

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ОТЧЁТ ЛАБОРАТОРНОЙ РАБОТЫ №1
ПО ДИСЦИПЛИНЕ «ОПЕРАЦИОННЫЕ СИСТЕМЫ И СИСТЕМНОЕ
ПРОГРАММИРОВАНИЕ»
ЗА II СЕМЕСТР

Выполнил:
студент 3-го курса
V-го семестр
факультета ЭИС
группы ПО-4(2)
зачётная книжка №190333
Галанин П. И.
«__» _____ 2021 г.

Проверил:
ассистент
кафедры ИИТ
Дряпко А. В.
«__» _____ 2021 г.

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

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).

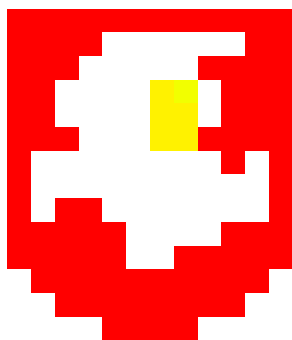


Рисунок 1 – Иконка игры

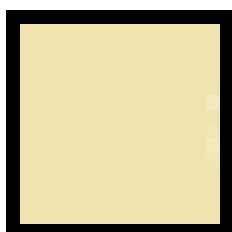


Рисунок 2 – Пустая ячейка

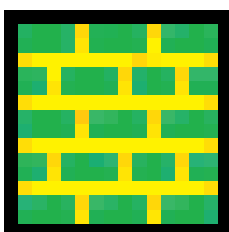


Рисунок 3 – Ячейка с коробкой

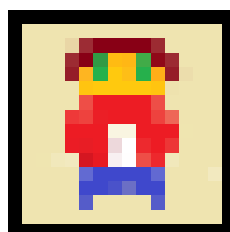


Рисунок 4 – Ячейка с игроком

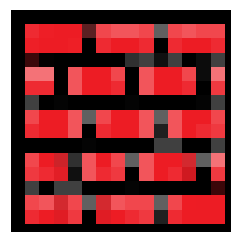


Рисунок 5 – Ячейка со стеной

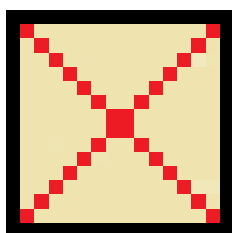


Рисунок 6 – Финиш для коробки

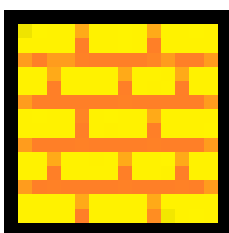


Рисунок 7 – Коробка на финише

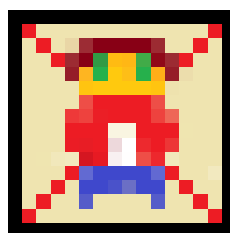


Рисунок 8 – Игрок на финише

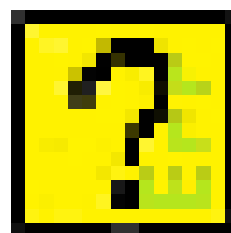


Рисунок 9 – Ячейка без функционала

					ОЛР.ПО4.190333.01 81 00		
Изм.	Лист	№ докум.	Подп.	Дата			
Разраб.	Галанин				ОТЧЁТ ЛАБОРАТОРНОЙ РАБОТЫ №1		
Пров.	Дряпко						
Н. контр.	Дряпко						
Утв.							
						Лит.	Лист
						О	Л
						Р	2
						Листов	
						21	
						БрГТУ	

../gpi_osisp5_option5/gpi_osisp5_option5.pro

```
1 QT      += core gui
2
3 greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
4
5 CONFIG += c++11
6
7 # You can make your code fail to compile if it uses deprecated APIs.
8 # In order to do so, uncomment the following line.
9 #DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000   # disables all the APIs deprecated before Qt
10 6.0.0
11
12 SOURCES += \
13     gpi_gamewindow.cpp \
14     gpi_gamewindow__gpi_drawAxes.cpp \
15     gpi_gamewindow__gpi_drawTextures.cpp \
16     gpi_gamewindow__gpi_generateLevel1.cpp \
17     gpi_gamewindow__gpi_goBottom.cpp \
18     gpi_gamewindow__gpi_goLeft.cpp \
19     gpi_gamewindow__gpi_goRight.cpp \
20     gpi_gamewindow__gpi_goTop.cpp \
21     gpi_gamewindow__gpi_sayWon.cpp \
22     main.cpp \
23     gpi_mainwindow.cpp
24
25 HEADERS += \
26     gpi_gamewindow.hpp \
27     gpi_mainwindow.hpp
28
29 FORMS += \
30     gpi_gamewindow.ui \
31     gpi_mainwindow.ui
32
33 # Default rules for deployment.
34 qnx: target.path = /tmp/${TARGET}/bin
35 else: unix:!android: target.path = /opt/${TARGET}/bin
36 !isEmpty(target.path): INSTALLS += target
37
38 RESOURCES += \
39     gpi_gamewindow.qrc
```

../gpi_osisp5_option5/main.cpp

```
1 #include "gpi_mainwindow.hpp"
2
3 #include <QApplication>
4
5 int main (int argc, char *argv[])
6 {
7     QApplication a (argc, argv);
8     gpi_MainWindow w;
9     w.show ();
10    return a.exec ();
11 }
```

../gpi_osisp5_option5/gpi_mainwindow.hpp

```
1 #ifndef GPI_MAINWINDOW_HPP
2 #define GPI_MAINWINDOW_HPP
```

