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).

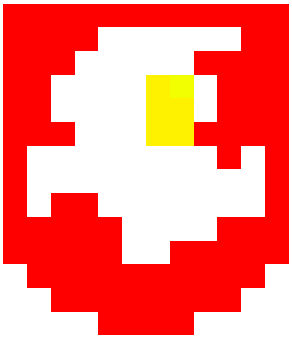

Рисунок 1 – Иконка игры

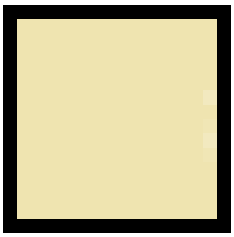

Рисунок 2 – Пустая ячейка

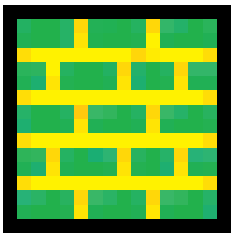

Рисунок 3 – Ячейка с коробкой

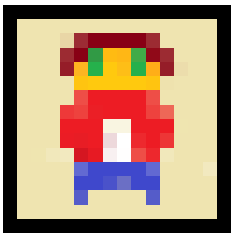

Рисунок 4 – Ячейка с игроком

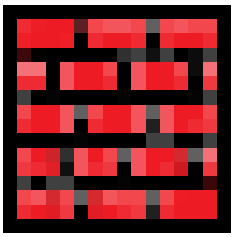

Рисунок 5 – Ячейка со стеной

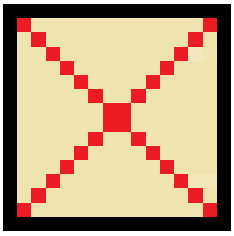

Рисунок 6 – Финиш для коробки

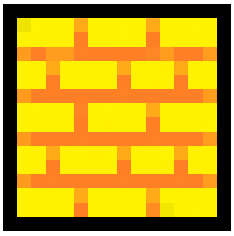

Рисунок 7 – Коробка на финише

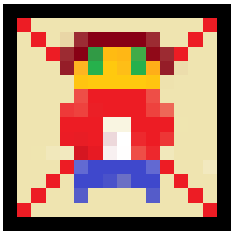

Рисунок 8 – Игрок на финише

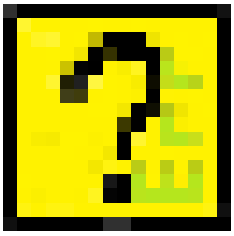

Рисунок 9 – Ячейка без функционала

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

```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

```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;
```

```
17
18              this->map[this->xPlayer][this->yPlayer + 1] = floor;
19              this->map[this->xPlayer][this->yPlayer    ] = player;
20
21              return;
22          }
23
24          // finPlayer -> floor
25          // finish    -> player
26          if (f0 == finPlayer && f1 == floor)
27          {
28              this->yPlayer -= 1;
29
30              this->map[this->xPlayer][this->yPlayer + 1] = finish;
31              this->map[this->xPlayer][this->yPlayer    ] = player;
32
33              return;
34          }
35
36          // player -> box    -> floor
37          // floor  -> player -> box
38          if (f0 == player && f1 == box && f2 == floor)
39          {
40              this->yPlayer -= 1;
41
42              this->map[this->xPlayer][this->yPlayer + 1] = floor;
43              this->map[this->xPlayer][this->yPlayer    ] = player;
44              this->map[this->xPlayer][this->yPlayer - 1] = box;
45
46              return;
47          }
48
49          // finPlayer -> box    -> floor
50          // finish    -> player -> box
51          if (f0 == finPlayer && f1 == box && f2 == floor)
52          {
53              this->yPlayer -= 1;
54
55              this->map[this->xPlayer][this->yPlayer + 1] = finish;
56              this->map[this->xPlayer][this->yPlayer    ] = player;
57              this->map[this->xPlayer][this->yPlayer - 1] = box;
58
59              return;
60          }
61
62          // player -> box    -> finish
63          // floor  -> player -> finBox
64          if (f0 == player && f1 == box && f2 == finish)
65          {
66              this->yPlayer -= 1;
67
68              this->map[this->xPlayer][this->yPlayer + 1] = floor;
69              this->map[this->xPlayer][this->yPlayer    ] = player;
70              this->map[this->xPlayer][this->yPlayer - 1] = finBox;
71
72              return;
73          }
74
75          // finPlayer -> box    -> finish
76          // finish    -> player -> finBox
```

