

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра информационных систем

ОТЧЕТ
по курсовой работе
по дисциплине «Объектно-ориентированное программирование»
Вариант 4

Студенты гр. 3374

Лобачев И.М.
Савенков Д.А.

Преподаватель

Егоров С.С.

Санкт-Петербург
2024

Постановка задачи

В прямоугольной комнате размером $L \times W$ имеется 1 дверь и 2 окна шириной L_d и L_o соответственно. Положение двери и окон в комнате, а также указанные параметры по умолчанию требуется задать во внешнем файле. Есть следующий список мебели: стол, стул, кресло, диван, шкаф (их размеры по проекции сверху являются параметрами и определяются пользователем в интерактивном режиме). Требуется, выбирая последовательно предметы мебели, расставить их в комнате допустимым образом. Возможна их перестановка. Конечное состояние комнаты с мебелью требуется записать в файл, при загрузке которого оно отображается изначально.

Лексема	Элемент объектной модели
Прямоугольная комната	Класс
Окно	Метод комнаты
Дверь	Метод комнаты
Мебель	Класс
Стол	Входит в класс мебели
Стул	Входит в класс мебели
Кресло	Входит в класс мебели
Диван	Входит в класс мебели
Шкаф	Входит в класс мебели

Диаграммы объектной модели Модель сущность-связь

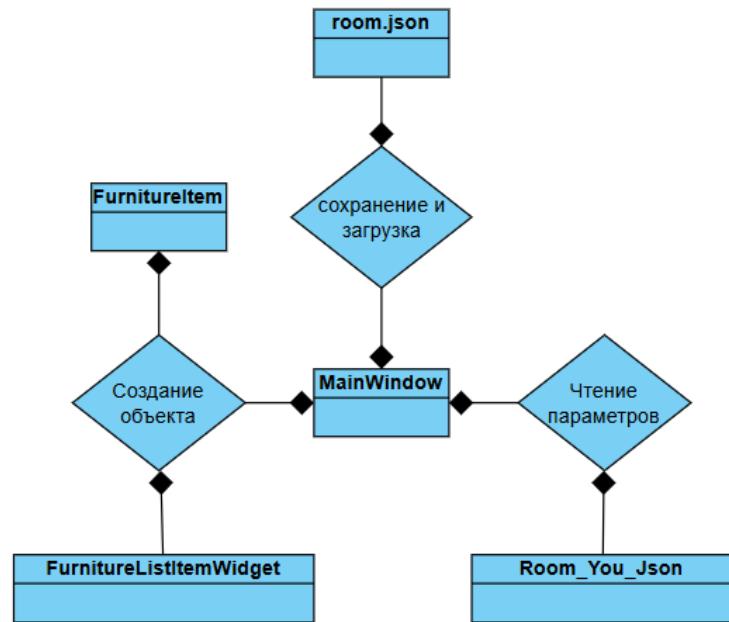
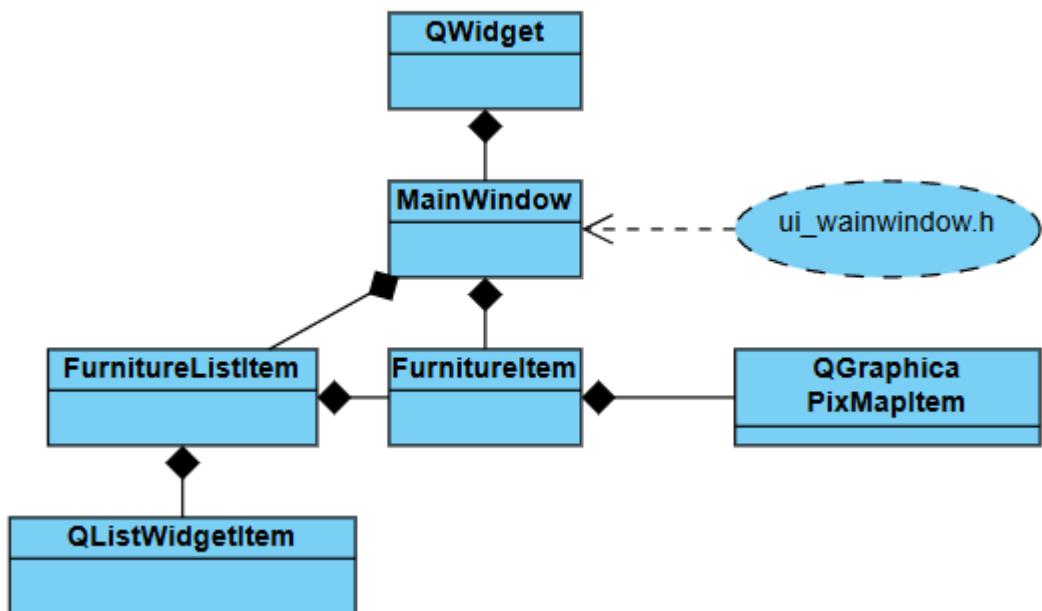


Диаграмма классов



Предметная область	Таблица соответствия Модель «Сущность-связь»	Диаграмма классов
---------------------------	---	--------------------------

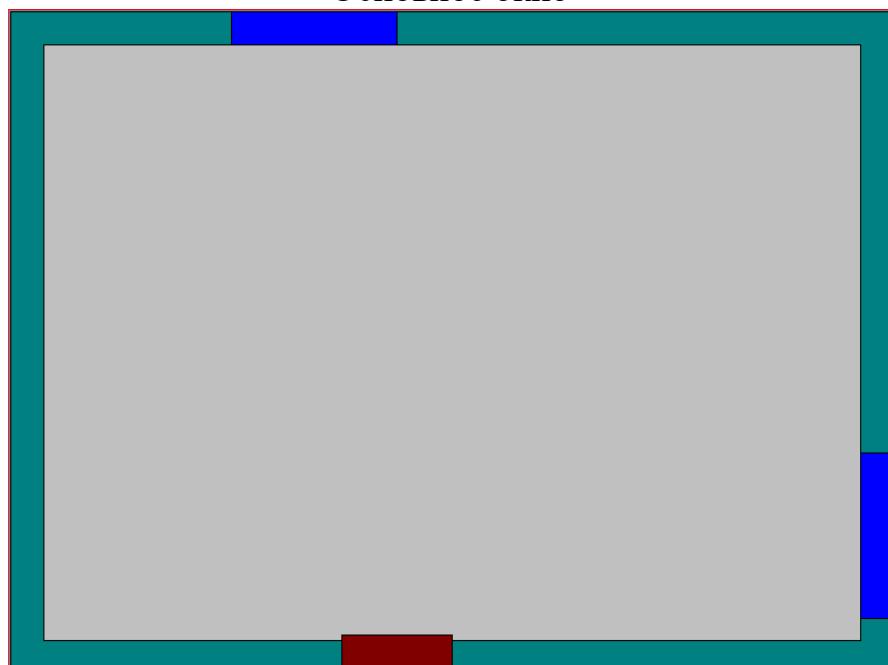
Мебель	Мебель	Furniture
Прямоугольная комната	Прямоугольная комната	MainWindow

**Таблица событий
Обобщенная последовательность действий**

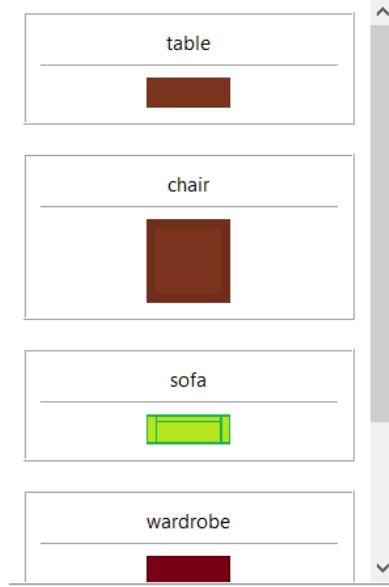
Начальное состояние	Пустая комната
Выбор мебели	В комнате появляется предмет мебели, который можно установить туда, где есть свободное место
Сохранить	Вся установленная мебель в комнате и сама комната сохраняется в формате json
Загрузить интерьер	Загружает сохраненную комнату с расстановкой

Подсистема «Интерфейс»

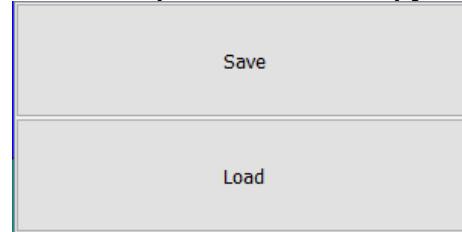
Основное окно



Окно с мебелью



Окно с сохранением и загрузкой



Подсистема «Модель»

Перечень событий, изменяющий состояние модели.

- Запуск программы:
 - Выбор мебели
 - Установка мебели в комнате
 - Кнопка сохранения комнаты в png

Заголовочные файлы классов модели ПрО

1. window.h

```
#ifndef WINDOW_H
#define WINDOW_H
#include "furniture.h"
class Window : public Furniture
{
public:
    Window();
    ~Window();
};
Window::Window()
{
    this->name = "window";
    this->image_path = "";
    this->x = this->y = 0;
    this->width = 150;
    this->height = 50;
    this->rotation = 0;
}
#endif // WINDOW_H
```

2. mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <furnitureitem.h>
#include <QMainWindow>
#include <QGraphicsScene>
#include <QGraphicsPixmapItem>
#include <QListWidgetItem>
#include <QMouseEvent>
#include <QContextMenuEvent>
#include <QJsonArray>
QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE
```

```

class MainWindow : public QMainWindow
{
    Q_OBJECT
public:
    explicit MainWindow(QWidget *parent = nullptr);
    ~MainWindow();
private slots:
    void on_listFurniture_itemClicked(QListWidgetItem *item);
    void saveRoom();
    void loadRoom();
    void on_saveButton_clicked();
    void on_loadButton_clicked();
private:
    Ui::MainWindow *ui;
    QGraphicsScene *scene;
    QList<FurnitureItem *> furnitureItems;
    void loadRoomLayout();
    void setupRoom();
    void createFurnitureList();
    void addFurnitureToScene(const QString &name);
};
#endif // MAINWINDOW_H

```

3. application.h

```

#ifndef APPLICATION_H
#define APPLICATION_H
class Application
{
public:
    Application();
};
#endif // APPLICATION_H

```

4. door.h

```

#ifndef DOOR_H
#define DOOR_H
#include "furniture.h"
class Door : public Furniture
{
public:
    Door();
    ~Door();
};
Door::Door()
{
    this->name = "door";
    this->image_path = "";

```

```

this->x = this->y = 0;
this->width = 100;
this->height = 70;
this->rotation = 0;
}
#endif // DOOR_H

```

5. furniture.h

```

#ifndef FURNITURE_H
#define FURNITURE_H
#include <QString>
class Furniture
{
protected:
    QString name;
    QString image_path;
    int x, y;
    int width, height;
    int rotation;
public:
    Furniture();
    Furniture(const QString name, const QString& image_path, int x, int y,
    int width, int height, int rotation);
    ~Furniture();
    int getX();
    int getY();
    int getWidth();
    int getHeight();
    QString getImagePath();
    int setPosition(int x, int y);
    void resize(int width, int height);
    void rotate(int angle);
};
Furniture::Furniture()
{
    this->name = this->image_path = "";
    this->x = this->y = 0;
    this->width = this->height = 0;
    this->rotation = 0;
}
Furniture::Furniture(const QString name, const QString& image_path, int x,
int y, int width, int height, int rotation)
{
    this->name = name;
    this->image_path = image_path;
    this->x = x;
}

```