					ОЛР.ПО4.190333.01 81 00	Лист
Изм	Лист	№ докум.	Подп.	Дата		3

```

3
4 #include <QMainWindow>
5 #include "gpi_gamewindow.hpp"
6
7 QT_BEGIN_NAMESPACE
8 namespace Ui { class gpi_MainWindow; }
9 QT_END_NAMESPACE
10
11 class gpi_MainWindow : public QMainWindow
12 {
13     Q_OBJECT
14
15 public:
16     gpi_MainWindow (QWidget *parent = nullptr);
17     ~gpi_MainWindow ();
18
19 private slots:
20     void on_pushButton_ShowGameWindow_clicked ();
21     void on_pushButton_CloseMainWindow_clicked ();
22
23 private:
24     Ui::gpi_MainWindow *ui;
25     gpi_GameWindow gpi_gw;
26 };
27 #endif // GPI_MAINWINDOW_HPP

```

../gpi_osisp5_option5/gpi_mainwindow.cpp

```

1 #include "gpi_mainwindow.hpp"
2 #include "ui_gpi_mainwindow.h"
3
4 gpi_MainWindow::gpi_MainWindow (QWidget *parent)
5 : QMainWindow (parent)
6 , ui (new Ui::gpi_MainWindow)
7 {
8     ui->setupUi (this);
9     this->setWindowIcon (QIcon (":/@gpi@/_assets/gpi_MainWindow__favicon.png"));
10 }
11
12 gpi_MainWindow::~gpi_MainWindow ()
13 {
14     delete ui;
15 }
16
17
18
19 void gpi_MainWindow::on_pushButton_ShowGameWindow_clicked ()
20 {
21     gpi_gw.show ();
22 }
23
24
25 void gpi_MainWindow::on_pushButton_CloseMainWindow_clicked ()
26 {
27     this->close ();
28 }

```

../gpi_osisp5_option5/gpi_gamewindow.hpp

```

1 #ifndef GPI_GAMEWINDOW_HPP

```

					ОЛР.ПО4.190333.01 81 00	Лист
Изм	Лист	№ докум.	Подп.	Дата		4

```

2  #define GPI_GAMEWINDOW_HPP
3  #define LENGTH 15
4
5  #include <QWidget>
6  #include <QPainter>
7  #include <QMessageBox>
8  #include <QKeyEvent>
9  #include <QTimer>
10 #include <QIcon>
11
12 namespace Ui {
13     class gpi_GameWindow;
14 }
15
16 class gpi_GameWindow : public QWidget
17 {
18     Q_OBJECT
19
20 public:
21     explicit gpi_GameWindow (QWidget *parent = nullptr);
22     ~gpi_GameWindow ();
23
24 public slots:
25     void gpi_gamewindow_animate();
26
27 protected:
28     void paintEvent (QPaintEvent *event) override;
29     void keyPressEvent (QKeyEvent *e) override;
30
31 private:
32     Ui::gpi_GameWindow *ui;
33
34     const int    gpi_length          = LENGTH;
35     const int    gpi_WinWidth        = 500;
36     const int    gpi_WinHeight       = 500;
37     int          gpi_xPlayer;
38     int          gpi_yPlayer;
39     // =====
40
41     enum gpi_MapChar {
42         err ,
43         wall ,
44         floor ,
45         player ,
46         finPlayer ,
47         box ,
48         finBox ,
49         finish ,
50     } gpi_map[LENGTH][LENGTH] = {{err}};
51
52     void gpi_drawAxes (QPainter* painter);
53     void gpi_drawTextures (QPainter *gpi_painter);
54     void gpi_generateLevel1 ();
55     void gpi_goBottom ();
56     void gpi_goLeft ();
57     void gpi_goRight ();
58     void gpi_goTop ();
59     void gpi_sayWon ();
60 };
61

```

```

62 #endif // GPI_GAMEWINDOW_HPP

        ../gpi_osisp5_option5/gpi_gamewindow.cpp

1  #include "gpi_gamewindow.hpp"
2  #include "ui_gpi_gamewindow.h"
3
4  gpi_GameWindow::gpi_GameWindow (QWidget *parent) :
5      QWidget (parent),
6      ui (new Ui::gpi_GameWindow)
7  {
8      QTimer*          timer          = new QTimer (this);
9      // =====
10
11      ui->setupUi (this);
12      this->setWindowIcon (QIcon (":/@gpi@/_assets/gpi_MainWindow__favicon.png"));
13
14      connect (timer, SIGNAL (timeout ()), this, SLOT (gpi_gamewindow_animate ()));
15      timer->start (100);
16
17      this->setFixedSize (this->gpi_WinWidth, this->gpi_WinHeight);
18      this->gpi_generateLevel1 ();
19  }
20
21  gpi_GameWindow::~gpi_GameWindow ()
22  {
23      delete ui;
24  }
25
26  void gpi_GameWindow::paintEvent (QPaintEvent *event)
27  {
28      QPainter          gpiPainter (this);
29      // =====
30
31      Q_UNUSED (event);
32      this->gpi_sayWon ();
33      this->gpi_drawAxes (&gpiPainter);
34      this->gpi_drawTextures (&gpiPainter);
35  }
36
37  void gpi_GameWindow::gpi_gamewindow_animate ()
38  {
39      repaint ();
40  }
41
42  void gpi_GameWindow::keyPressEvent (QKeyEvent *e)
43  {
44      switch (e->key ())
45      {
46          case Qt::Key_Up:
47          case Qt::Key_W:
48              this->gpi_goTop ();
49              break;
50          case Qt::Key_Right:
51          case Qt::Key_D:
52              this->gpi_goRight ();
53              break;
54          case Qt::Key_Down:
55          case Qt::Key_S:
56              this->gpi_goBottom ();

```

```

57         break;
58     case Qt::Key_Left:
59     case Qt::Key_A:
60         this->gpi_goLeft();
61         break;
62     }
63 }

