МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра интелектуальных информационных технологий

Отчёт по лабораторной работе №1
по дисциплине «Операционные Системы и Системное Программирование» за II семестр
Специальность: Программное Обеспечение Информационных Технологий

Выполнил:
студент 3-его курса
V-го семестр
факультета ЭИС
группы ПО-4(1)
зачётная книжка №190333
Галанин П. И.
«__» _________ 2021 г.

Проверил:
старщий преподаватель
кафедры ИИТ
Дряпко А. В.
«__» _________ 2021 г.

Брест 2021

Цель работы: приобрести практические навыки проектирования и разработки приложений с графическим пользовательским интерфейсом в ОС Windows средствами Qt.

Задание. Вариант 5. Игра «Сокобан». Один уровень игры. Общая идея: имеется комната-лабиринт (15X15 ячеек), в которой необходимо расставить ящики (5 штук) на указанные позиции. Главный герой может лишь толкать ящики вперед. Таким образом, возможны конфигурации, из которых не возможно построить желаемое решение (например, если ящик был задвинут в тупик).

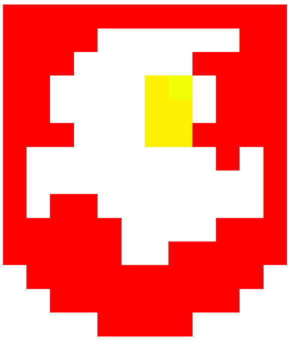

Рисунок 1 – Иконка игры

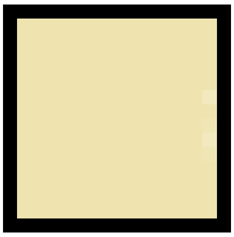

Рисунок 2 – Пустая ячейка

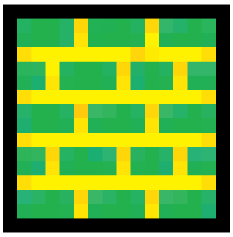

Рисунок 3 – Ячейка с коробкой

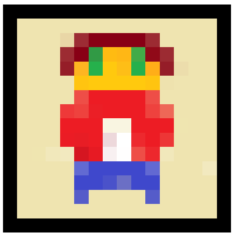

Рисунок 4 – Ячейка с игроком

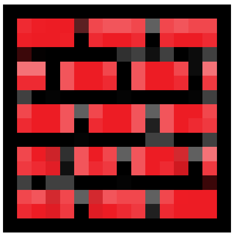

Рисунок 5 – Ячейка со стеной

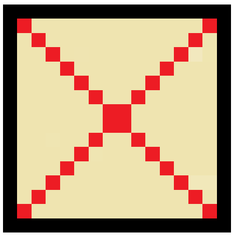

Рисунок 6 – Финиш для коробки

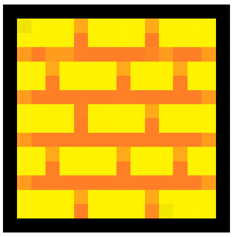

Рисунок 7 – Коробка на финише

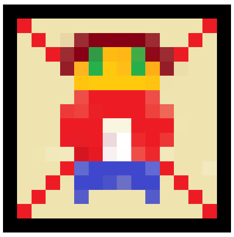

Рисунок 8 – Игрок на финише

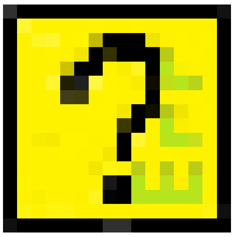

Рисунок 9 – Ячейка без функционала

					ОЛР.190333.ПО4.01 81 00			
Изм.	Лист	№ докум.	Подп.	Дата				
Разраб.		Галанин			Отчёт по лабораторной работе №1	Лит.	Лист	Листов
Пров.		Дряпко				О Л Р	2	20
						БрГТУ		
Н. контр.		Дряпко						
Утв.								