```

this->y = y;
this->width = width;
this->height = height;
this->rotation = rotation;
}
int Furniture::getX() { return this->x; }
int Furniture::getY() { return this->y; }
int Furniture::getWidth() { return this->width; }
int Furniture::getHeight() { return this->height; }
QString Furniture::getImagePath() { return this->image_path; }
void Furniture::resize(int width, int height)
{
    this->width = width;
    this->height = height;
}
void Furniture::rotate(int angle)
{
    this->rotation = (this->rotation + angle) % 360;
}
#endif // FURNITURE_H

```

6. furnitureitem.h

```

#ifndef FURNITUREITEM_H
#define FURNITUREITEM_H
#include <QObject>
#include <QGraphicsPixmapItem>
#include <QGraphicsRectItem>
#include <QGraphicsSceneMouseEvent>
#include <QGraphicsSceneContextMenuEvent>
#include <QMenu>
#include <QPen>
class FurnitureItem : public QObject, public QGraphicsPixmapItem
{
    Q_OBJECT
public:
    int csize = 0;
    explicit FurnitureItem(const QPixmap &pixmap);
protected:
    void mousePressEvent(QGraphicsSceneMouseEvent *event) override;
    void mouseMoveEvent(QGraphicsSceneMouseEvent *event) override;
    void mouseReleaseEvent(QGraphicsSceneMouseEvent *event) override;
    void contextMenuEvent(QGraphicsSceneContextMenuEvent *event)
override;
    QPainterPath shape() const override;
private:
    QPointF previousPosition;

```

```
QGraphicsRectItem *collisionOutline;  
  
    void checkCollisions();  
};  
#endif // FURNITUREITEM_H
```

7. furniturelistitemwidget.h