```
77          if (f0 == finPlayer && f1 == box && f2 == finish)
78          {
79              this->yPlayer -= 1;
80
81              this->map[this->xPlayer][this->yPlayer + 1] = finish;
82              this->map[this->xPlayer][this->yPlayer    ] = player;
83              this->map[this->xPlayer][this->yPlayer - 1] = finBox;
84
85              return;
86          }
87
88          // player -> finBox    -> floor
89          // floor  -> finPlayer -> box
90          if (f0 == player && f1 == finBox && f2 == floor)
91          {
92              this->yPlayer -= 1;
93
94              this->map[this->xPlayer][this->yPlayer + 1] = floor;
95              this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
96              this->map[this->xPlayer][this->yPlayer - 1] = box;
97
98              return;
99          }
100
101         // finPlayer -> finBox    -> floor
102         // finish    -> finPlayer -> box
103         if (f0 == finPlayer && f1 == finBox && f2 == floor)
104         {
105             this->yPlayer -= 1;
106
107             this->map[this->xPlayer][this->yPlayer + 1] = finish;
108             this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
109             this->map[this->xPlayer][this->yPlayer - 1] = box;
110
111             return;
112         }
113
114         // player -> finBox    -> finish
115         // floor  -> finPlayer -> finBox
116         if (f0 == player && f1 == finBox && f2 == finish)
117         {
118             this->yPlayer -= 1;
119
120             this->map[this->xPlayer][this->yPlayer + 1] = floor;
121             this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
122             this->map[this->xPlayer][this->yPlayer - 1] = finBox;
123             return;
124         }
125
126         // finPlayer -> finBox    -> finish
127         // finish    -> finPlayer -> finBox
128         if (f0 == finPlayer && f1 == finBox && f2 == finish)
129         {
130             this->yPlayer -= 1;
131
132             this->map[this->xPlayer][this->yPlayer + 1] = finish;
133             this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
134             this->map[this->xPlayer][this->yPlayer - 1] = finBox;
135             return;
136         }
```

```
137
138         // player -> finish
139         // floor  -> finPlayer
140         if (f0 == player && f1 == finish)
141         {
142             this->yPlayer -= 1;
143
144             this->map[this->xPlayer][this->yPlayer + 1] = floor;
145             this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
146             return;
147         }
148
149         // finPlayer -> finish
150         // finish    -> finPlayer
151         if (f0 == finPlayer && f1 == finish)
152         {
153             this->yPlayer -= 1;
154
155             this->map[this->xPlayer][this->yPlayer + 1] = finish;
156             this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
157             return;
158         }
159     }
160 }
```

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

```
1  #include "mainwindow.h"
2  #include "ui_mainwindow.h"
3
4  void MainWindow::goRight()
5  {
6      if (this->xPlayer != this->length - 1)
7      {
8          MapChar f0 = this->map[this->xPlayer    ][this->yPlayer];
9          MapChar f1 = this->map[this->xPlayer + 1][this->yPlayer];
10         MapChar f2 = this->map[this->xPlayer + 2][this->yPlayer];
11
12         // player -> floor
13         // floor  -> player
14         if (f0 == player && f1 == floor)
15         {
16             this->xPlayer += 1;
17
18             this->map[this->xPlayer - 1][this->yPlayer] = floor;
19             this->map[this->xPlayer    ][this->yPlayer] = player;
20
21             return;
22         }
23
24         // finPlayer -> floor
25         // finish    -> player
26         if (f0 == finPlayer && f1 == floor)
27         {
28             this->xPlayer += 1;
29
30             this->map[this->xPlayer - 1][this->yPlayer] = finish;
31             this->map[this->xPlayer    ][this->yPlayer] = player;
32
33             return;
```