Листинг: ../src/QtSokoban/img.qrc

```
<RCC>
    <qresource prefix="/img">
        <file>_pics/favicon.png</file>
        <file>_pics/err.png</file>
        <file>_pics/wall.png</file>
        <file>_pics/floor.png</file>
        <file>_pics/player.png</file>
        <file>_pics/finPlayer.png</file>
        <file>_pics/box.png</file>
        <file>_pics/finBox.png</file>
        <file>_pics/finish.png</file>
    </qresource>
</RCC>
```

Листинг: ../src/QtSokoban/QtSokoban.pro

```
QT       += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

CONFIG += c++11

# You can make your code fail to compile if it uses deprecated APIs.
# In order to do so, uncomment the following line.
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000    # disables all the APIs deprecated before Qt 6.0.0

SOURCES += \
    components/mainwindow/drawAxes.cpp \
    components/mainwindow/drawTextures.cpp \
    components/mainwindow/generate1Level.cpp \
    components/mainwindow/goBottom.cpp \
    components/mainwindow/goLeft.cpp \
    components/mainwindow/goRight.cpp \
    components/mainwindow/goTop.cpp \
    components/mainwindow/sayWon.cpp \
    main.cpp \
    mainwindow.cpp

HEADERS += \
    mainwindow.h

FORMS += \
    mainwindow.ui

RESOURCES += \
    img.qrc

# Default rules for deployment.
qnx: target.path = /tmp/$${TARGET}/bin
else: unix:!android: target.path = /opt/$${TARGET}/bin
!isEmpty(target.path): INSTALLS += target
```

Листинг: ../src/QtSokoban/main.cpp

```
#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.setWindowIcon(QIcon(":/img/_pics/favicon.png"));
    w.setWindowTitle("Sokoban");
    w.show();
    return a.exec();
}
```

Листинг: ../src/QtSokoban/mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    this->setFixedSize(this->WinWidth, this->WinHeight);

    this->generate1Level();

    auto timer = new QTimer(this);
    connect(timer, SIGNAL(timeout()), this, SLOT(animate()));
    timer->start(100);
}

void MainWindow::animate()
{
    repaint();
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::keyPressEvent(QKeyEvent *e)
{
    if(e->key() == Qt::Key_Up || e->key() == Qt::Key_W)
    {
        this->goTop();
    }
    else if(e->key() == Qt::Key_Right || e->key() == Qt::Key_D)
    {
        this->goRight();
    }
    else if(e->key() == Qt::Key_Down || e->key() == Qt::Key_S)
    {
        this->goBottom();
    }
    else if(e->key() == Qt::Key_Left || e->key() == Qt::Key_A)
    {
        this->goLeft();
    }
}

void MainWindow::paintEvent(QPaintEvent *event)
{
    Q_UNUSED(event);
    QPainter painter(this);
    this->sayWon();
    this->drawAxes(&painter);
    this->drawTextures(&painter);
}
```

Листинг: ../src/QtSokoban/mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#define LENGTH 15

#include <QMainWindow>
#include <QKeyEvent>
#include <QPainter>
#include <QRect>
#include <QPixmap>
#include <QTimer>
#include <QMessageBox>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public slots:
    void animate();

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

protected:
    void generate1Level();
    void keyPressEvent(QKeyEvent* e) override;
    void goTop();
    void goRight();
    void goBottom();
    void goLeft();
    void paintEvent(QPaintEvent* event) override;
    void drawAxes(QPainter* painter);
    void drawTextures(QPainter* painter);
    void sayWon();

private:
    Ui::MainWindow *ui;
    const int length = LENGTH;
    const int WinWidth = 500;
    const int WinHeight = 500;
    enum MapChar {
        err,
        wall,
        floor,
        player,
        finPlayer,
        box,
        finBox,
        finish,
    } map[LENGTH][LENGTH];

    int xPlayer = 9;
    int yPlayer = 11;
};
#endif // MAINWINDOW_H
```