```
#ifndef FURNITURELISTITEMWIDGET_H  
#define FURNITURELISTITEMWIDGET_H  
#include <QWidget>  
#include <QLabel>  
#include <QVBoxLayout>  
class FurnitureListItemWidget : public QWidget  
{  
    Q_OBJECT  
public:  
    explicit FurnitureListItemWidget(const QString &name, const QPixmap  
&image, QWidget *parent = nullptr);  
    QString getName() const;  
private:  
    QLabel *nameLabel;  
    QLabel *imageLabel;  
    QString furnitureName;  
};  
#endif // FURNITURELISTITEMWIDGET_H
```

Скриншоты контрольного примера работы реализованного приложения

При запуске приложения рисуется пустая комната, с дверью и двумя окнами, координаты которых взяты из файла формата json. Слева отображается комната, слева – список из предметов мебели.

При клике в списке на предмет он появляется в середине комнаты, после чего его можно изменять и передвигать. При клике на объект левой кнопкой мыши он становится выбранным, у него появляются пунктирные рамки, его можно передвигать и проводить манипуляции. При клике на предмет левой кнопкой мыши, появляется контекстное меню с командами:

1. Повернуть по часовой стрелке (поворот выполняется на 45°).
2. Повернуть против часовой стрелке (аналогично с 1 п.).