```

../gpi_osisp5_option5/gpi_gamewindow.qrc

```

1 <RCC>
2     <qresource prefix="@gpi@">
3         <file> _assets/gpi_MainWindow__favicon.png</file>
4         <file> _assets/gpi_GameWindow__err.png</file>
5         <file> _assets/gpi_GameWindow__wall.png</file>
6         <file> _assets/gpi_GameWindow__floor.png</file>
7         <file> _assets/gpi_GameWindow__player.png</file>
8         <file> _assets/gpi_GameWindow__finPlayer.png</file>
9         <file> _assets/gpi_GameWindow__box.png</file>
10        <file> _assets/gpi_GameWindow__finBox.png</file>
11        <file> _assets/gpi_GameWindow__finish.png</file>
12    </qresource>
13 </RCC>

```

../gpi_osisp5_option5/gpi_gamewindow__gpi_drawAxes.cpp

```

1 #include "gpi_gamewindow.hpp"
2 #include "ui_gpi_gamewindow.h"
3
4 void gpi_GameWindow::gpi_drawAxes(QPainter *gpiPainter)
5 {
6     int gpi_x = this->gpi_WinWidth / this->gpi_length;
7     int gpi_y = this->gpi_WinHeight / this->gpi_length;
8     const int gpi_l = this->gpi_length;
9     int gpi_i;
10    int gpi_j;
11    // =====
12
13    gpi_i = 0;
14    while(gpi_i < gpi_l)
15    {
16        gpi_j = 0;
17        while(gpi_j < gpi_l)
18        {
19            QRect gpi_rect(gpi_x * gpi_i, gpi_y * gpi_j, gpi_x, gpi_y);
20            gpiPainter->drawRect(gpi_rect);
21            ++gpi_j;
22        }
23        ++gpi_i;
24    }
25 }

```

../gpi_osisp5_option5/gpi_gamewindow__gpi_drawTextures.cpp

```

1 #include "gpi_gamewindow.hpp"
2 #include "ui_gpi_gamewindow.h"
3
4 void gpi_GameWindow::gpi_drawTextures(QPainter *gpiPainter)
5 {
6     int gpi_x = this->gpi_WinWidth / this->gpi_length;

```

Изм	Лист	№ докум.	Подп.	Дата

ОЛР.ПО4.190333.01 81 00

Лист

7

```

7      int          gpi_y          = this->gpi_WinHeight / this->gpi_length;
8      const int    gpi_l          = this->gpi_length;
9      QString      gpi_path       = " :/img/_pics/err.png";
10     int           gpi_i;
11     int           gpi_j;
12     // =====
13
14     gpi_i = 0;
15     while(gpi_i < gpi_l)
16     {
17         gpi_j = 0;
18         while(gpi_j < gpi_l)
19         {
20             switch (this->gpi_map[gpi_i][gpi_j])
21             {
22                 case finish:
23                     gpi_path = " :/@gpi@/_assets/gpi_GameWindow__finish.png";
24                     break;
25                 case floor:
26                     gpi_path = " :/@gpi@/_assets/gpi_GameWindow__floor.png";
27                     break;
28                 case wall:
29                     gpi_path = " :/@gpi@/_assets/gpi_GameWindow__wall.png";
30                     break;
31                 case player:
32                     gpi_path = " :/@gpi@/_assets/gpi_GameWindow__player.png";
33                     break;
34                 case finPlayer:
35                     gpi_path = " :/@gpi@/_assets/gpi_GameWindow__finPlayer.png";
36                     break;
37                 case box:
38                     gpi_path = " :/@gpi@/_assets/gpi_GameWindow__box.png";
39                     break;
40                 case finBox:
41                     gpi_path = " :/@gpi@/_assets/gpi_GameWindow__finBox.png";
42                     break;
43                 default:
44                     gpi_path = " :/@gpi@/_assets/gpi_GameWindow__err.png";
45                     break;
46             }
47
48             QPixmap gpi_pixmap(gpi_path);
49             gpiPainter->drawPixmap(gpi_x * gpi_i, gpi_y * gpi_j, gpi_x, gpi_y, gpi_pixmap);
50             ++gpi_j;
51         }
52         ++gpi_i;
53     }
54 }

```

../gpi_osisp5_option5/gpi_gamewindow__gpi_generateLevel1.cpp

```

1  #include "gpi_gamewindow.hpp"
2  #include "ui_gpi_gamewindow.h"
3
4  void gpi_GameWindow::gpi_generateLevel1 ()
5  {
6      int          gpi_iter;
7      int          gpi_i;
8      int          gpi_j;
9      int          gpi_l          = this->gpi_length;

```

Изм	Лист	№ докум.	Подп.	Дата

ОЛР.ПО4.190333.01 81 00

Лист

8


```

10     char          gpi_strMap[] = "\
11     xxxxxxxxxxxxxxx\n\
12     xxxxxxxxxxxxxxx\n\
13     xxxxxxxxxxxxxxx\n\
14     xxxxxxxxxxxxxxx\n\
15     xxxxxx...xxxxx\n\
16     xxxxf@b..xxxxxx\n\
17     xxxxxx.bfxxxxxx\n\
18     xxxxfxxb..xxxxxx\n\
19     xxxx.x.f.xxxxxx\n\
20     xxxxb.Bbbfxxxxx\n\
21     xxxx...f..xxxxx\n\
22     xxxxxxxxxxxxxxx\n\
23     xxxxxxxxxxxxxxx\n\
24     xxxxxxxxxxxxxxx\n\
25     xxxxxxxxxxxxxxx\n\
26 ";
27     int          gpi_strMapLen = strlen(gpi_strMap);
28     // =====
29
30     gpi_iter = 0;
31     gpi_i = 0;
32     gpi_j = 0;
33     while (gpi_iter < gpi_strMapLen)
34     {
35         if (gpi_strMap[gpi_iter] == '\n')
36         {
37             ++gpi_j;
38             gpi_i = 0;
39         }
40         else if (gpi_i > gpi_l)
41         {
42             continue;
43         }
44         else
45         {
46             switch(gpi_strMap[gpi_iter])
47             {
48                 case '.':
49                     this->gpi_map[gpi_i][gpi_j] = floor;
50                     break;
51                 case 'x':
52                     this->gpi_map[gpi_i][gpi_j] = wall;
53                     break;
54                 case '@':
55                     this->gpi_map[gpi_i][gpi_j] = player;
56                     this->gpi_xPlayer = gpi_i;
57                     this->gpi_yPlayer = gpi_j;
58                     break;
59                 case 'f':
60                     this->gpi_map[gpi_i][gpi_j] = finish;
61                     break;
62                 case 'A':
63                     this->gpi_map[gpi_i][gpi_j] = finPlayer;
64                     this->gpi_xPlayer = gpi_i;
65                     this->gpi_yPlayer = gpi_j;
66                     break;
67                 case 'b':
68                     this->gpi_map[gpi_i][gpi_j] = box;
69                     break;