Листинг: ../src/QtSokoban/components/mainwindow/drawAxes.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

void MainWindow::drawAxes(QPainter *painter)
{
    int x = this->WinWidth / this->length;
    int y = this->WinHeight / this->length;
    for (int i = 0; i < this->length; i += 1)
    {
        for (int j = 0; j < this->length; j += 1)
        {
            QRect rect(x*i, y*j, x, y);
            painter->drawRect(rect);
        }
    }
}
```

Листинг: ../src/QtSokoban/components/mainwindow/drawTextures.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

void MainWindow::drawTextures(QPainter *painter)
{
    int x = this->WinWidth / this->length;
    int y = this->WinHeight / this->length;
    for (int i = 0; i < this->length; i += 1)
    {
        for (int j = 0; j < this->length; j += 1)
        {
            QString path = ":/img/_pics/err.png";
            if (this->map[i][j] == finish)
            {
                path = ":/img/_pics/finish.png";
            }
            else if (this->map[i][j] == floor)
            {
                path = ":/img/_pics/floor.png";
            }
            else if (this->map[i][j] == wall)
            {
                path = ":/img/_pics/wall.png";
            }
            else if (this->map[i][j] == player)
            {
                path = ":/img/_pics/player.png";
            }
            else if (this->map[i][j] == finPlayer)
            {
                path = ":/img/_pics/finPlayer.png";
            }
            else if (this->map[i][j] == box)
            {
                path = ":/img/_pics/box.png";
            }
            else if (this->map[i][j] == finBox)
            {
                path = ":/img/_pics/finBox.png";
            }

            QPixmap pixmap(path);
            painter->drawPixmap(x*i, y*j, x, y, pixmap);
        }
    }
}
```

Листинг: ../src/QtSokoban/components/mainwindow/generate1Level.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

void MainWindow::generate1Level()
{
    char StrMap[] ="\
xxxxxxxxxxxxxxx\n\
xxxxxxxxxxxxxxx\n\
xxxxxxxxxxxxxxx\n\
xxxxxxxxxxxxxxx\n\
xxxxxx...xxxxxx\n\
xxxxf@b..xxxxxx\n\
xxxxxx.bfxxxxxx\n\
xxxxfxxb.xxxxxx\n\
xxxx.x.f.xxxxxx\n\
xxxxb.Bbbfxxxxx\n\
xxxx...f..xxxxx\n\
xxxxxxxxxxxxxxx\n\
xxxxxxxxxxxxxxx\n\
xxxxxxxxxxxxxxx\n\
xxxxxxxxxxxxxxx\n\
";

    size_t StrMapLen = strlen(StrMap);
    for (int iter = 0, i = 0, j = 0; iter < StrMapLen; iter += 1)
    {

        if (StrMap[iter] == '\n')
        {
            j += 1;
            i = 0;
        }
        else if (i > (this->length))
        {
            continue;
        }
        else
        {
            switch(StrMap[iter])
            {
                case '.':
                    map[i][j] = floor;
                    break;
                case 'x':
                    map[i][j] = wall;
                    break;
                case '@':
                    map[i][j] = player;
                    this->xPlayer = i;
                    this->yPlayer = j;
                    break;
                case 'f':
                    map[i][j] = finish;
                    break;
                case 'A':
                    map[i][j] = finPlayer;
                    this->xPlayer = i;
                    this->yPlayer = j;
                    break;
                case 'b':
                    map[i][j] = box;
                    break;
                case 'B':
                    map[i][j] = finBox;
                    break;
                default:
                    map[i][j] = err;
                    break;
            }