3. Увеличить размер (размер увеличивается на 10%).
4. Уменьшить размер (размер уменьшается на 10%).
5. Вернуть размер.

Заполним интерьер мебелью (рисунки 1 – 15)

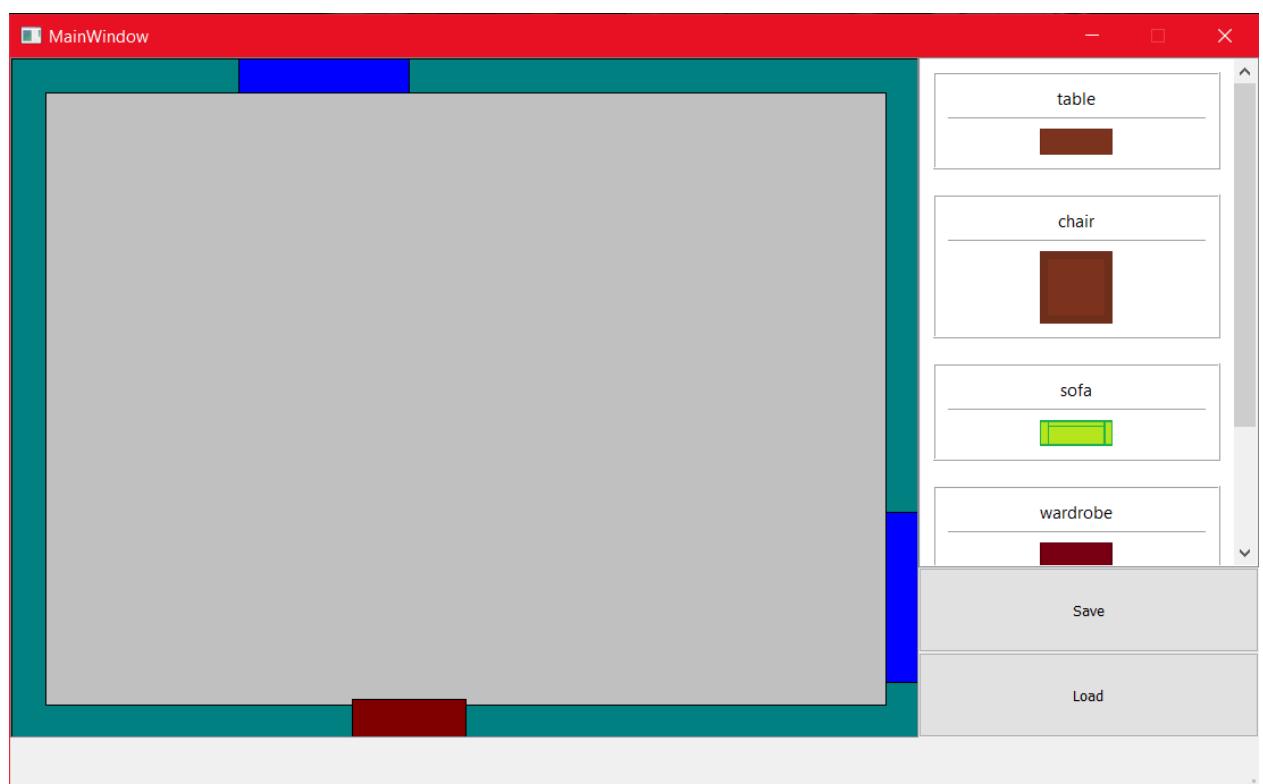


Рисунок 1. Пустая комната при запуске программы

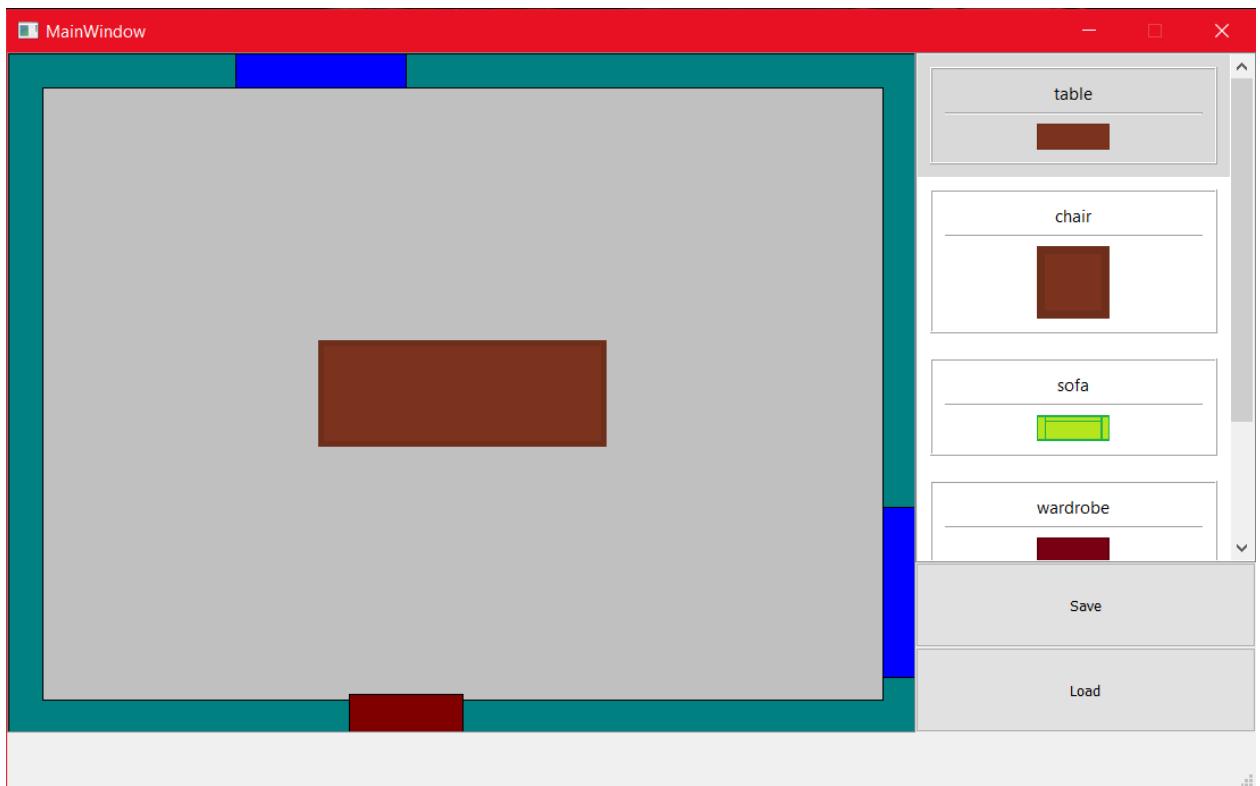


Рисунок 2. Размещение стола(table)

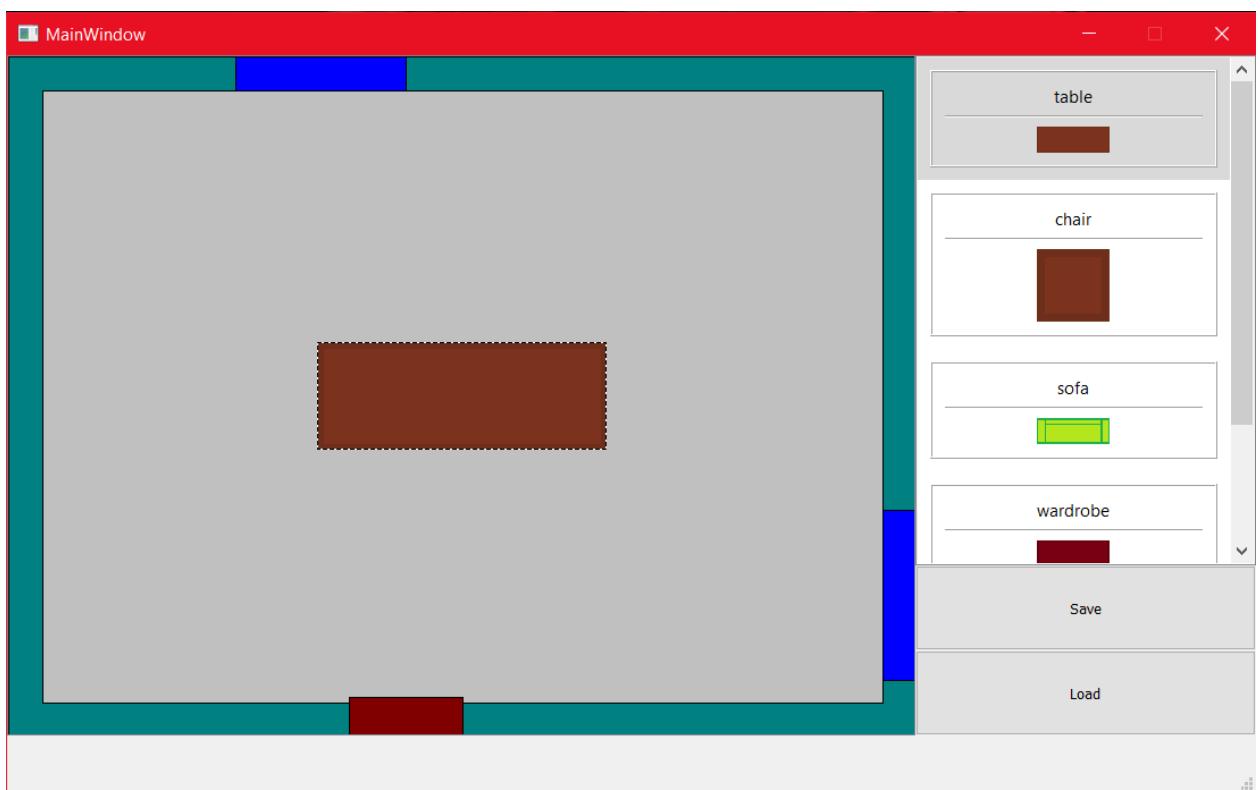


Рисунок 3. Выделение объекта при нажатии по нему

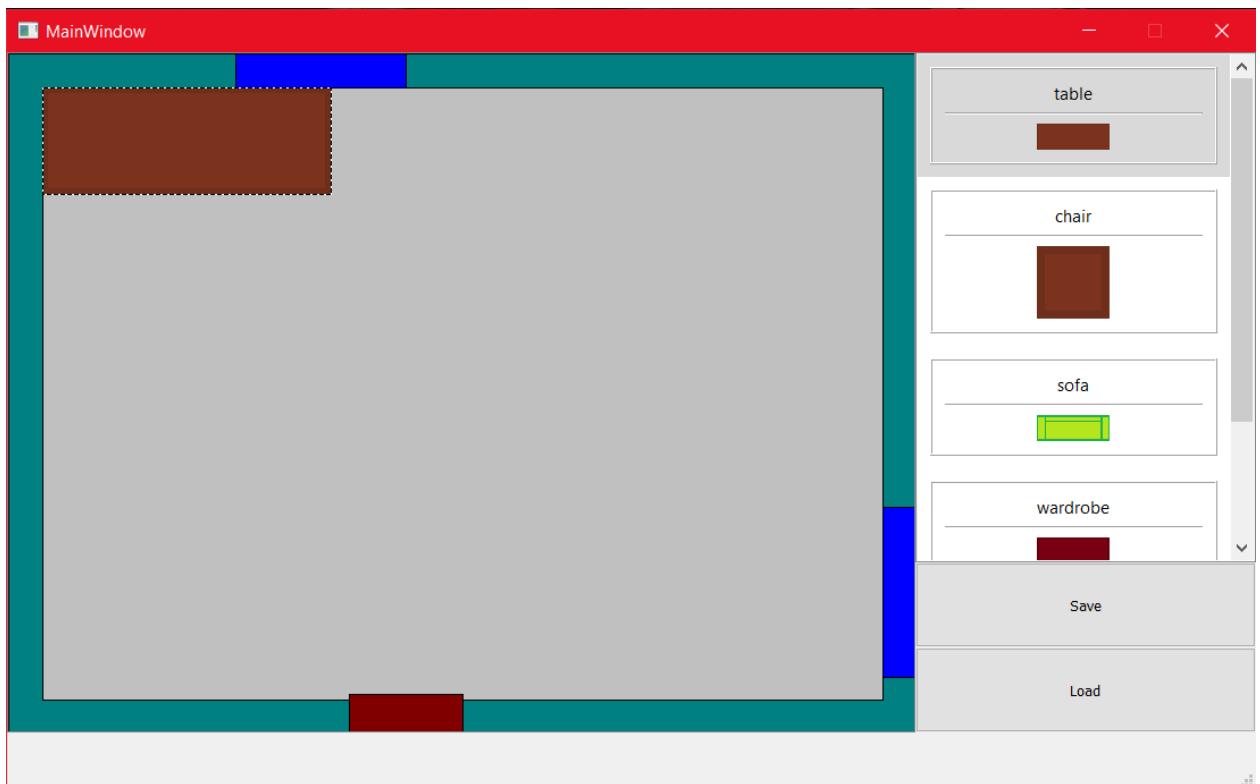


Рисунок 4

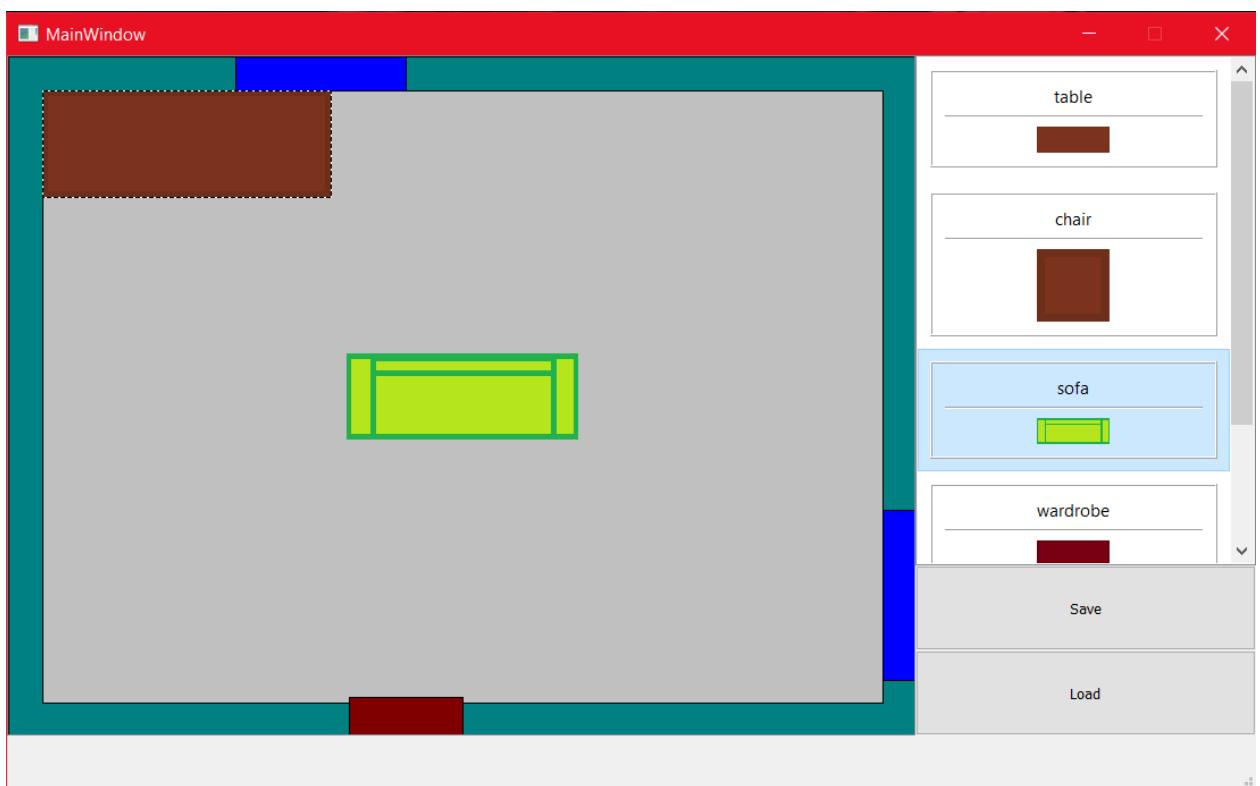


Рисунок 5

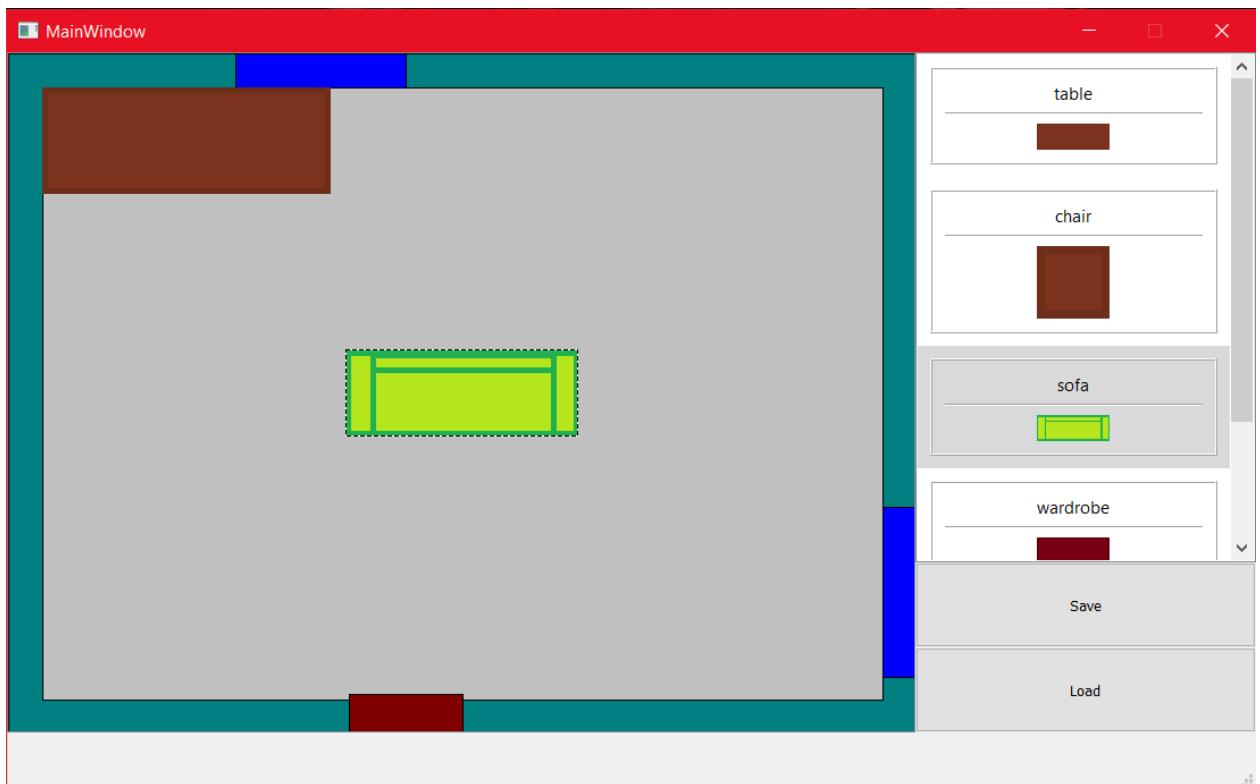


Рисунок 6