```

```

70         case 'B':
71             this->gpi_map[gpi_i][gpi_j] = finBox;
72             break;
73         default:
74             this->gpi_map[gpi_i][gpi_j] = err;
75             break;
76     }
77     ++gpi_i;
78 }
79 ++gpi_iter;
80 }
81 }

```

../gpi_osisp5_option5/gpi_gamewindow__gpi_goBottom.cpp

```

1  #include "gpi_gamewindow.hpp"
2  #include "ui_gpi_gamewindow.h"
3
4  void gpi_GameWindow::gpi_goBottom ()
5  {
6      int          gpi_l          = this->gpi_length;
7      int          gpi_px         = this->gpi_xPlayer;
8      int          gpi_py         = this->gpi_yPlayer;
9      gpi_MapChar gpi_c0;
10     gpi_MapChar gpi_c1;
11     gpi_MapChar gpi_c2;
12     // =====
13
14     if (gpi_py != gpi_l - 1)
15     {
16         gpi_c0 = this->gpi_map[gpi_px][gpi_py];
17         gpi_c1 = this->gpi_map[gpi_px][gpi_py + 1];
18         gpi_c2 = this->gpi_map[gpi_px][gpi_py + 2];
19
20         // player -> floor
21         // floor -> player
22         if (gpi_c0 == player && gpi_c1 == floor)
23         {
24             gpi_py += 1;
25             this->gpi_yPlayer = gpi_py;
26             this->gpi_map[gpi_px][gpi_py - 1] = floor;
27             this->gpi_map[gpi_px][gpi_py      ] = player;
28             return;
29         }
30
31         // finPlayer -> floor
32         // finish    -> player
33         if (gpi_c0 == finPlayer && gpi_c1 == floor)
34         {
35             gpi_py += 1;
36             this->gpi_yPlayer = gpi_py;
37             this->gpi_map[gpi_px][gpi_py - 1] = finish;
38             this->gpi_map[gpi_px][gpi_py      ] = player;
39             return;
40         }
41
42         // player -> box    -> floor
43         // floor -> player -> box
44         if (gpi_c0 == player && gpi_c1 == box && gpi_c2 == floor)
45         {

```

```

46     gpi_py += 1;
47     this->gpi_yPlayer = gpi_py;
48     this->gpi_map[gpi_px][gpi_py - 1] = floor;
49     this->gpi_map[gpi_px][gpi_py] = player;
50     this->gpi_map[gpi_px][gpi_py + 1] = box;
51     return;
52 }
53
54 // finPlayer -> box -> floor
55 // finish -> player -> box
56 if (gpi_c0 == finPlayer && gpi_c1 == box && gpi_c2 == floor)
57 {
58     gpi_py += 1;
59     this->gpi_yPlayer = gpi_py;
60     this->gpi_map[gpi_px][gpi_py - 1] = finish;
61     this->gpi_map[gpi_px][gpi_py] = player;
62     this->gpi_map[gpi_px][gpi_py + 1] = box;
63     return;
64 }
65
66 // player -> finBox -> floor
67 // floor -> finPlayer -> box
68 if (gpi_c0 == player && gpi_c1 == finBox && gpi_c2 == floor)
69 {
70     gpi_py += 1;
71     this->gpi_yPlayer = gpi_py;
72     this->gpi_map[gpi_px][gpi_py - 1] = floor;
73     this->gpi_map[gpi_px][gpi_py] = finPlayer;
74     this->gpi_map[gpi_px][gpi_py + 1] = box;
75     return;
76 }
77
78 // finPlayer -> finBox -> floor
79 // finish -> finPlayer -> box
80 if (gpi_c0 == finPlayer && gpi_c1 == finBox && gpi_c2 == floor)
81 {
82     gpi_py += 1;
83     this->gpi_yPlayer = gpi_py;
84     this->gpi_map[gpi_px][gpi_py - 1] = finish;
85     this->gpi_map[gpi_px][gpi_py] = finPlayer;
86     this->gpi_map[gpi_px][gpi_py + 1] = box;
87     return;
88 }
89
90 // player -> box -> finish
91 // floor -> player -> finBox
92 if (gpi_c0 == player && gpi_c1 == box && gpi_c2 == finish)
93 {
94     gpi_py += 1;
95     this->gpi_yPlayer = gpi_py;
96     this->gpi_map[gpi_px][gpi_py - 1] = floor;
97     this->gpi_map[gpi_px][gpi_py] = player;
98     this->gpi_map[gpi_px][gpi_py + 1] = finBox;
99     return;
100 }
101
102 // finPlayer -> box -> finish
103 // finish -> player -> finBox
104 if (gpi_c0 == finPlayer && gpi_c1 == box && gpi_c2 == finish)
105 {

