Оглавление

[Функциональные схема проекта: 1](#_Toc106022587)

[ПО и средства разработки: 2](#_Toc106022588)

[Оборудование, используемое для проекта: 3](#_Toc106022589)

[Описание реализации задач проекта: 3](#_Toc106022590)

[Применение приложения “Навигатор” на этапе администрирования. 3](#_Toc106022591)

[Код программы на ПК 15](#_Toc106022592)

[Код android-приложения 27](#_Toc106022593)

[Источники: 31](#_Toc106022594)

[Приложение 1: 31](#_Toc106022595)

[Приложение 2: 35](#_Toc106022596)

# Функциональные схема проекта:

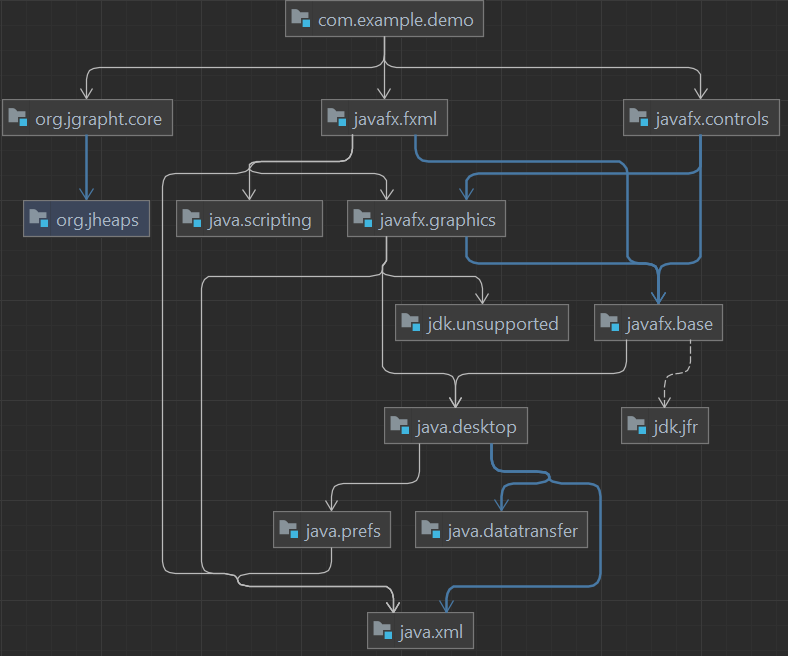


Рис.1. Схема классов ПК программы.