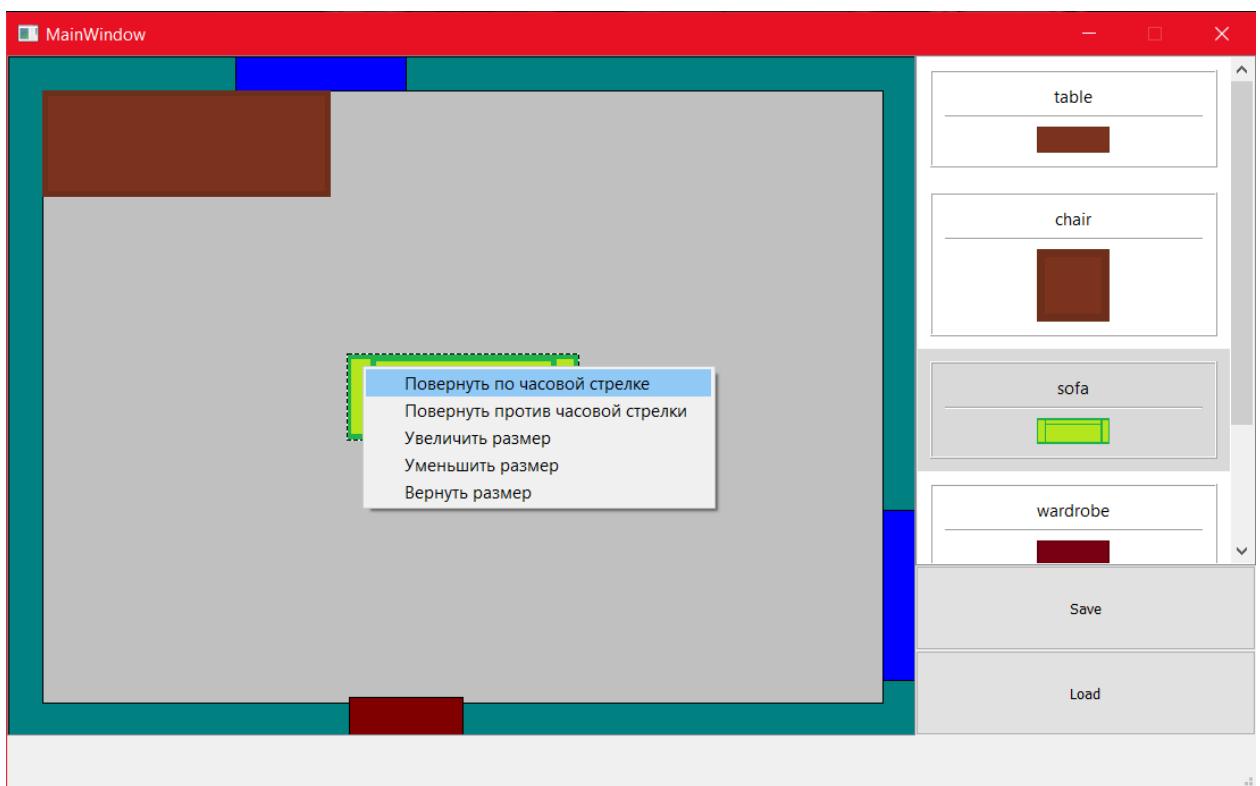


Рисунок 7. Контекстное меню объекта

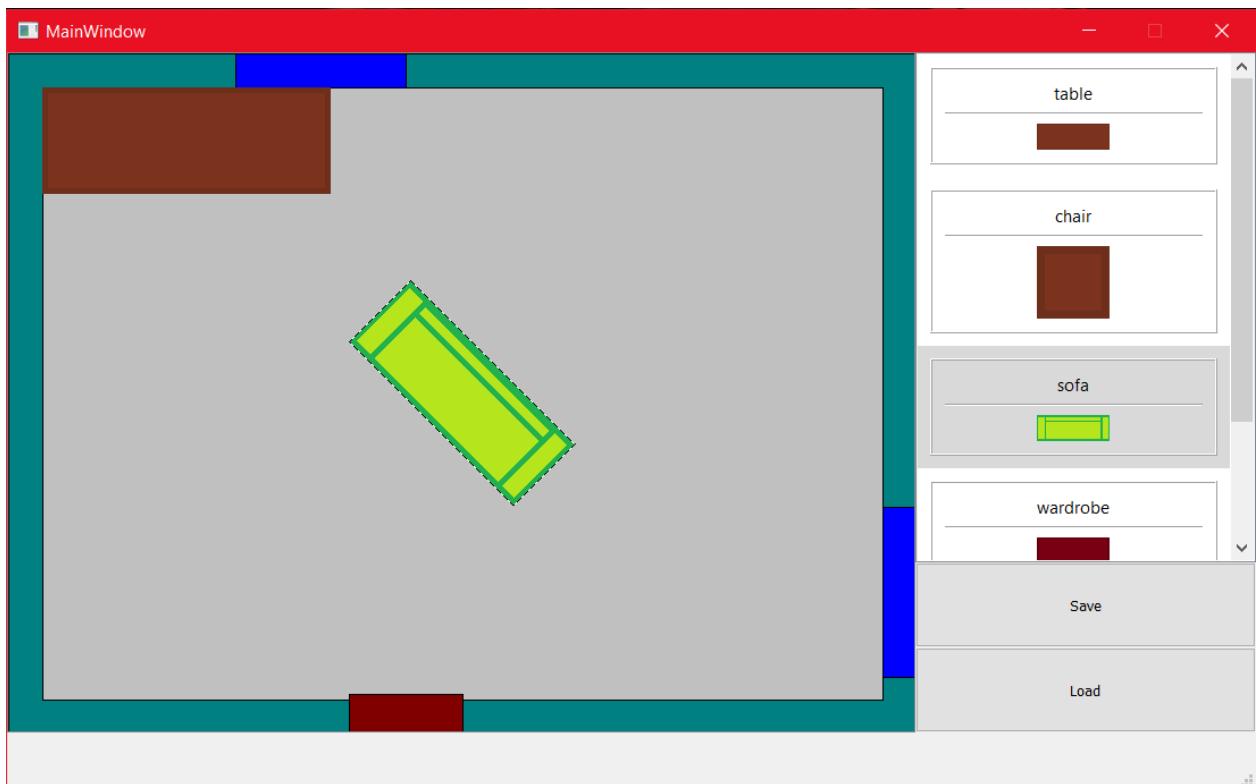


Рисунок 8

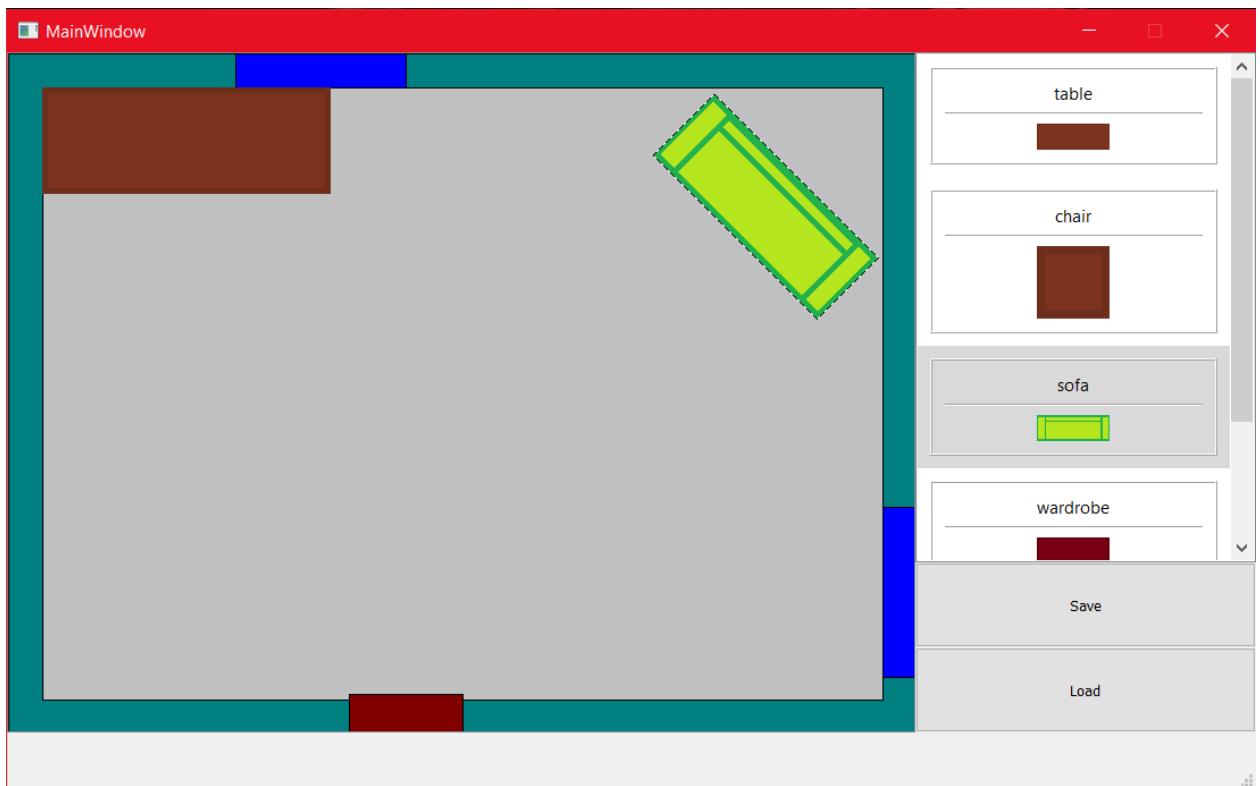


Рисунок 9

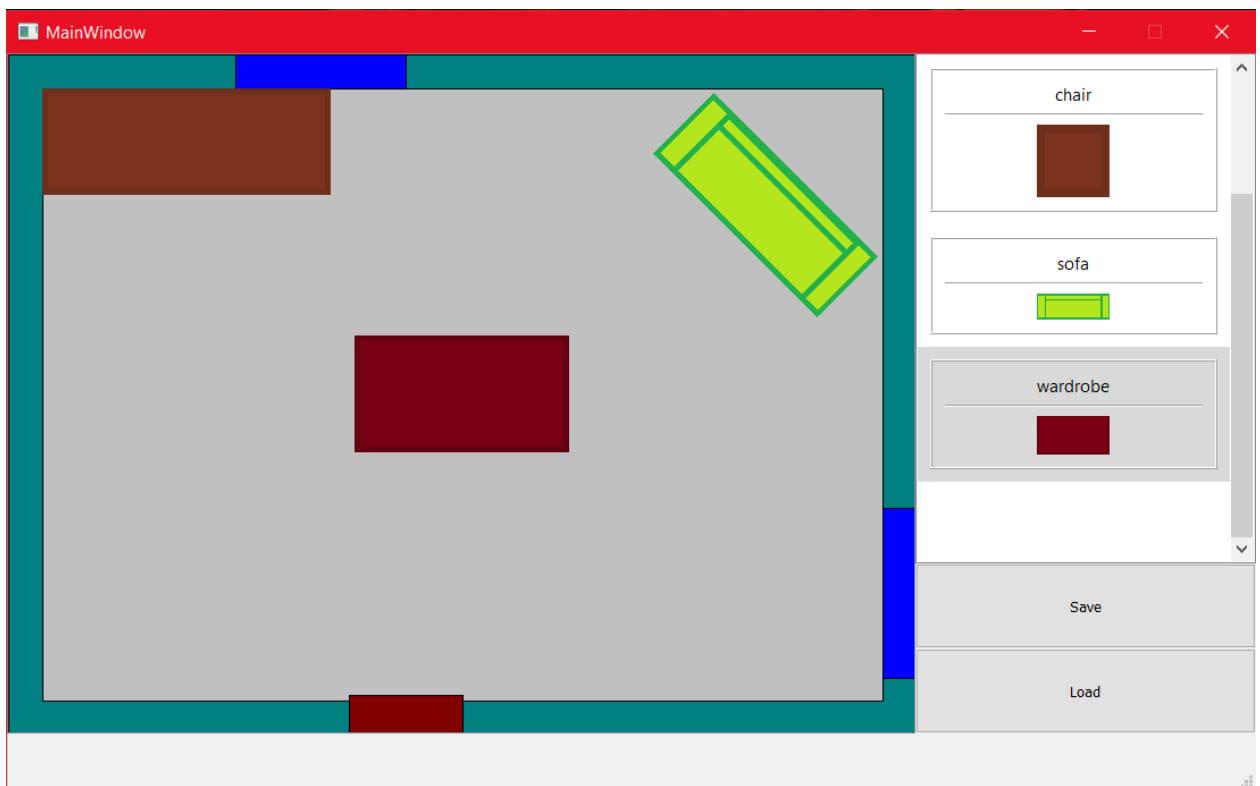


Рисунок 10

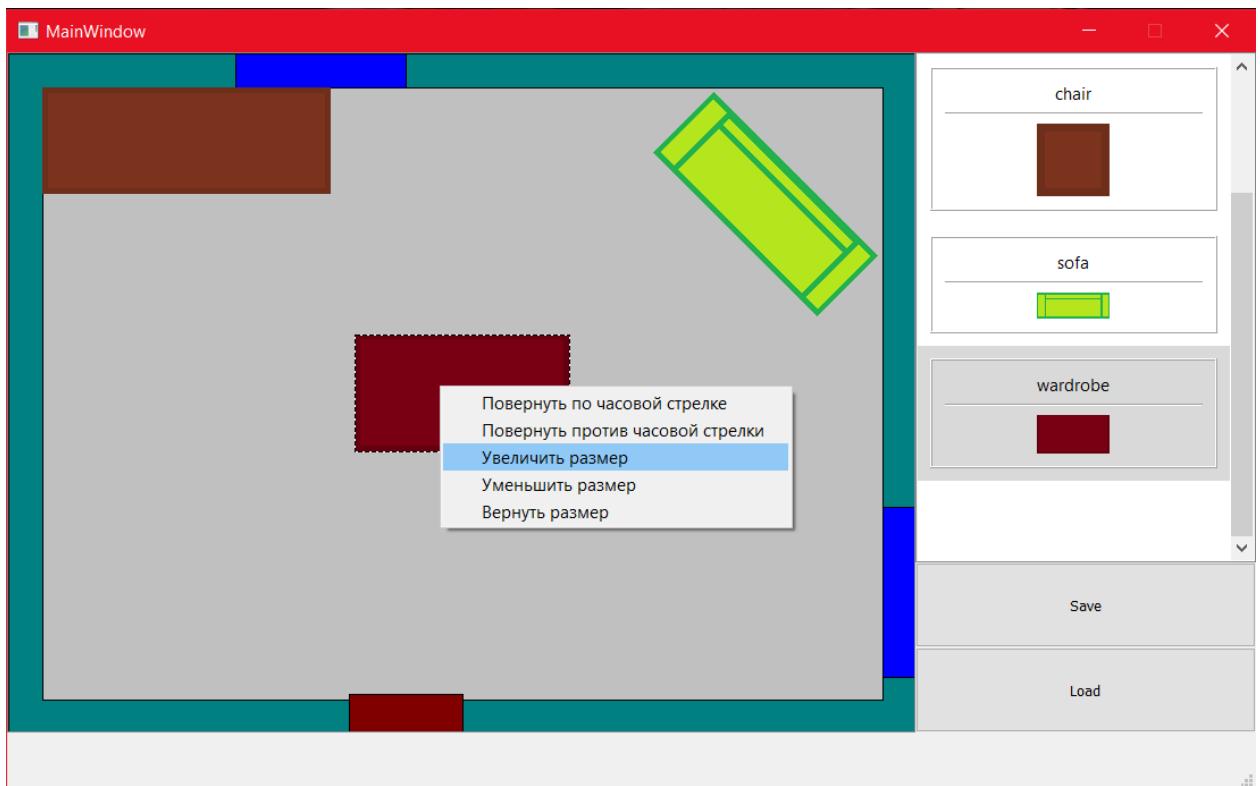


Рисунок 11

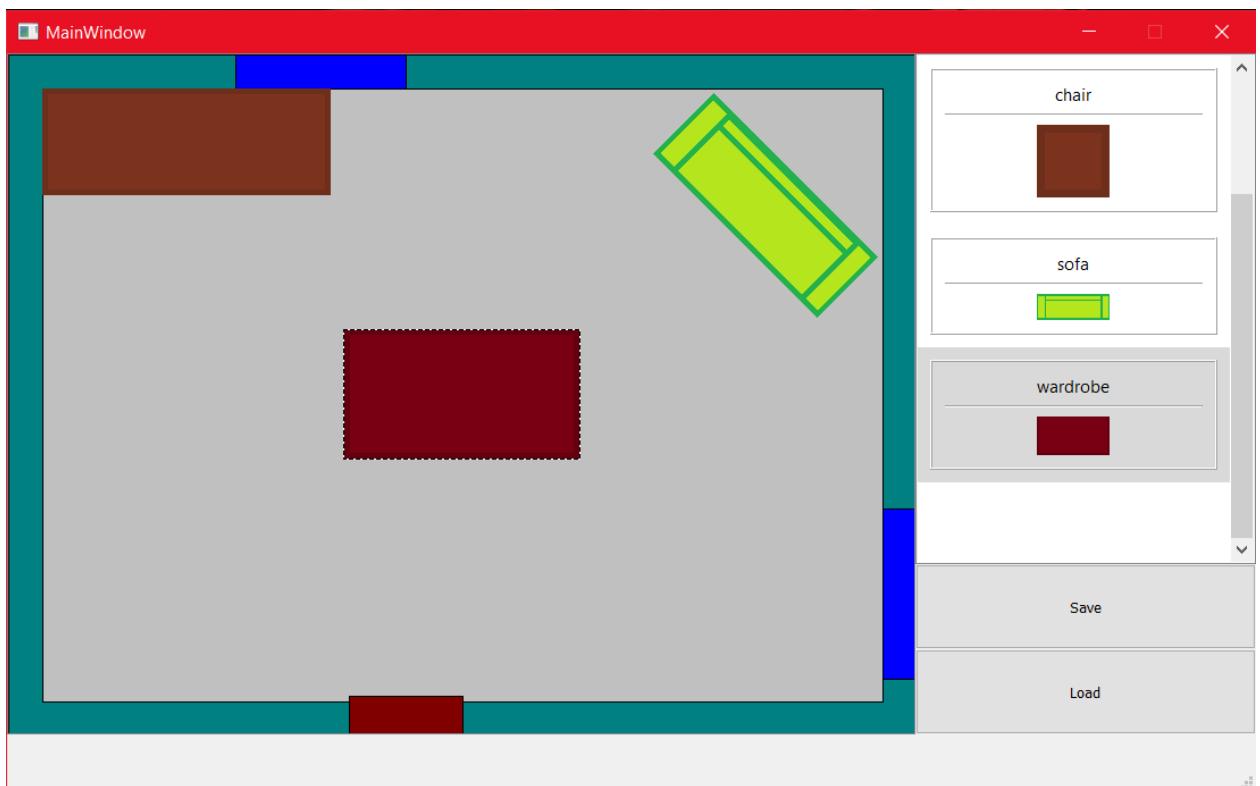


Рисунок 12

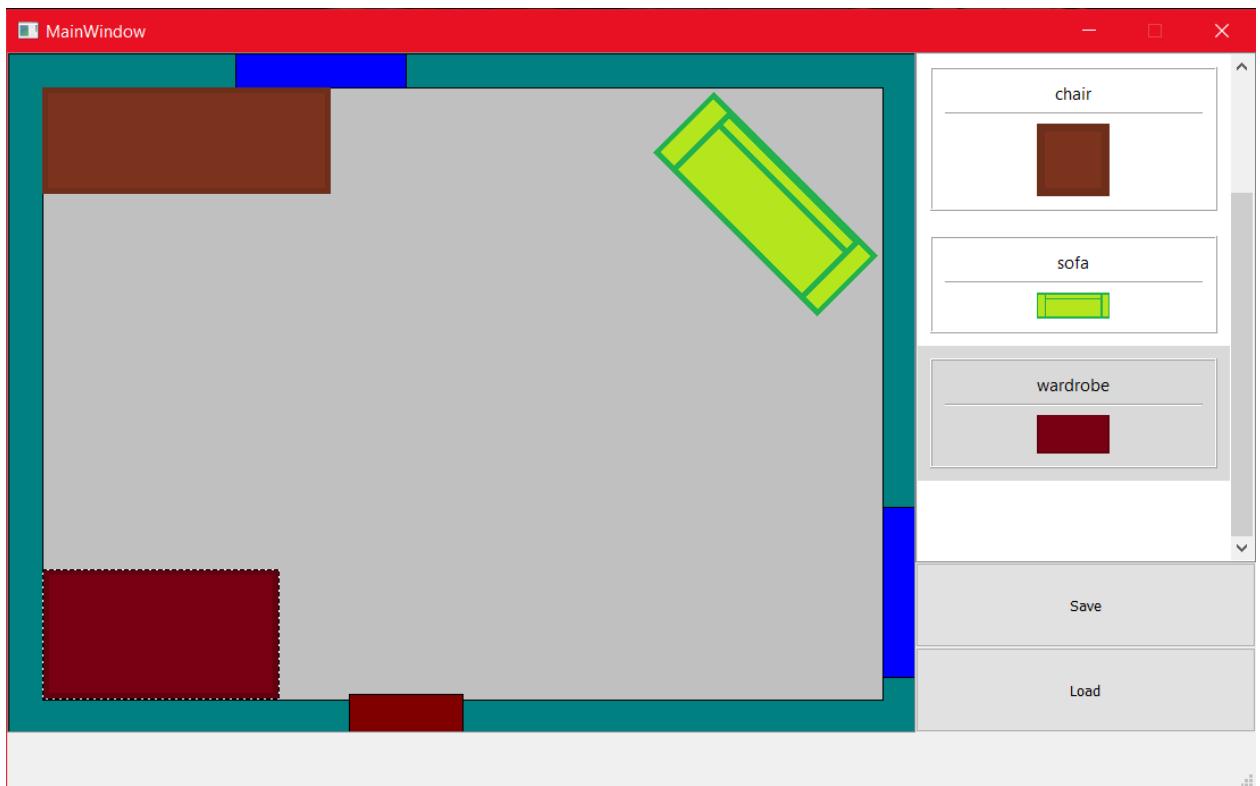


Рисунок 13

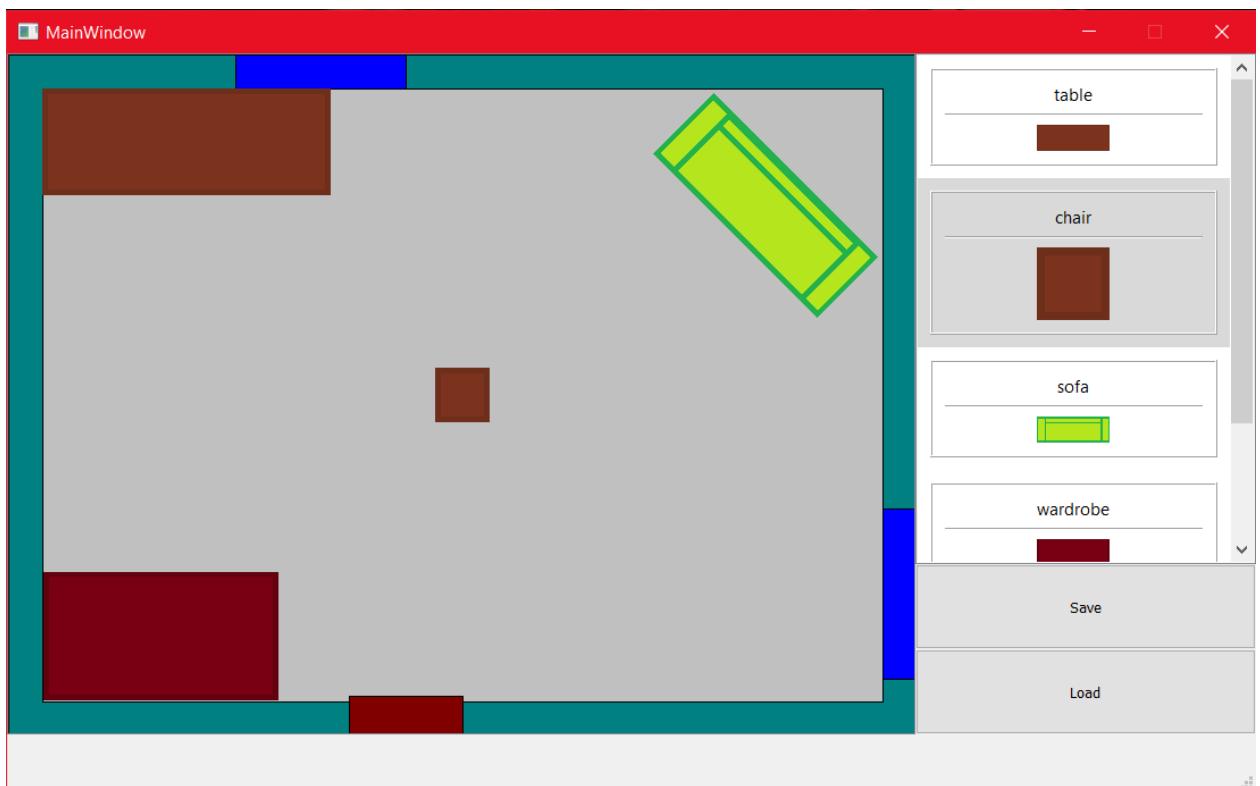


Рисунок 14

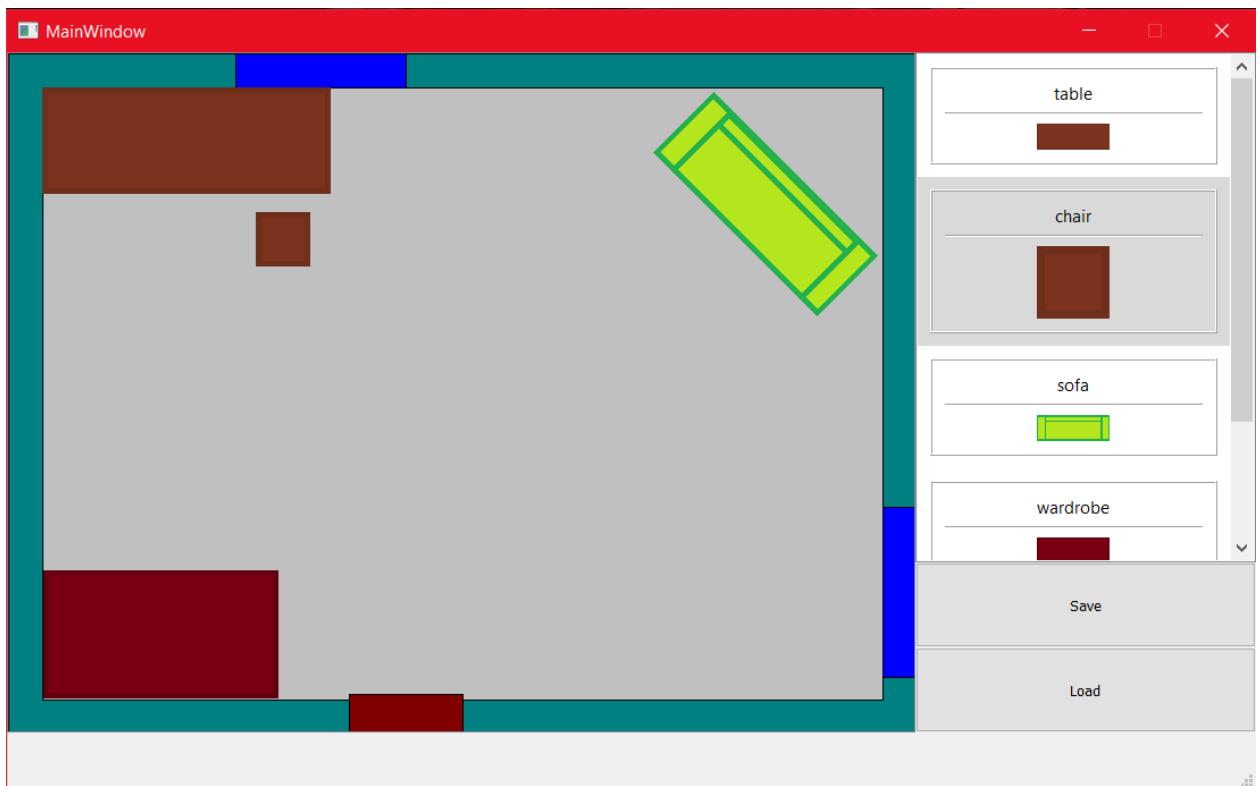


Рисунок 15

Сохраним текущий интерьер, нажав на кнопку Save (рисунок 16). Затем разместим еще один объект. (рисунки 17 – 18).

