Ministerul Educației al Republicii Moldova

Universitatea Tehnică a Moldovei
Facultatea CIM

Catedra Automatica și Tehnologii Informaționale

RAPORT

Lucrare de laborator Nr.6
La MIDPS

A efectuat: st. Gr. TI-142
Cuțitaru Adrian

A verificat: lect. asist.
Cojanu Irina

Chișinău 2016

Lucrarea de laborator nr.6

Scopul lucrarii:

Crearea unei aplicatii complexe in echipa

Divizarea sarcinilor pe membrii echipei

Obiective:

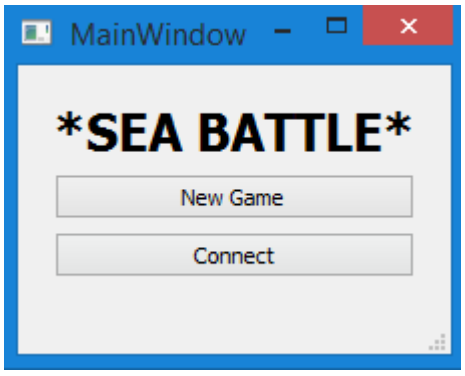
Dezvolatarea unei aplicatii

-Game Development (web,mobile,desktop)

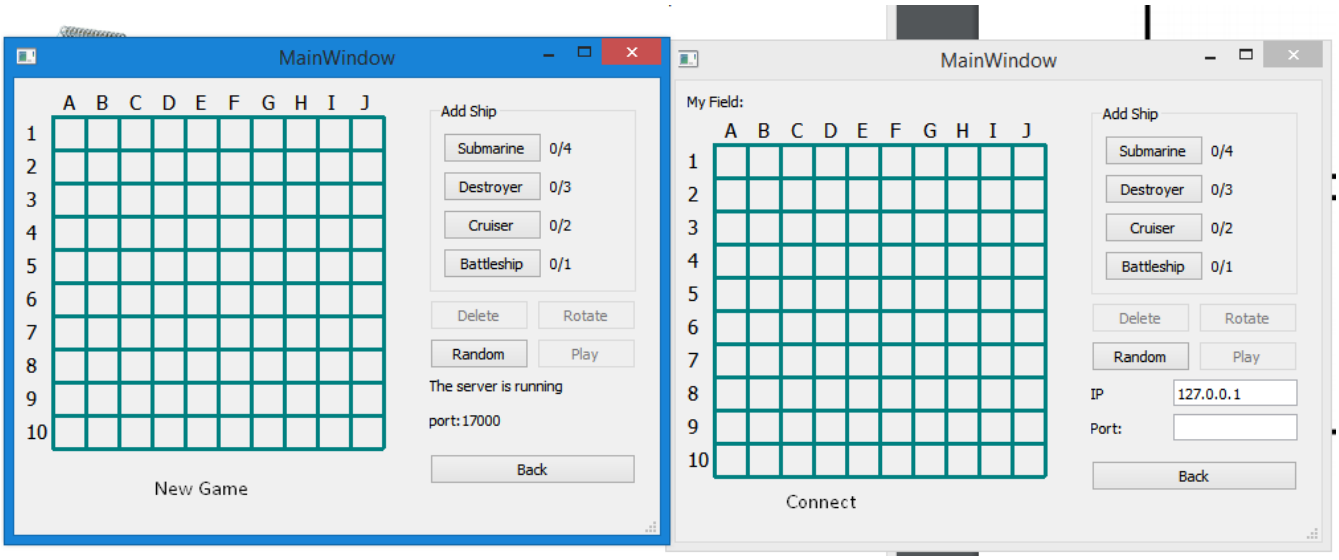
Realizarea lucrarii de laborator:

Jocul contine 3 ferestre:

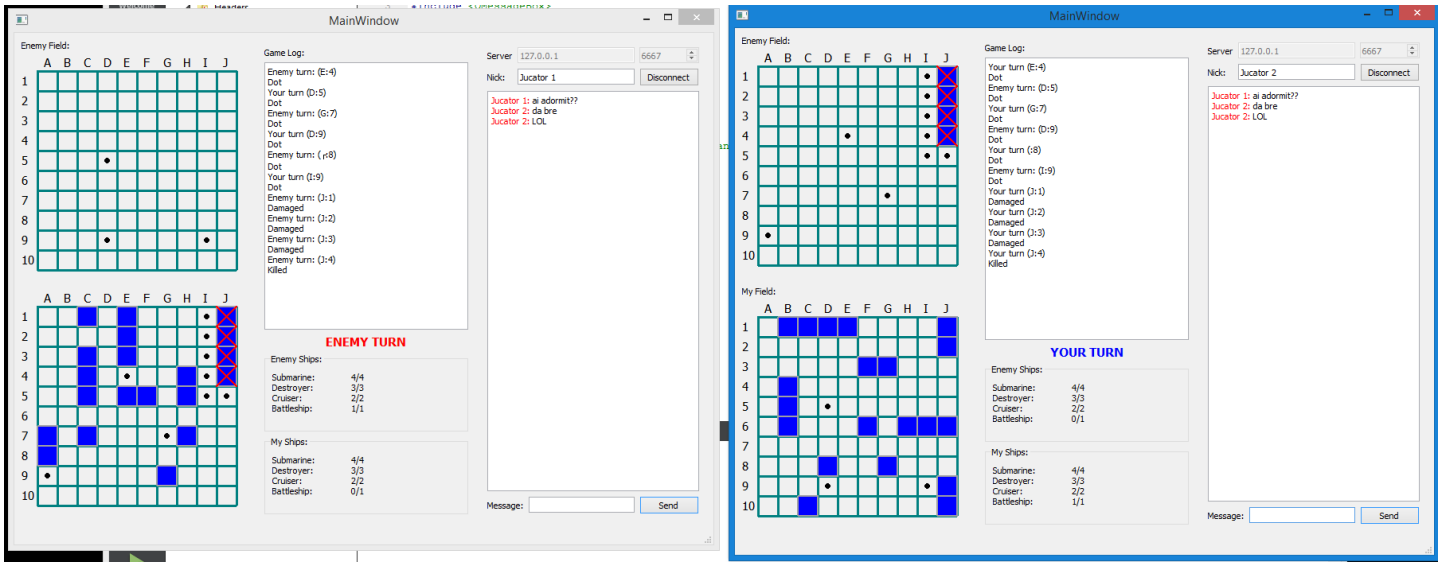
1. Fereastra initiala



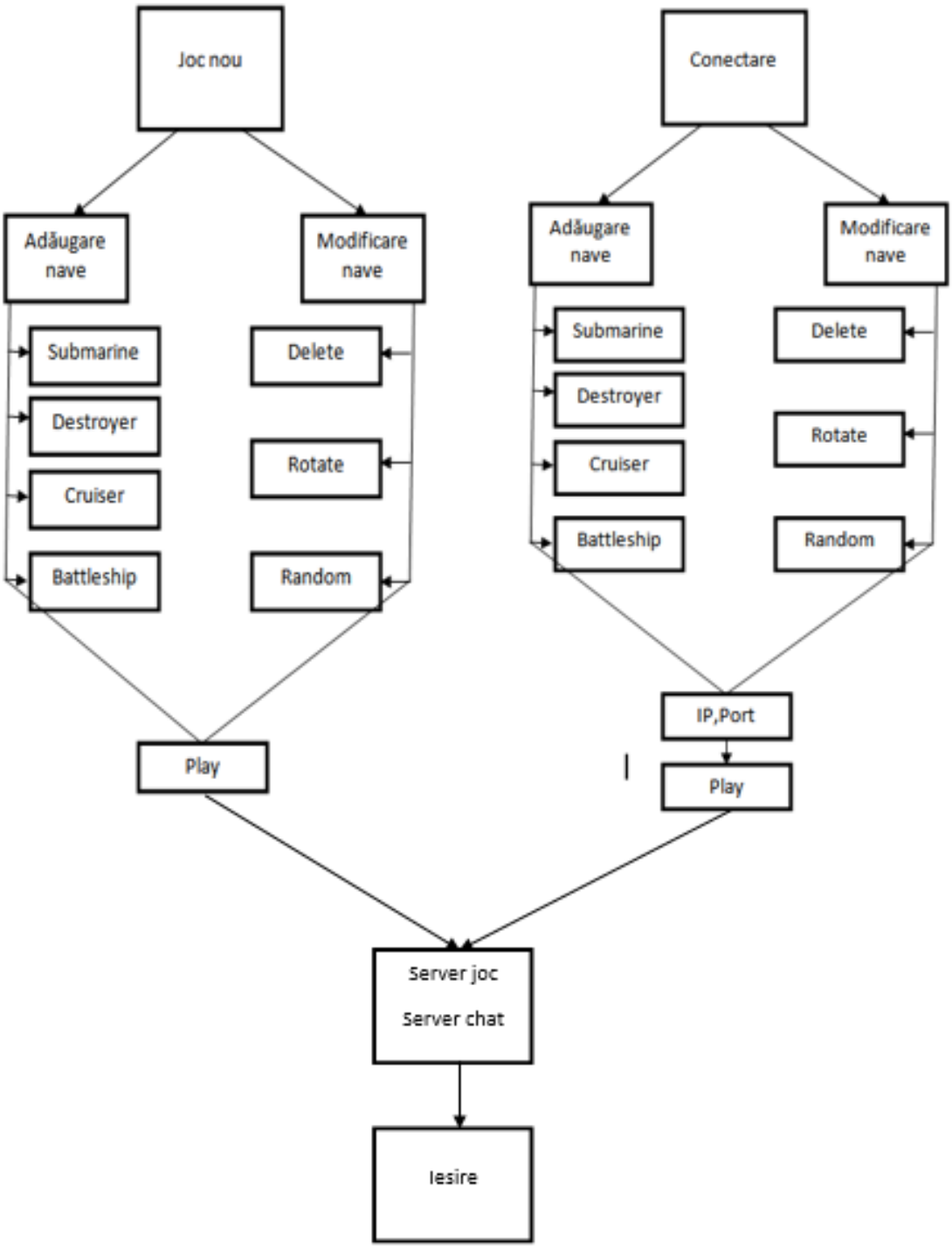
2. Fereastrele care apar in urma apasarii butonului New Game shi butonul Connect