```

```

106         gpi_py += 1;
107         this->gpi_yPlayer = gpi_py;
108         this->gpi_map[gpi_px][gpi_py - 1] = finish;
109         this->gpi_map[gpi_px][gpi_py] = player;
110         this->gpi_map[gpi_px][gpi_py + 1] = finBox;
111         return;
112     }
113
114     // player -> finBox    -> finish
115     // floor  -> finPlayer -> finBox
116     if (gpi_c0 == player && gpi_c1 == finBox && gpi_c2 == finish)
117     {
118         gpi_py += 1;
119         this->gpi_yPlayer = gpi_py;
120         this->gpi_map[gpi_px][gpi_py - 1] = floor;
121         this->gpi_map[gpi_px][gpi_py] = finPlayer;
122         this->gpi_map[gpi_px][gpi_py + 1] = finBox;
123         return;
124     }
125
126     // finPlayer -> finBox    -> finish
127     // finish    -> finPlayer -> finBox
128     if (gpi_c0 == finPlayer && gpi_c1 == finBox && gpi_c2 == finish)
129     {
130         gpi_py += 1;
131         this->gpi_yPlayer = gpi_py;
132         this->gpi_map[gpi_px][gpi_py - 1] = finish;
133         this->gpi_map[gpi_px][gpi_py] = finPlayer;
134         this->gpi_map[gpi_px][gpi_py + 1] = finBox;
135         return;
136     }
137
138     // player -> finish
139     // floor  -> finPlayer
140     if (gpi_c0 == player && gpi_c1 == finish)
141     {
142         gpi_py += 1;
143         this->gpi_yPlayer = gpi_py;
144         this->gpi_map[gpi_px][gpi_py - 1] = floor;
145         this->gpi_map[gpi_px][gpi_py] = finPlayer;
146         return;
147     }
148
149     // finPlayer -> finish
150     // finish    -> finPlayer
151     if (gpi_c0 == finPlayer && gpi_c1 == finish)
152     {
153         gpi_py += 1;
154         this->gpi_yPlayer = gpi_py;
155         this->gpi_map[gpi_px][gpi_py - 1] = finish;
156         this->gpi_map[gpi_px][gpi_py] = finPlayer;
157         return;
158     }
159 }
160 }

```

../gpi_osisp5_option5/gpi_gamewindow__gpi_goLeft.cpp

```

1 #include "gpi_gamewindow.hpp"
2 #include "ui_gpi_gamewindow.h"

```

					ОЛР.ПО4.190333.01 81 00	Лист
Изм	Лист	№ докум.	Подп.	Дата		12

```

3
4 void gpi_GameWindow::gpi_goLeft ()
5 {
6     int          gpi_px          = this->gpi_xPlayer;
7     int          gpi_py          = this->gpi_yPlayer;
8     gpi_MapChar  gpi_c0;
9     gpi_MapChar  gpi_c1;
10    gpi_MapChar  gpi_c2;
11    // = = = = =
12
13    if (gpi_px != 0)
14    {
15        gpi_c0 = this->gpi_map[gpi_px][gpi_py];
16        gpi_c1 = this->gpi_map[gpi_px - 1][gpi_py];
17        gpi_c2 = this->gpi_map[gpi_px - 2][gpi_py];
18
19        // player -> floor
20        // floor -> player
21        if (gpi_c0 == player && gpi_c1 == floor)
22        {
23            gpi_px -= 1;
24            this->gpi_xPlayer = gpi_px;
25            this->gpi_map[gpi_px + 1][gpi_py] = floor;
26            this->gpi_map[gpi_px][gpi_py] = player;
27            return;
28        }
29
30        // finPlayer -> floor
31        // finish -> player
32        if (gpi_c0 == finPlayer && gpi_c1 == floor)
33        {
34            gpi_px -= 1;
35            this->gpi_xPlayer = gpi_px;
36            this->gpi_map[gpi_px + 1][gpi_py] = finish;
37            this->gpi_map[gpi_px][gpi_py] = player;
38            return;
39        }
40
41        // player -> box -> floor
42        // floor -> player -> box
43        if (gpi_c0 == player && gpi_c1 == box && gpi_c2 == floor)
44        {
45            gpi_px -= 1;
46            this->gpi_xPlayer = gpi_px;
47            this->gpi_map[gpi_px + 1][gpi_py] = floor;
48            this->gpi_map[gpi_px][gpi_py] = player;
49            this->gpi_map[gpi_px - 1][gpi_py] = box;
50            return;
51        }
52
53        // finPlayer -> box -> floor
54        // finish -> player -> box
55        if (gpi_c0 == finPlayer && gpi_c1 == box && gpi_c2 == floor)
56        {
57            gpi_px -= 1;
58            this->gpi_xPlayer = gpi_px;
59            this->gpi_map[gpi_px + 1][gpi_py] = finish;
60            this->gpi_map[gpi_px][gpi_py] = player;
61            this->gpi_map[gpi_px - 1][gpi_py] = box;
62            return;