            i += 1;
        }
    }
}
```

Листинг: ../src/QtSokoban/components/mainwindow/goTop.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

void MainWindow::goTop()
{
    if (this->yPlayer != 0)
    {
        MapChar f0 = this->map[this->xPlayer][this->yPlayer];
        MapChar f1 = this->map[this->xPlayer][this->yPlayer - 1];
        MapChar f2 = this->map[this->xPlayer][this->yPlayer - 2];

        // player -> floor
        // floor  -> player
        if (f0 == player && f1 == floor)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = player;

            return;
        }

        // finPlayer -> floor
        // finish    -> player
        if (f0 == finPlayer && f1 == floor)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = player;

            return;
        }

        // player -> box    -> floor
        // floor  -> player -> box
        if (f0 == player && f1 == box && f2 == floor)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = player;
            this->map[this->xPlayer][this->yPlayer - 1] = box;

            return;
        }

        // finPlayer -> box    -> floor
        // finish    -> player -> box
        if (f0 == finPlayer && f1 == box && f2 == floor)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = player;
            this->map[this->xPlayer][this->yPlayer - 1] = box;

            return;
        }

        // player -> box    -> finish
        // floor  -> player -> finBox
        if (f0 == player && f1 == box && f2 == finish)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = player;
            this->map[this->xPlayer][this->yPlayer - 1] = finBox;

            return;
        }

        // finPlayer -> box    -> finish
        // finish    -> player -> finBox
        if (f0 == finPlayer && f1 == box && f2 == finish)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = player;
            this->map[this->xPlayer][this->yPlayer - 1] = finBox;

            return;
        }

        // player -> finBox    -> floor
        // floor  -> finPlayer -> box
        if (f0 == player && f1 == finBox && f2 == floor)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            this->map[this->xPlayer][this->yPlayer - 1] = box;

            return;
        }

        // finPlayer -> finBox    -> floor
        // finish    -> finPlayer -> box
        if (f0 == finPlayer && f1 == finBox && f2 == floor)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            this->map[this->xPlayer][this->yPlayer - 1] = box;

            return;
        }

        // player -> finBox    -> finish
        // floor  -> finPlayer -> finBox
        if (f0 == player && f1 == finBox && f2 == finish)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            this->map[this->xPlayer][this->yPlayer - 1] = finBox;
            return;
        }

        // finPlayer -> finBox    -> finish
        // finish    -> finPlayer -> finBox
        if (f0 == finPlayer && f1 == finBox && f2 == finish)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            this->map[this->xPlayer][this->yPlayer - 1] = finBox;
            return;
        }

        // player -> finish
        // floor  -> finPlayer
        if (f0 == player && f1 == finish)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            return;
        }

        // finPlayer -> finish
        // finish    -> finPlayer
        if (f0 == finPlayer && f1 == finish)
        {
            this->yPlayer -= 1;

            this->map[this->xPlayer][this->yPlayer + 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            return;
        }
    }
}
```

Листинг: ../src/QtSokoban/components/mainwindow/goRight.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

void MainWindow::goRight()
{
    if (this->xPlayer != this->length - 1)
    {
        MapChar f0 = this->map[this->xPlayer    ][this->yPlayer];
        MapChar f1 = this->map[this->xPlayer + 1][this->yPlayer];
        MapChar f2 = this->map[this->xPlayer + 2][this->yPlayer];

        // player -> floor
        // floor  -> player
        if (f0 == player && f1 == floor)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = player;

            return;
        }

        // finPlayer -> floor
        // finish    -> player
        if (f0 == finPlayer && f1 == floor)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = player;

            return;
        }

        // player -> box    -> floor
        // floor  -> player -> box
        if (f0 == player && f1 == box && f2 == floor)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = player;
            this->map[this->xPlayer + 1][this->yPlayer] = box;

            return;
        }

        // finPlayer -> box    -> floor
        // finish    -> player -> box
        if (f0 == finPlayer && f1 == box && f2 == floor)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = player;
            this->map[this->xPlayer + 1][this->yPlayer] = box;

            return;
        }

        // finPlayer -> finBox -> floor
        // finish    -> finPlayer -> box
        if (f0 == finPlayer && f1 == finBox && f2 == floor)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            this->map[this->xPlayer + 1][this->yPlayer] = box;

            return;
        }

        // player -> finBox    -> floor
        // floor  -> finPlayer -> box
        if (f0 == player && f1 == finBox && f2 == floor)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            this->map[this->xPlayer + 1][this->yPlayer] = box;