3. Cimpurile de lupta



Schema Functionala

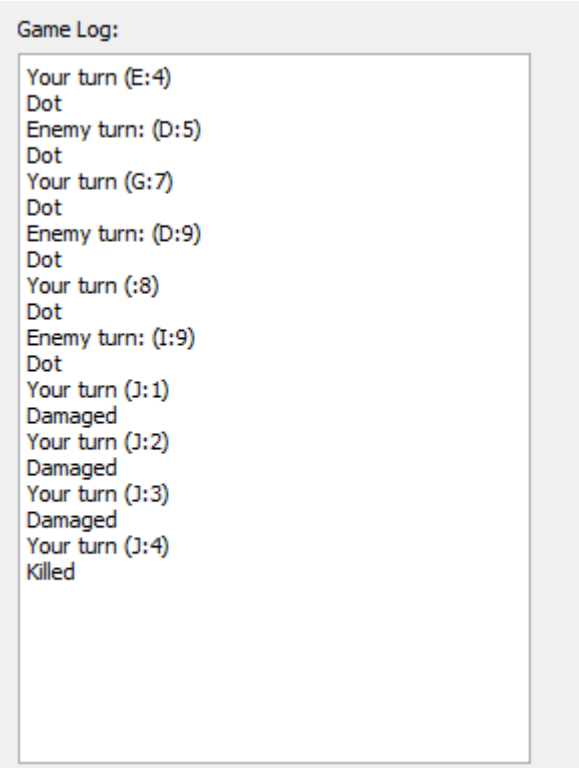


In urma divizarii am avut ca sarcine crearea main window, meniu de loguri, statistica.

MainWindow contine urmatoarele functii:

```
private:

        QHBoxLayout* centralWidgetLayout;
        CimpLupta* myBF; // cimpul meu
        CimpLupta* enemyBF; // cimpul ofonentului
        GameMenu* gMenu; // menu joc
        CorabieMenu* ssMenu; // meniu amplasare corabii
        LogMenu* lMenu; // log
        QLabel* myShips;
        QLabel* enemyShips;
        ChatClient *chatClient;

        char abc[10];

        bool isServer; // daca aplicatia este in calitate de server
        bool clientGata; // clientul e gata
        bool serverGata; // serverul e gata
        GAME_STATUS gstatus; // starea jocului
        Statistica statLupta; // statistica
        bool gameover; // GAME OVER

        //retea
        QString ipAdress; // IP
        int port; // port
        QTcpServer *tcpServer; // TCP server
        QTcpSocket *tcpSocket; // TCP port
        quint16 blockSize; // marimea pachetului

public slots:
    void creazaJoc(); // joc nou
    void conectareLaJoc(); // conectare la joc
    void inapoiLaMenu(); // meniul jocului
    void startJoc(); // de la redactor la inceputul jocului
    void incearcaStartJoc(); // serverul verifica daca ambii jucatori sun gata
    void gameOver(PLAYERS pl); // sfirsit

    //retea
    void startServer(); // start TCP server
    void stopServer(); // stop TCP server
    void startClient(); // start TCP client
    void stopClient(); // stop TCP client
    bool conectareLaServer(); // conectare la sever
    void acceptaConexiune(); //accepta conexiune
    void citireDinSocket(); // citire date
    void transmiteRind(int x, int y); // transmitem miscarea oponentului
    void transmiteRindRaspuns(int x, int y, CELL_STATUS st, Corabie* s); // raspuns
    void transmiteGataServer(); // transmitem starea la server
    void transmiteStareJoc(GAME_STATUS st); // serverul decide a cui este I miscare
    void displayEroare(QAbstractSocket::SocketError socketError); // eroare
```