```

```

63     }
64
65     // player -> finBox    -> floor
66     // floor  -> finPlayer -> box
67     if (gpi_c0 == player && gpi_c1 == finBox && gpi_c2 == floor)
68     {
69         gpi_px -= 1;
70         this->gpi_xPlayer = gpi_px;
71         this->gpi_map[gpi_px + 1][gpi_py] = floor;
72         this->gpi_map[gpi_px    ][gpi_py] = finPlayer;
73         this->gpi_map[gpi_px - 1][gpi_py] = box;
74         return;
75     }
76
77     // finPlayer -> finBox    -> floor
78     // finish    -> finPlayer -> box
79     if (gpi_c0 == finPlayer && gpi_c1 == finBox && gpi_c2 == floor)
80     {
81         gpi_px -= 1;
82         this->gpi_xPlayer = gpi_px;
83         this->gpi_map[gpi_px + 1][gpi_py] = finish;
84         this->gpi_map[gpi_px    ][gpi_py] = finPlayer;
85         this->gpi_map[gpi_px - 1][gpi_py] = box;
86         return;
87     }
88
89     // player -> box      -> finish
90     // floor  -> player  -> finBox
91     if (gpi_c0 == player && gpi_c1 == box && gpi_c2 == finish)
92     {
93         gpi_px -= 1;
94         this->gpi_xPlayer = gpi_px;
95         this->gpi_map[gpi_px + 1][gpi_py] = floor;
96         this->gpi_map[gpi_px    ][gpi_py] = player;
97         this->gpi_map[gpi_px - 1][gpi_py] = finBox;
98         return;
99     }
100
101     // finPlayer -> box      -> finish
102     // finish    -> player  -> finBox
103     if (gpi_c0 == finPlayer && gpi_c1 == box && gpi_c2 == finish)
104     {
105         gpi_px -= 1;
106         this->gpi_xPlayer = gpi_px;
107         this->gpi_map[gpi_px + 1][gpi_py] = finish;
108         this->gpi_map[gpi_px    ][gpi_py] = player;
109         this->gpi_map[gpi_px - 1][gpi_py] = finBox;
110         return;
111     }
112
113     // player -> finBox    -> finish
114     // floor  -> finPlayer -> finBox
115     if (gpi_c0 == player && gpi_c1 == finBox && gpi_c2 == finish)
116     {
117         gpi_px -= 1;
118         this->gpi_xPlayer = gpi_px;
119         this->gpi_map[gpi_px + 1][gpi_py] = floor;
120         this->gpi_map[gpi_px    ][gpi_py] = finPlayer;
121         this->gpi_map[gpi_px - 1][gpi_py] = finBox;
122         return;

```

Изм	Лист	№ докум.	Подп.	Дата

ОЛР.ПО4.190333.01 81 00

Лист

14

```

123     }
124
125     // finPlayer -> finBox    -> finish
126     // finish    -> finPlayer -> finBox
127     if (gpi_c0 == finPlayer && gpi_c1 == finBox && gpi_c2 == finish)
128     {
129         gpi_px -= 1;
130         this->gpi_xPlayer = gpi_px;
131         this->gpi_map[gpi_px + 1][gpi_py] = finish;
132         this->gpi_map[gpi_px    ][gpi_py] = finPlayer;
133         this->gpi_map[gpi_px - 1][gpi_py] = finBox;
134         return;
135     }
136
137     // player -> finish
138     // floor  -> finPlayer
139     if (gpi_c0 == player && gpi_c1 == finish)
140     {
141         gpi_px -= 1;
142         this->gpi_xPlayer = gpi_px;
143         this->gpi_map[gpi_px + 1][gpi_py] = floor;
144         this->gpi_map[gpi_px    ][gpi_py] = finPlayer;
145         return;
146     }
147
148     // finPlayer -> finish
149     // finish    -> finPlayer
150     if (gpi_c0 == finPlayer && gpi_c1 == finish)
151     {
152         gpi_px -= 1;
153         this->gpi_xPlayer = gpi_px;
154         this->gpi_map[gpi_px + 1][gpi_py] = finish;
155         this->gpi_map[gpi_px    ][gpi_py] = finPlayer;
156         return;
157     }
158 }
159 }

```

../gpi_osisp5_option5/gpi_gamewindow__gpi_goRight.cpp

```

1  #include "gpi_gamewindow.h"
2  #include "ui_gpi_gamewindow.h"
3
4  void gpi_GameWindow::gpi_goRight ()
5  {
6      int          gpi_l          = this->gpi_length;
7      int          gpi_px         = this->gpi_xPlayer;
8      int          gpi_py         = this->gpi_yPlayer;
9      gpi_MapChar gpi_c0;
10     gpi_MapChar gpi_c1;
11     gpi_MapChar gpi_c2;
12     // =====
13
14     if (gpi_px != gpi_l - 1)
15     {
16         gpi_c0 = this->gpi_map[gpi_px    ][gpi_py];
17         gpi_c1 = this->gpi_map[gpi_px + 1][gpi_py];
18         gpi_c2 = this->gpi_map[gpi_px + 2][gpi_py];
19
20         // player -> floor

```

```

21 // floor -> player
22 if (gpi_c0 == player && gpi_c1 == floor)
23 {
24     gpi_px += 1;
25     this->gpi_xPlayer = gpi_px;
26     this->gpi_map[gpi_px - 1][gpi_py] = floor;
27     this->gpi_map[gpi_px][gpi_py] = player;
28     return;
29 }
30
31 // finPlayer -> floor
32 // finish -> player
33 if (gpi_c0 == finPlayer && gpi_c1 == floor)
34 {
35     gpi_px += 1;
36     this->gpi_xPlayer = gpi_px;
37     this->gpi_map[gpi_px - 1][gpi_py] = finish;
38     this->gpi_map[gpi_px][gpi_py] = player;
39     return;
40 }
41
42 // player -> box -> floor
43 // floor -> player -> box
44 if (gpi_c0 == player && gpi_c1 == box && gpi_c2 == floor)
45 {
46     gpi_px += 1;
47     this->gpi_xPlayer = gpi_px;
48     this->gpi_map[gpi_px - 1][gpi_py] = floor;
49     this->gpi_map[gpi_px][gpi_py] = player;
50     this->gpi_map[gpi_px + 1][gpi_py] = box;
51     return;
52 }
53
54 // finPlayer -> box -> floor
55 // finish -> player -> box
56 if (gpi_c0 == finPlayer && gpi_c1 == box && gpi_c2 == floor)
57 {
58     gpi_px += 1;
59     this->gpi_xPlayer = gpi_px;
60     this->gpi_map[gpi_px - 1][gpi_py] = finish;
61     this->gpi_map[gpi_px][gpi_py] = player;
62     this->gpi_map[gpi_px + 1][gpi_py] = box;
63     return;
64 }
65
66 // finPlayer -> finBox -> floor
67 // finish -> finPlayer -> box
68 if (gpi_c0 == finPlayer && gpi_c1 == finBox && gpi_c2 == floor)
69 {
70     gpi_px += 1;
71     this->gpi_xPlayer = gpi_px;
72     this->gpi_map[gpi_px - 1][gpi_py] = finish;
73     this->gpi_map[gpi_px][gpi_py] = finPlayer;
74     this->gpi_map[gpi_px + 1][gpi_py] = box;
75     return;
76 }
77
78 // player -> finBox -> floor
79 // floor -> finPlayer -> box
80 if (gpi_c0 == player && gpi_c1 == finBox && gpi_c2 == floor)