            return;
        }

        // player -> box    -> finish
        // floor  -> player -> finBox
        if (f0 == player && f1 == box && f2 == finish)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = player;
            this->map[this->xPlayer + 1][this->yPlayer] = finBox;

            return;
        }

        // finPlayer -> box    -> finish
        // finish    -> player -> finBox
        if (f0 == finPlayer && f1 == box && f2 == finish)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = player;
            this->map[this->xPlayer + 1][this->yPlayer] = finBox;

            return;
        }

        // player -> finBox    -> finish
        // floor  -> finPlayer -> finBox
        if (f0 == player && f1 == finBox && f2 == finish)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            this->map[this->xPlayer + 1][this->yPlayer] = finBox;
            return;
        }

        // finPlayer -> finBox    -> finish
        // finish    -> finPlayer -> finBox
        if (f0 == finPlayer && f1 == finBox && f2 == finish)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            this->map[this->xPlayer + 1][this->yPlayer] = finBox;
            return;
        }

        // player -> finish
        // floor  -> finPlayer
        if (f0 == player && f1 == finish)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            return;
        }

        // finPlayer -> finish
        // finish    -> finPlayer
        if (f0 == finPlayer && f1 == finish)
        {
            this->xPlayer += 1;

            this->map[this->xPlayer - 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            return;
        }
    }
}
```

Листинг: ../src/QtSokoban/components/mainwindow/goBottom.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

void MainWindow::goBottom()
{
    if (this->yPlayer != this->length - 1)
    {
        MapChar f0 = this->map[this->xPlayer][this->yPlayer];
        MapChar f1 = this->map[this->xPlayer][this->yPlayer + 1];
        MapChar f2 = this->map[this->xPlayer][this->yPlayer + 2];

        // player -> floor
        // floor  -> player
        if (f0 == player && f1 == floor)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = player;

            return;
        }

        // finPlayer -> floor
        // finish    -> player
        if (f0 == finPlayer && f1 == floor)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = player;

            return;
        }

        // player -> box    -> floor
        // floor  -> player -> box
        if (f0 == player && f1 == box && f2 == floor)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = player;
            this->map[this->xPlayer][this->yPlayer + 1] = box;

            return;
        }

        // finPlayer -> box    -> floor
        // finish    -> player -> box
        if (f0 == finPlayer && f1 == box && f2 == floor)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = player;
            this->map[this->xPlayer][this->yPlayer + 1] = box;

            return;
        }

        // player -> finBox    -> floor
        // floor  -> finPlayer -> box
        if (f0 == player && f1 == finBox && f2 == floor)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            this->map[this->xPlayer][this->yPlayer + 1] = box;

            return;
        }

        // finPlayer -> finBox    -> floor
        // finish    -> finPlayer -> box
        if (f0 == finPlayer && f1 == finBox && f2 == floor)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            this->map[this->xPlayer][this->yPlayer + 1] = box;

            return;
        }

        // player -> box    -> finish
        // floor  -> player -> finBox
        if (f0 == player && f1 == box && f2 == finish)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = player;
            this->map[this->xPlayer][this->yPlayer + 1] = finBox;

            return;
        }

        // finPlayer -> box    -> finish
        // finish    -> player -> finBox
        if (f0 == finPlayer && f1 == box && f2 == finish)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = player;
            this->map[this->xPlayer][this->yPlayer + 1] = finBox;

            return;
        }

        // player -> finBox    -> finish
        // floor  -> finPlayer -> finBox
        if (f0 == player && f1 == finBox && f2 == finish)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            this->map[this->xPlayer][this->yPlayer + 1] = finBox;
            return;
        }

        // finPlayer -> finBox    -> finish
        // finish    -> finPlayer -> finBox
        if (f0 == finPlayer && f1 == finBox && f2 == finish)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            this->map[this->xPlayer][this->yPlayer + 1] = finBox;
            return;
        }

        // player -> finish
        // floor  -> finPlayer
        if (f0 == player && f1 == finish)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = floor;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            return;
        }

        // finPlayer -> finish
        // finish    -> finPlayer
        if (f0 == finPlayer && f1 == finish)
        {
            this->yPlayer += 1;

            this->map[this->xPlayer][this->yPlayer - 1] = finish;
            this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
            return;
        }
    }
}
```