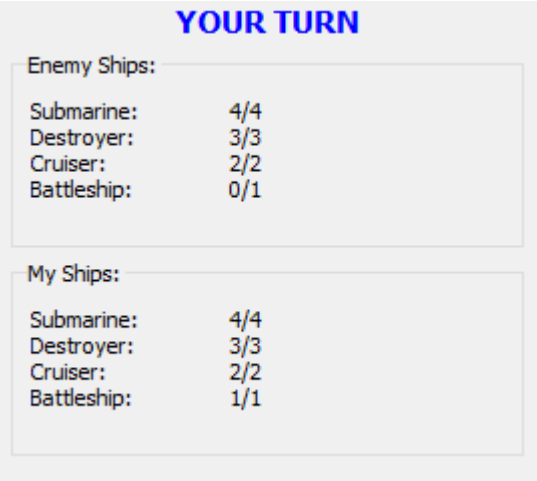
LogMenu:



LogMenu
contine informatie
despre miscarile
anterioare ale jucatorilor si
are strinsa legatura cu
clasa statistica si corabii.

```
class LogMenu : public QWidget
{
    Q_OBJECT
private:
    QTextBrowser *logText; //text browser unde se afiseaza log-ul
    GAME_STATUS *gstatus; //pointer la statutul jocului
    Statistica *statLupta; //pointer la obiect din clasa statisticii
    QLabel *statusLabel;
    QGroupBox *myCorabiiGB;
    QGroupBox *enemyCorabiiGB;
    QLabel *myCorabiiLabel;
    QLabel *enemyCorabiiLabel;
public:
    explicit LogMenu( GAME_STATUS *_gstatus,  Statistica *_statLupta, QWidget *parent = 0);
signals:

public slots:
    void adaugaLogString(QString text); //adaugam text in log
    void updateStatusLabel(); //update la statistica
};
```

Statistica:



Clasa Statistica initializeaza nr de corabii a ambilor jucatori, si duce contul corabiilor dupa marime si a corabiilor distruse.

Daca toate corabiile unuia din jucatori sunt distruse
se emite un semnal despre finisarea jocului.

```
//clasa statisticii corabiilor si pt determinarea invingatorului
class Statistica : public QObject
{
    Q_OBJECT
private:
    int myShips[4]; //statistica corabiilor mele, myShips[i]-numarului corabiilor intregi cu marimea i+1
    int enemyShips[4]; //statistica corabiilor dusmanului
public:
    explicit Statistica(QObject *parent = 0);

signals:
    void gameOver(PLAYERS pl);

public slots:
    void corabieDistrusa(PLAYERS pl,int size); //stergerea corabiei din statistica
    int  statisticaCorabii(PLAYERS pl, int size); //obtinerea statisticii despre corabii de marimea size
    void verificaStatus(); //verificam finisarea jocului
};

#endif // STATISTICA_H
```

Concluzie

In urma efectuarii acestei lucrari de laborator am luat cunostinta cu mediu de dezvoltare pentru aplicatii vizuale Qt, care permite realizarea rapida si usoara a aplicatiilor indiferent de platforma folosita. In lucrarea data am intilnit dificultati cu conexiunea la server, acceptarea conexiunii si citirii datelor, deci pentru usurinta am decis ca jocul nostru sa functioneze doar daca jucatorii sunt in aceeasi retea. Am creat deprinderi de lugru in grup shi repartizarea sarcinilor.