButton>

#include "contestant.h"

class ShowButton : public QPushButton

{

Q\_OBJECT

Clue\* clue;

public:

ShowButton();

ShowButton(const QString & text, QWidget \* parent = 0);

void setParams(Clue\*);

signals:

void showClick(Clue\*);

public slots:

void clickToShow();

};

# ВЫПОЛНЕНИЕ ПРОГРАММЫ

На рис. 1 программа находится в начальном состоянии, ни одна улика не отмечена.

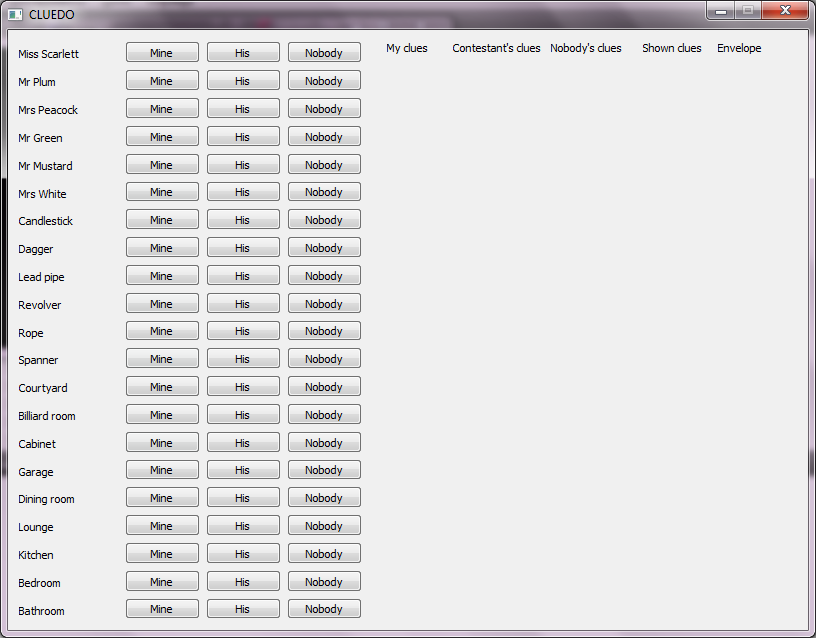


Рис. 1

На рис. 2 отображен момент, когда некоторые улики добавлены в списки, а некоторые — показаны.

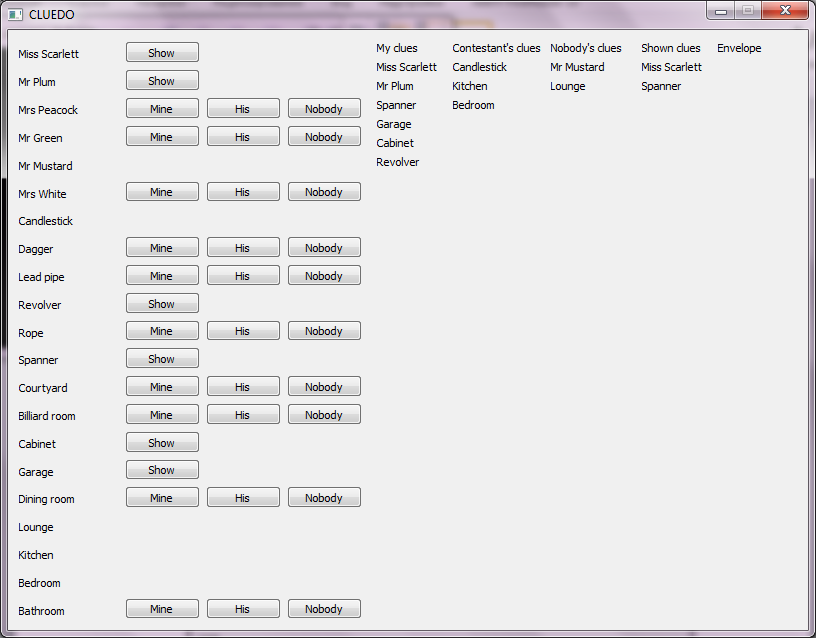


Рис. 2

На рис. 3 отображено состояние, когда определены две улики, лежащие в конверте.

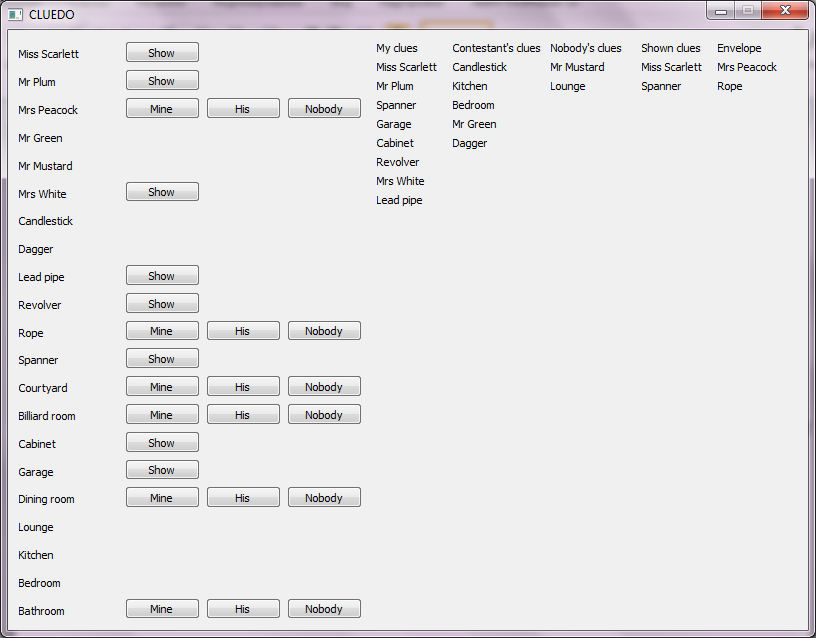


Рис. 3

# ЗАКЛЮЧЕНИЕ

В результате выполнения работы было создано приложение-ассистент для настольной игры CLUEDO. Оно соответствует заявленным требованиям и выполняет свои функции корректно, а значит, поставленная цель достигнута.

Необходимо подчеркнуть, что полученную программу нужно значительно улучшить. Необходимо сделать ее более привлекательной эстетически. Также нужно добавить режимы для 3-6 игроков. Кроме того, можно портировать программу на мобильные платформы, чтобы ее было удобнее использовать во время игры