Листинг: ../src/QtSokoban/components/mainwindow/goLeft.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

void MainWindow::goLeft()
{
    if (this->xPlayer != 0)
    {
        MapChar f0 = this->map[this->xPlayer    ][this->yPlayer];
        MapChar f1 = this->map[this->xPlayer - 1][this->yPlayer];
        MapChar f2 = this->map[this->xPlayer - 2][this->yPlayer];

        // player -> floor
        // floor  -> player
        if (f0 == player && f1 == floor)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = player;

            return;
        }

        // finPlayer -> floor
        // finish    -> player
        if (f0 == finPlayer && f1 == floor)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = player;

            return;
        }

        // player -> box    -> floor
        // floor  -> player -> box
        if (f0 == player && f1 == box && f2 == floor)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = player;
            this->map[this->xPlayer - 1][this->yPlayer] = box;

            return;
        }

        // finPlayer -> box    -> floor
        // finish    -> player -> box
        if (f0 == finPlayer && f1 == box && f2 == floor)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = player;
            this->map[this->xPlayer - 1][this->yPlayer] = box;

            return;
        }

        // player -> finBox    -> floor
        // floor  -> finPlayer -> box
        if (f0 == player && f1 == finBox && f2 == floor)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            this->map[this->xPlayer - 1][this->yPlayer] = box;

            return;
        }

        // finPlayer -> finBox    -> floor
        // finish    -> finPlayer -> box
        if (f0 == finPlayer && f1 == finBox && f2 == floor)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            this->map[this->xPlayer - 1][this->yPlayer] = box;

            return;
        }

        // player -> box    -> finish
        // floor  -> player -> finBox
        if (f0 == player && f1 == box && f2 == finish)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = player;
            this->map[this->xPlayer - 1][this->yPlayer] = finBox;

            return;
        }

        // finPlayer -> box    -> finish
        // finish    -> player -> finBox
        if (f0 == finPlayer && f1 == box && f2 == finish)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = player;
            this->map[this->xPlayer - 1][this->yPlayer] = finBox;

            return;
        }

        // player -> finBox    -> finish
        // floor  -> finPlayer -> finBox
        if (f0 == player && f1 == finBox && f2 == finish)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            this->map[this->xPlayer - 1][this->yPlayer] = finBox;
            return;
        }

        // finPlayer -> finBox    -> finish
        // finish    -> finPlayer -> finBox
        if (f0 == finPlayer && f1 == finBox && f2 == finish)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            this->map[this->xPlayer - 1][this->yPlayer] = finBox;
            return;
        }

        // player -> finish
        // floor  -> finPlayer
        if (f0 == player && f1 == finish)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = floor;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            return;
        }

        // finPlayer -> finish
        // finish    -> finPlayer
        if (f0 == finPlayer && f1 == finish)
        {
            this->xPlayer -= 1;

            this->map[this->xPlayer + 1][this->yPlayer] = finish;
            this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
            return;
        }
    }
}
```

Листинг: ../src/QtSokoban/components/mainwindow/sayWon.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

void MainWindow::sayWon()
{
    for (int i = 0; i < this->length; i += 1)
    {
        for (int j = 0; j < this->length; j += 1)
        {
            if (this->map[i][j] == finish)
            {
                return;
            }
        }
    }

    this->close();

    QMessageBox msgBox;

    QString StrWinWidth = QString::number(this->WinWidth);
    msgBox.setStyleSheet("QLabel{min-width: " + StrWinWidth + "px; }");
    msgBox.setText("You won");
    msgBox.setWindowTitle("You won");

    msgBox.exec();
}
```