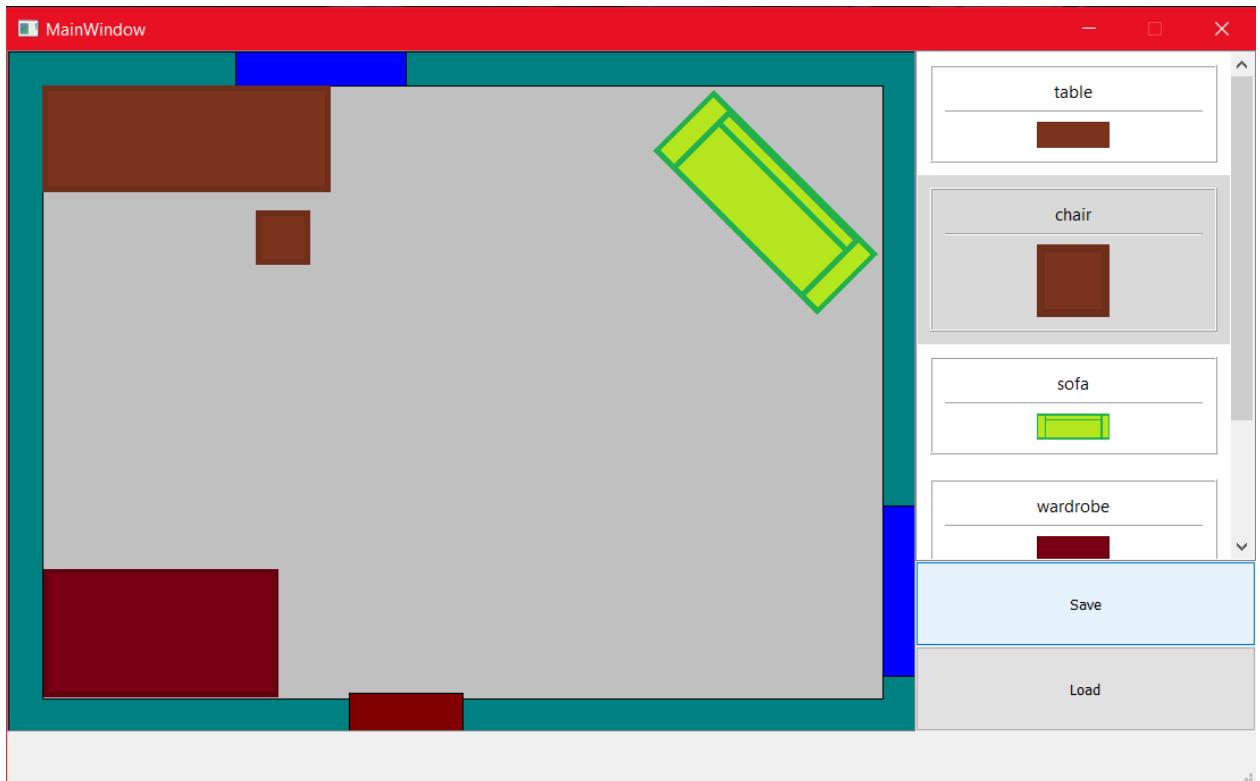


Рисунок 16

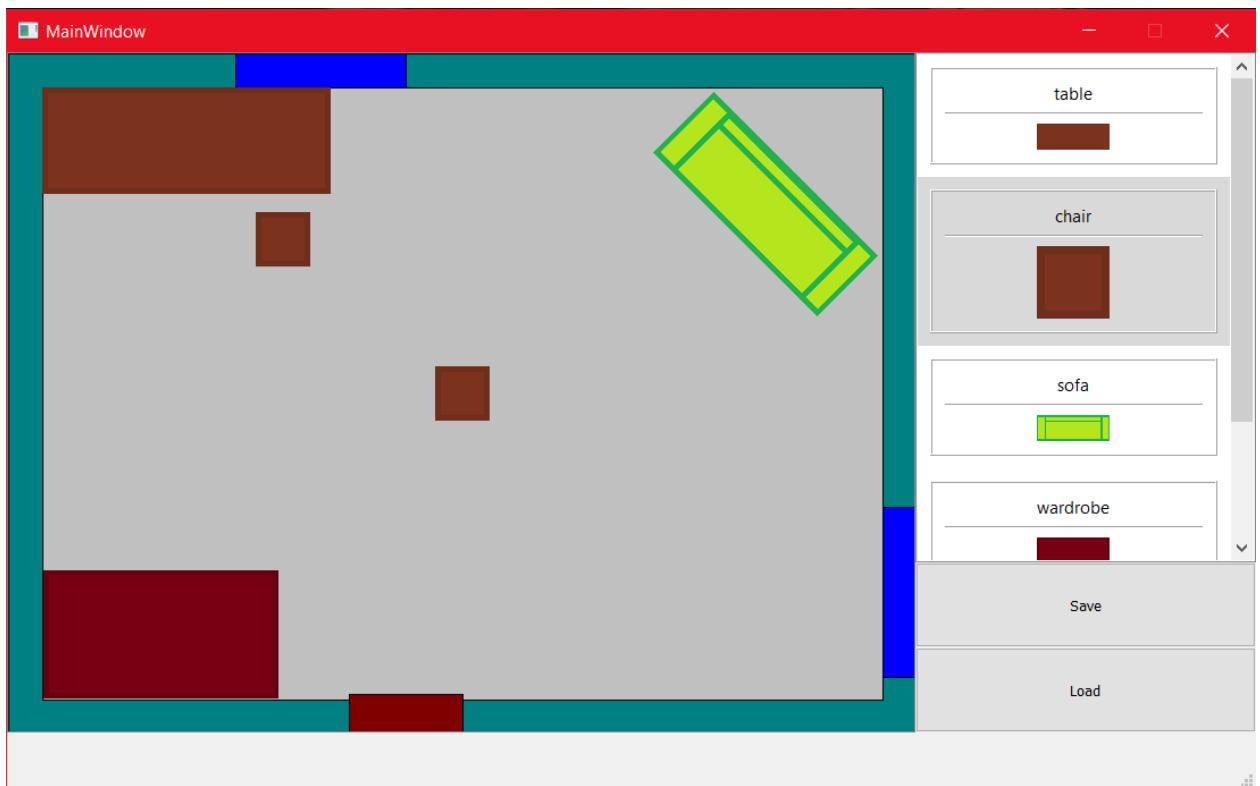


Рисунок 17

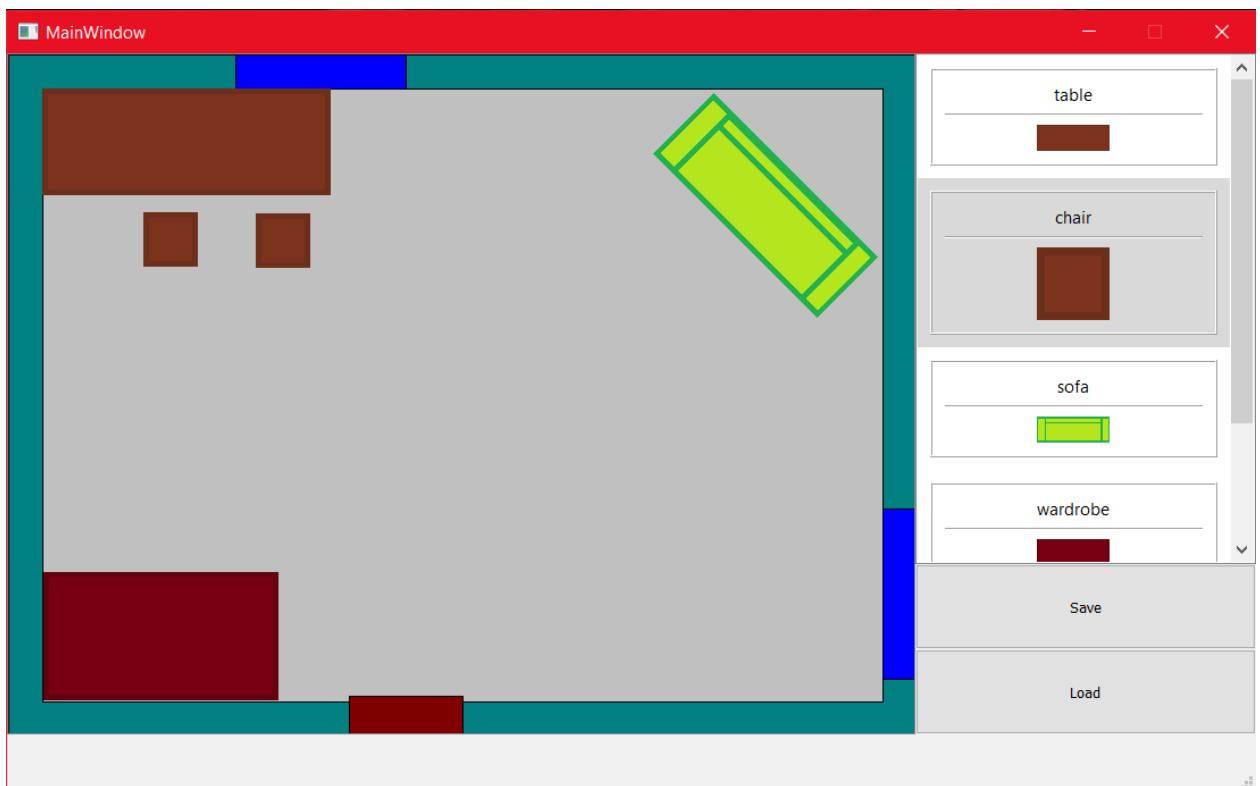


Рисунок 18

Загрузим сохраненный интерьер, нажав на кнопку Load (рисунок 19). Результат можно увидеть на рисунке 20.

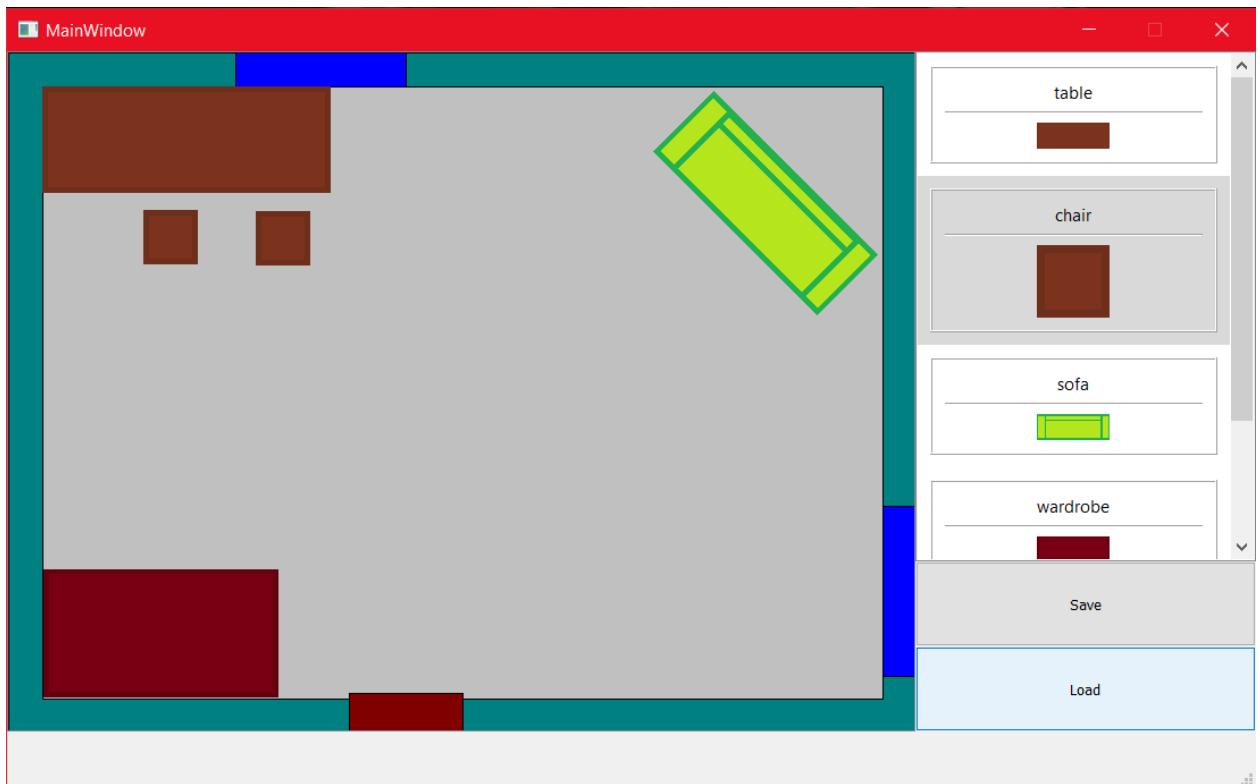


Рисунок 19

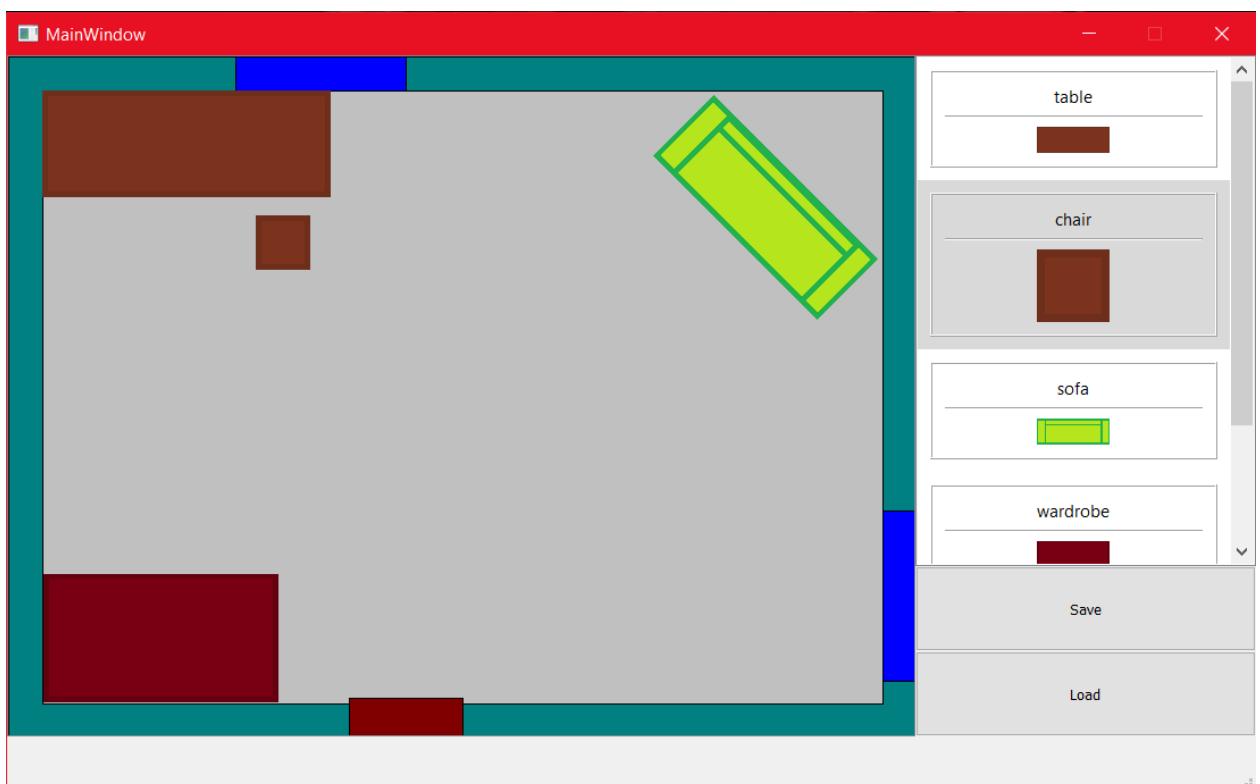
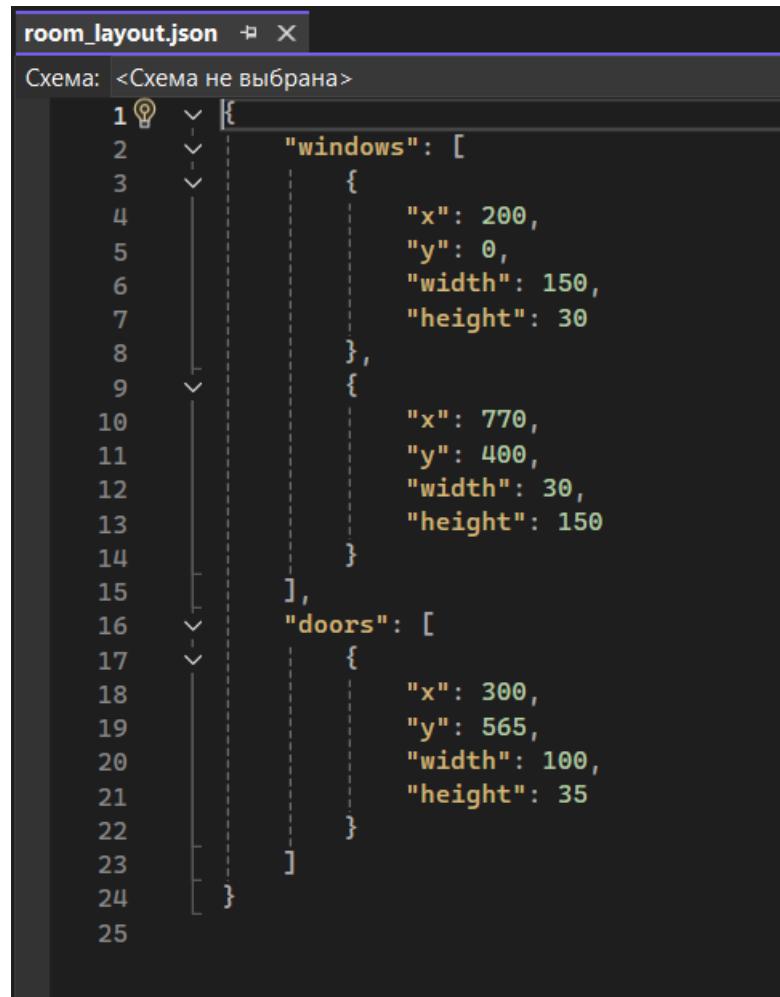


Рисунок 20

Зайдем в файл room_layout.json (рисунок 21), который хранит в себе данные о положении и размера двери и окон и изменим значение первого окна с 200 на 400 (рисунок 22).