```



```

81     {
82         gpi_px += 1;
83         this->gpi_xPlayer = gpi_px;
84         this->gpi_map[gpi_px - 1][gpi_py] = floor;
85         this->gpi_map[gpi_px][gpi_py] = finPlayer;
86         this->gpi_map[gpi_px + 1][gpi_py] = box;
87         return;
88     }
89
90     // player -> box -> finish
91     // floor -> player -> finBox
92     if (gpi_c0 == player && gpi_c1 == box && gpi_c2 == finish)
93     {
94         gpi_px += 1;
95         this->gpi_xPlayer = gpi_px;
96         this->gpi_map[gpi_px - 1][gpi_py] = floor;
97         this->gpi_map[gpi_px][gpi_py] = player;
98         this->gpi_map[gpi_px + 1][gpi_py] = finBox;
99         return;
100     }
101
102     // finPlayer -> box -> finish
103     // finish -> player -> finBox
104     if (gpi_c0 == finPlayer && gpi_c1 == box && gpi_c2 == finish)
105     {
106         gpi_px += 1;
107         this->gpi_xPlayer = gpi_px;
108         this->gpi_map[gpi_px - 1][gpi_py] = finish;
109         this->gpi_map[gpi_px][gpi_py] = player;
110         this->gpi_map[gpi_px + 1][gpi_py] = finBox;
111         return;
112     }
113
114     // player -> finBox -> finish
115     // floor -> finPlayer -> finBox
116     if (gpi_c0 == player && gpi_c1 == finBox && gpi_c2 == finish)
117     {
118         gpi_px += 1;
119         this->gpi_xPlayer = gpi_px;
120         this->gpi_map[gpi_px - 1][gpi_py] = floor;
121         this->gpi_map[gpi_px][gpi_py] = finPlayer;
122         this->gpi_map[gpi_px + 1][gpi_py] = finBox;
123         return;
124     }
125
126     // finPlayer -> finBox -> finish
127     // finish -> finPlayer -> finBox
128     if (gpi_c0 == finPlayer && gpi_c1 == finBox && gpi_c2 == finish)
129     {
130         gpi_px += 1;
131         this->gpi_xPlayer = gpi_px;
132         this->gpi_map[gpi_px - 1][gpi_py] = finish;
133         this->gpi_map[gpi_px][gpi_py] = finPlayer;
134         this->gpi_map[gpi_px + 1][gpi_py] = finBox;
135         return;
136     }
137
138     // player -> finish
139     // floor -> finPlayer
140     if (gpi_c0 == player && gpi_c1 == finish)

```

```

141     {
142         gpi_px += 1;
143         this->gpi_xPlayer = gpi_px;
144         this->gpi_map[gpi_px - 1][gpi_py] = floor;
145         this->gpi_map[gpi_px][gpi_py] = finPlayer;
146         return;
147     }
148
149     // finPlayer -> finish
150     // finish -> finPlayer
151     if (gpi_c0 == finPlayer && gpi_c1 == finish)
152     {
153         gpi_px += 1;
154         this->gpi_xPlayer = gpi_px;
155         this->gpi_map[gpi_px - 1][gpi_py] = finish;
156         this->gpi_map[gpi_px][gpi_py] = finPlayer;
157         return;
158     }
159 }
160 }

```

../gpi_osisp5_option5/gpi_gamewindow__gpi_goTop.cpp

```

1  #include "gpi_gamewindow.hpp"
2  #include "ui_gpi_gamewindow.h"
3
4  void gpi_GameWindow::gpi_goTop ()
5  {
6      int gpi_px = this->gpi_xPlayer;
7      int gpi_py = this->gpi_yPlayer;
8      gpi_MapChar gpi_c0;
9      gpi_MapChar gpi_c1;
10     gpi_MapChar gpi_c2;
11     // =====
12
13     if (gpi_py != 0)
14     {
15         gpi_c0 = this->gpi_map[gpi_px][gpi_py];
16         gpi_c1 = this->gpi_map[gpi_px][gpi_py - 1];
17         gpi_c2 = this->gpi_map[gpi_px][gpi_py - 2];
18
19         // player -> floor
20         // floor -> player
21         if (gpi_c0 == player && gpi_c1 == floor)
22         {
23             gpi_py -= 1;
24             this->gpi_yPlayer = gpi_py;
25             this->gpi_map[gpi_px][gpi_py + 1] = floor;
26             this->gpi_map[gpi_px][gpi_py] = player;
27             return;
28         }
29
30         // finPlayer -> floor
31         // finish -> player
32         if (gpi_c0 == finPlayer && gpi_c1 == floor)
33         {
34             gpi_py -= 1;
35             this->gpi_yPlayer = gpi_py;
36             this->gpi_map[gpi_px][gpi_py + 1] = finish;
37             this->gpi_map[gpi_px][gpi_py] = player;