```
34          }
35
36          // player -> box    -> floor
37          // floor  -> player -> box
38          if (f0 == player && f1 == box && f2 == floor)
39          {
40              this->xPlayer += 1;
41
42              this->map[this->xPlayer - 1][this->yPlayer] = floor;
43              this->map[this->xPlayer    ][this->yPlayer] = player;
44              this->map[this->xPlayer + 1][this->yPlayer] = box;
45
46              return;
47          }
48
49          // finPlayer -> box    -> floor
50          // finish    -> player -> box
51          if (f0 == finPlayer && f1 == box && f2 == floor)
52          {
53              this->xPlayer += 1;
54
55              this->map[this->xPlayer - 1][this->yPlayer] = finish;
56              this->map[this->xPlayer    ][this->yPlayer] = player;
57              this->map[this->xPlayer + 1][this->yPlayer] = box;
58
59              return;
60          }
61
62          // finPlayer -> finBox -> floor
63          // finish    -> finPlayer -> box
64          if (f0 == finPlayer && f1 == finBox && f2 == floor)
65          {
66              this->xPlayer += 1;
67
68              this->map[this->xPlayer - 1][this->yPlayer] = finish;
69              this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
70              this->map[this->xPlayer + 1][this->yPlayer] = box;
71
72              return;
73          }
74
75          // player -> finBox    -> floor
76          // floor  -> finPlayer -> box
77          if (f0 == player && f1 == finBox && f2 == floor)
78          {
79              this->xPlayer += 1;
80
81              this->map[this->xPlayer - 1][this->yPlayer] = floor;
82              this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
83              this->map[this->xPlayer + 1][this->yPlayer] = box;
84
85              return;
86          }
87
88          // player -> box    -> finish
89          // floor  -> player -> finBox
90          if (f0 == player && f1 == box && f2 == finish)
91          {
92              this->xPlayer += 1;
93
```

```
            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;
```

```
154
155             this->map[this->xPlayer - 1][this->yPlayer] = finish;
156             this->map[this->xPlayer    ][this->yPlayer] = finPlayer;
157             return;
158         }
159     }
160 }
```

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

```
1  #include "mainwindow.h"
2  #include "ui_mainwindow.h"
3
4  void MainWindow::goBottom()
5  {
6      if (this->yPlayer != this->length - 1)
7      {
8          MapChar f0 = this->map[this->xPlayer][this->yPlayer];
9          MapChar f1 = this->map[this->xPlayer][this->yPlayer + 1];
10         MapChar f2 = this->map[this->xPlayer][this->yPlayer + 2];
11
12         // player -> floor
13         // floor  -> player
14         if (f0 == player && f1 == floor)
15         {
16             this->yPlayer += 1;
17
18             this->map[this->xPlayer][this->yPlayer - 1] = floor;
19             this->map[this->xPlayer][this->yPlayer    ] = player;
20
21             return;
22         }
23
24         // finPlayer -> floor
25         // finish    -> player
26         if (f0 == finPlayer && f1 == floor)
27         {
28             this->yPlayer += 1;
29
30             this->map[this->xPlayer][this->yPlayer - 1] = finish;
31             this->map[this->xPlayer][this->yPlayer    ] = player;
32
33             return;
34         }
35
36         // player -> box    -> floor
37         // floor  -> player -> box
38         if (f0 == player && f1 == box && f2 == floor)
39         {
40             this->yPlayer += 1;
41
42             this->map[this->xPlayer][this->yPlayer - 1] = floor;
43             this->map[this->xPlayer][this->yPlayer    ] = player;
44             this->map[this->xPlayer][this->yPlayer + 1] = box;
45
46             return;
47         }
48
49         // finPlayer -> box    -> floor
50         // finish    -> player -> box
```