```
room_layout.json ✘ X
Схема: <Схема не выбрана>
1 1 {
2   "windows": [
3     {
4       "x": 200,
5       "y": 0,
6       "width": 150,
7       "height": 30
8     },
9     {
10       "x": 770,
11       "y": 400,
12       "width": 30,
13       "height": 150
14     }
15   ],
16   "doors": [
17     {
18       "x": 300,
19       "y": 565,
20       "width": 100,
21       "height": 35
22     }
23   ]
24 }
25 }
```

Рисунок 21

```
room_layout.json* ⓘ X
Схема: <Схема не выбрана>
1   {
2     "windows": [
3       {
4         "x": 400,
5         "y": 0,
6         "width": 150,
7         "height": 30
8       },
9       {
10        "x": 770,
11        "y": 400,
12        "width": 30,
13        "height": 150
14      }
15    ],
16    "doors": [
17      {
18        "x": 300,
19        "y": 565,
20        "width": 100,
21        "height": 35
22      }
23    ]
24  }
25
```

Рисунок 22

Закроем приложение и заново запустим его. В результате можно увидеть, что окно на верхней стене изменило свое положение (рисунок 23).

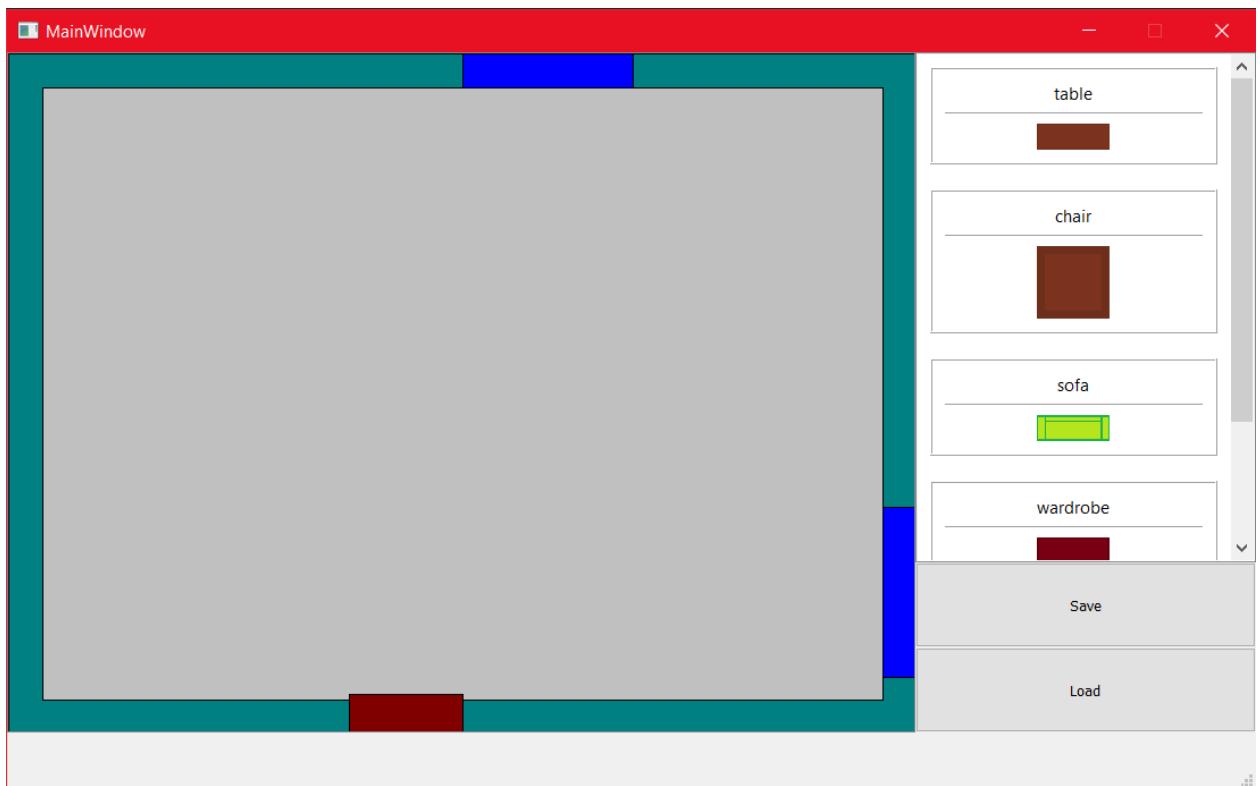


Рисунок 23

Загрузим интерьер. В итоге загружается сохраненный интерьер, положение окна не изменилось (рисунок 24).

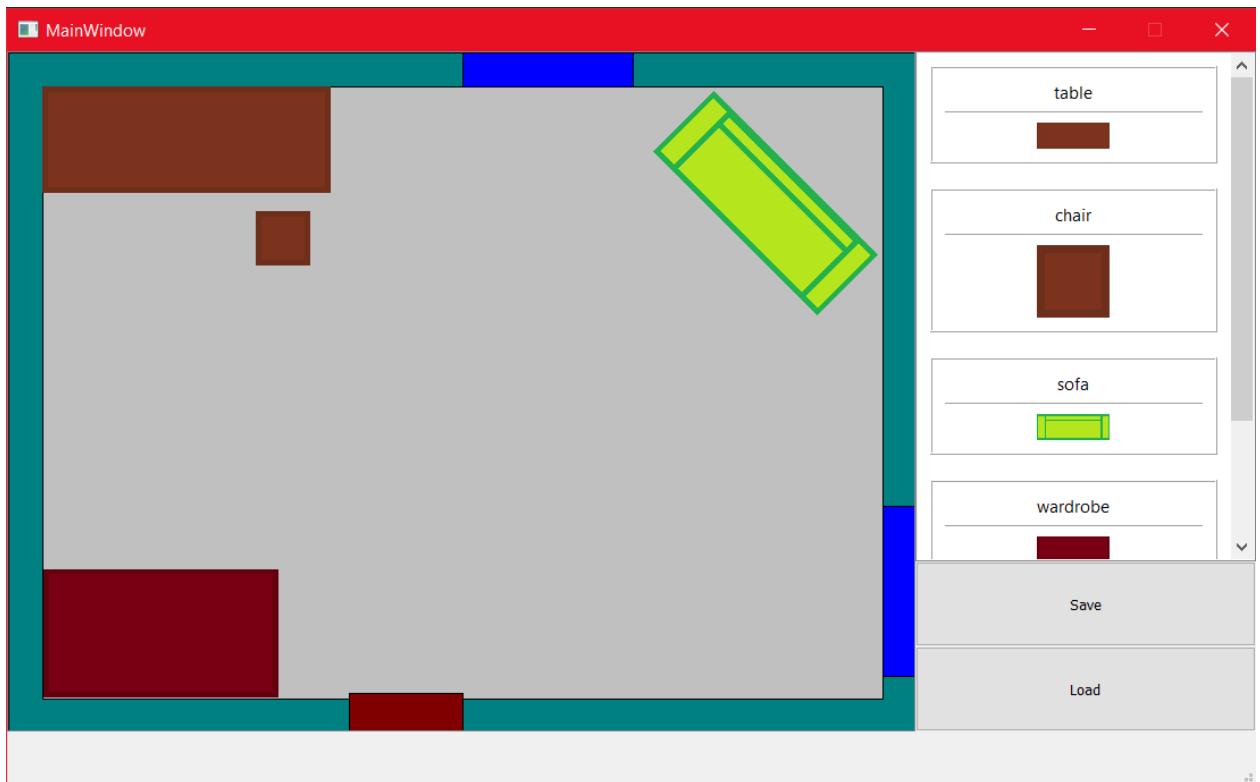


Рисунок 24.

Инструкция по работе с программой

В панели задач можно выбрать тип мебели и поместить его в интерьер комнаты. В пределах интерьера можно курсором и зажатой ЛКМ перемещать объекты по комнате. При нажатии ПКМ на объект, в пределах него появиться контекстное меню, в котором можно повернуть объект и изменить его размер. Чтобы сохранить интерьер, снизу на панели под элементами располагается кнопка, которая при нажатии сохраняет интерьер в формате json. Так же можно загрузить уже сохраненный интерьер.

Вывод

В результате работы было разработано приложение для моделирования расстановки мебели в прямоугольной комнате.

Основные этапы:

1. Чтение данных из файла: Загрузка параметров комнаты, включая размеры и расположение двери и окон.
2. Интерактивный ввод мебели: Пользователь может вводить размеры предметов мебели.
3. Расстановка мебели: Алгоритмы проверки допустимости расположения мебели с учетом дверей и окон.
4. Перестановка мебели: Возможность изменения расположения предметов.
5. Сохранение состояния: Запись конечного состояния комнаты в файл для дальнейшего использования.

Работа позволила закрепить навыки ООП и создать полезное приложение для визуализации интерьеров.