```

```

38         return;
39     }
40
41     // player -> box    -> floor
42     // floor  -> player -> box
43     if (gpi_c0 == player && gpi_c1 == box && gpi_c2 == floor)
44     {
45         gpi_py -= 1;
46         this->gpi_yPlayer = gpi_py;
47         this->gpi_map[gpi_px][gpi_py + 1] = floor;
48         this->gpi_map[gpi_px][gpi_py    ] = player;
49         this->gpi_map[gpi_px][gpi_py - 1] = box;
50         return;
51     }
52
53     // finPlayer -> box    -> floor
54     // finish   -> player -> box
55     if (gpi_c0 == finPlayer && gpi_c1 == box && gpi_c2 == floor)
56     {
57         gpi_py -= 1;
58         this->gpi_yPlayer = gpi_py;
59         this->gpi_map[gpi_px][gpi_py + 1] = finish;
60         this->gpi_map[gpi_px][gpi_py    ] = player;
61         this->gpi_map[gpi_px][gpi_py - 1] = box;
62         return;
63     }
64
65     // player -> box    -> finish
66     // floor  -> player -> finBox
67     if (gpi_c0 == player && gpi_c1 == box && gpi_c2 == finish)
68     {
69         gpi_py -= 1;
70         this->gpi_yPlayer = gpi_py;
71         this->gpi_map[gpi_px][gpi_py + 1] = floor;
72         this->gpi_map[gpi_px][gpi_py    ] = player;
73         this->gpi_map[gpi_px][gpi_py - 1] = finBox;
74         return;
75     }
76
77     // finPlayer -> box    -> finish
78     // finish   -> player -> finBox
79     if (gpi_c0 == finPlayer && gpi_c1 == box && gpi_c2 == finish)
80     {
81         gpi_py -= 1;
82         this->gpi_yPlayer = gpi_py;
83         this->gpi_map[gpi_px][gpi_py + 1] = finish;
84         this->gpi_map[gpi_px][gpi_py    ] = player;
85         this->gpi_map[gpi_px][gpi_py - 1] = finBox;
86         return;
87     }
88
89     // player -> finBox    -> floor
90     // floor  -> finPlayer -> box
91     if (gpi_c0 == player && gpi_c1 == finBox && gpi_c2 == floor)
92     {
93         gpi_py -= 1;
94         this->gpi_yPlayer = gpi_py;
95         this->gpi_map[gpi_px][gpi_py + 1] = floor;
96         this->gpi_map[gpi_px][gpi_py    ] = finPlayer;
97         this->gpi_map[gpi_px][gpi_py - 1] = box;

```

```

98         return;
99     }
100
101     // finPlayer -> finBox    -> floor
102     // finish    -> finPlayer -> box
103     if (gpi_c0 == finPlayer && gpi_c1 == finBox && gpi_c2 == floor)
104     {
105         gpi_py -= 1;
106         this->gpi_yPlayer = gpi_py;
107         this->gpi_map[gpi_px][gpi_py + 1] = finish;
108         this->gpi_map[gpi_px][gpi_py    ] = finPlayer;
109         this->gpi_map[gpi_px][gpi_py - 1] = box;
110         return;
111     }
112
113     // player -> finBox    -> finish
114     // floor  -> finPlayer -> finBox
115     if (gpi_c0 == player && gpi_c1 == finBox && gpi_c2 == finish)
116     {
117         gpi_py -= 1;
118         this->gpi_yPlayer = gpi_py;
119         this->gpi_map[gpi_px][gpi_py + 1] = floor;
120         this->gpi_map[gpi_px][gpi_py    ] = finPlayer;
121         this->gpi_map[gpi_px][gpi_py - 1] = finBox;
122         return;
123     }
124
125     // finPlayer -> finBox    -> finish
126     // finish    -> finPlayer -> finBox
127     if (gpi_c0 == finPlayer && gpi_c1 == finBox && gpi_c2 == finish)
128     {
129         gpi_py -= 1;
130         this->gpi_yPlayer = gpi_py;
131         this->gpi_map[gpi_px][gpi_py + 1] = finish;
132         this->gpi_map[gpi_px][gpi_py    ] = finPlayer;
133         this->gpi_map[gpi_px][gpi_py - 1] = finBox;
134         return;
135     }
136
137     // player -> finish
138     // floor  -> finPlayer
139     if (gpi_c0 == player && gpi_c1 == finish)
140     {
141         gpi_py -= 1;
142         this->gpi_yPlayer = gpi_py;
143         this->gpi_map[gpi_px][gpi_py + 1] = floor;
144         this->gpi_map[gpi_px][gpi_py    ] = finPlayer;
145         return;
146     }
147
148     // finPlayer -> finish
149     // finish    -> finPlayer
150     if (gpi_c0 == finPlayer && gpi_c1 == finish)
151     {
152         gpi_py -= 1;
153         this->gpi_yPlayer = gpi_py;
154         this->gpi_map[gpi_px][gpi_py + 1] = finish;
155         this->gpi_map[gpi_px][gpi_py    ] = finPlayer;
156         return;
157     }

```

```

158     }
159 }

    ../gpi_osisp5_option5/gpi_gamewindow__gpi_sayWon.cpp

1  #include "gpi_gamewindow.hpp"
2  #include "ui_gpi_gamewindow.h"
3
4  void gpi_GameWindow::gpi_sayWon ()
5  {
6      const int    gpi_w          = this->gpi_WinWidth;
7      const int    gpi_l          = this->gpi_length;
8      int          gpi_i;
9      int          gpi_j;
10     QMessageBox  gpi_msgBox;
11     // =====
12
13     gpi_i = 0;
14     while (gpi_i < gpi_l)
15     {
16         gpi_j = 0;
17         while (gpi_j < gpi_l)
18         {
19             if (
20                 this->gpi_map[gpi_i][gpi_j] == finish
21                 || this->gpi_map[gpi_i][gpi_j] == finPlayer
22             )
23             {
24                 return;
25             }
26             ++gpi_j;
27         }
28         ++gpi_i;
29     }
30
31     this->close ();
32     gpi_msgBox.setStyleSheet ("QLabel{min-width: " + QString::number (gpi_w) + "px; }");
33     gpi_msgBox.setText ("You won");
34     gpi_msgBox.setWindowTitle ("You won");
35     gpi_msgBox.exec ();
36 }

```