```
51          if (f0 == finPlayer && f1 == box && f2 == floor)
52          {
53              this->yPlayer += 1;
54
55              this->map[this->xPlayer][this->yPlayer - 1] = finish;
56              this->map[this->xPlayer][this->yPlayer    ] = player;
57              this->map[this->xPlayer][this->yPlayer + 1] = box;
58
59              return;
60          }
61
62          // player -> finBox    -> floor
63          // floor  -> finPlayer -> box
64          if (f0 == player && f1 == finBox && f2 == floor)
65          {
66              this->yPlayer += 1;
67
68              this->map[this->xPlayer][this->yPlayer - 1] = floor;
69              this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
70              this->map[this->xPlayer][this->yPlayer + 1] = box;
71
72              return;
73          }
74
75          // finPlayer -> finBox    -> floor
76          // finish    -> finPlayer -> box
77          if (f0 == finPlayer && f1 == finBox && f2 == floor)
78          {
79              this->yPlayer += 1;
80
81              this->map[this->xPlayer][this->yPlayer - 1] = finish;
82              this->map[this->xPlayer][this->yPlayer    ] = finPlayer;
83              this->map[this->xPlayer][this->yPlayer + 1] = box;
84
85              return;
86          }
87
88          // player -> box    -> finish
89          // floor  -> player -> finBox
90          if (f0 == player && f1 == box && f2 == finish)
91          {
92              this->yPlayer += 1;
93
94              this->map[this->xPlayer][this->yPlayer - 1] = floor;
95              this->map[this->xPlayer][this->yPlayer    ] = player;
96              this->map[this->xPlayer][this->yPlayer + 1] = finBox;
97
98              return;
99          }
100
101         // finPlayer -> box    -> finish
102         // finish    -> player -> finBox
103         if (f0 == finPlayer && f1 == box && f2 == finish)
104         {
105             this->yPlayer += 1;
106
107             this->map[this->xPlayer][this->yPlayer - 1] = finish;
108             this->map[this->xPlayer][this->yPlayer    ] = player;
109             this->map[this->xPlayer][this->yPlayer + 1] = finBox;
110
```

```
            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

```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;
```

```
119
120             this->map[this->xPlayer + 1][this->yPlayer] = floor;
121             this->map[this->xPlayer     ][this->yPlayer] = finPlayer;
122             this->map[this->xPlayer - 1][this->yPlayer] = finBox;
123             return;
124         }
125
126         // finPlayer -> finBox    -> finish
127         // finish    -> finPlayer -> finBox
128         if (f0 == finPlayer && f1 == finBox && f2 == finish)
129         {
130             this->xPlayer -= 1;
131
132             this->map[this->xPlayer + 1][this->yPlayer] = finish;
133             this->map[this->xPlayer     ][this->yPlayer] = finPlayer;
134             this->map[this->xPlayer - 1][this->yPlayer] = finBox;
135             return;
136         }
137
138         // player -> finish
139         // floor  -> finPlayer
140         if (f0 == player && f1 == finish)
141         {
142             this->xPlayer -= 1;
143
144             this->map[this->xPlayer + 1][this->yPlayer] = floor;
145             this->map[this->xPlayer     ][this->yPlayer] = finPlayer;
146             return;
147         }
148
149         // finPlayer -> finish
150         // finish    -> finPlayer
151         if (f0 == finPlayer && f1 == finish)
152         {
153             this->xPlayer -= 1;
154
155             this->map[this->xPlayer + 1][this->yPlayer] = finish;
156             this->map[this->xPlayer     ][this->yPlayer] = finPlayer;
157             return;
158         }
159     }
160 }
```

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();
}
```