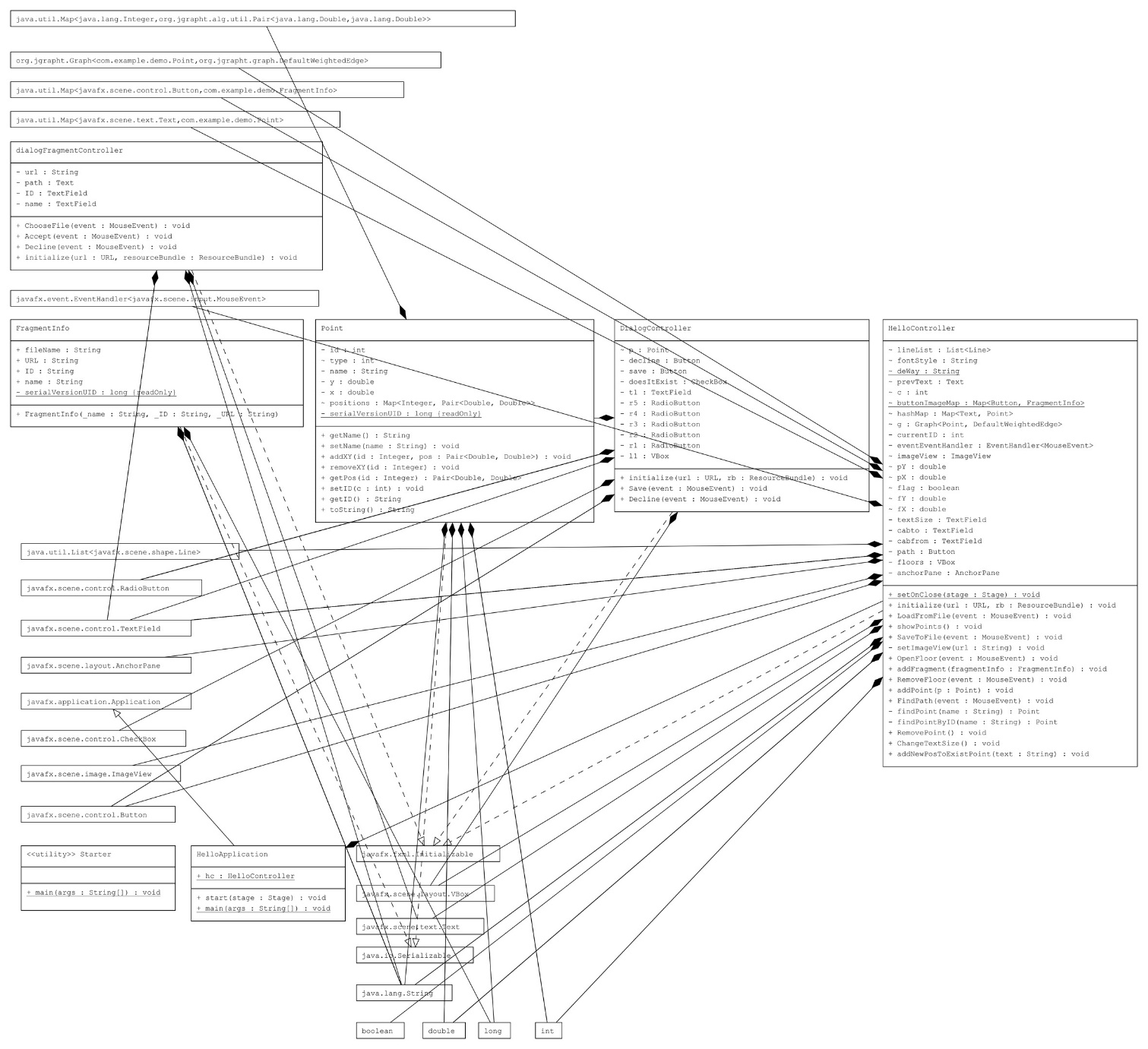


Рис.2. Авто сгенерированная UML диаграмма.

# ПО и средства разработки:

IntelliJ IDEA Community Edition 2021.3.3

Ссылка на скачивание: <https://www.jetbrains.com/idea/download/#section=windows>

Android Studio Chipmunk 2021.2.1

Ссылка на скачиваение: <https://developer.android.com/studio>

#### Java SE Development Kit 17.0.3.1

#### Ссылка на скачивание: <https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>

#### Scene Builder **18.0.0**

#### **Ссылка на скачивание: https://gluonhq.com/products/scene-builder/**

# Оборудование, используемое для проекта:

#### Ноутбук\ПК, телефон с ОС Android или его эмулятор.

# Описание реализации задач проекта:

#### Основной задачей проекта было написание навигатора по вузу. Навигатор должен обладать возможностью строить маршруты из точки А в точку Б. Для достижения поставленной цели было решено использовать неориентированный взвешенный граф, однако строить руками граф для всех кабинетов СПбГУТ’а было бы слишком трудоемко и не слишком удобно, потому что помимо названия кабинета и подходов надо было бы прописывать координаты для точки, а из-за отсутствия визуализации поиск ошибок стал бы крайне трудоемкой задачей. Таким образом, появилась сопутствующая задача – написать программу для ПК, способную строить карту для андроид-приложения.

Ниже предоставлен ее интерфейс.

# Применение приложения “Навигатор” на этапе администрирования.

Приложение “Навигатор” было разработано с учётом возможности изменения здания или помещения, с которым оно будет взаимодействовать, из чего следует возможность настроить локацию, по которой будет работать “Навигатор”.

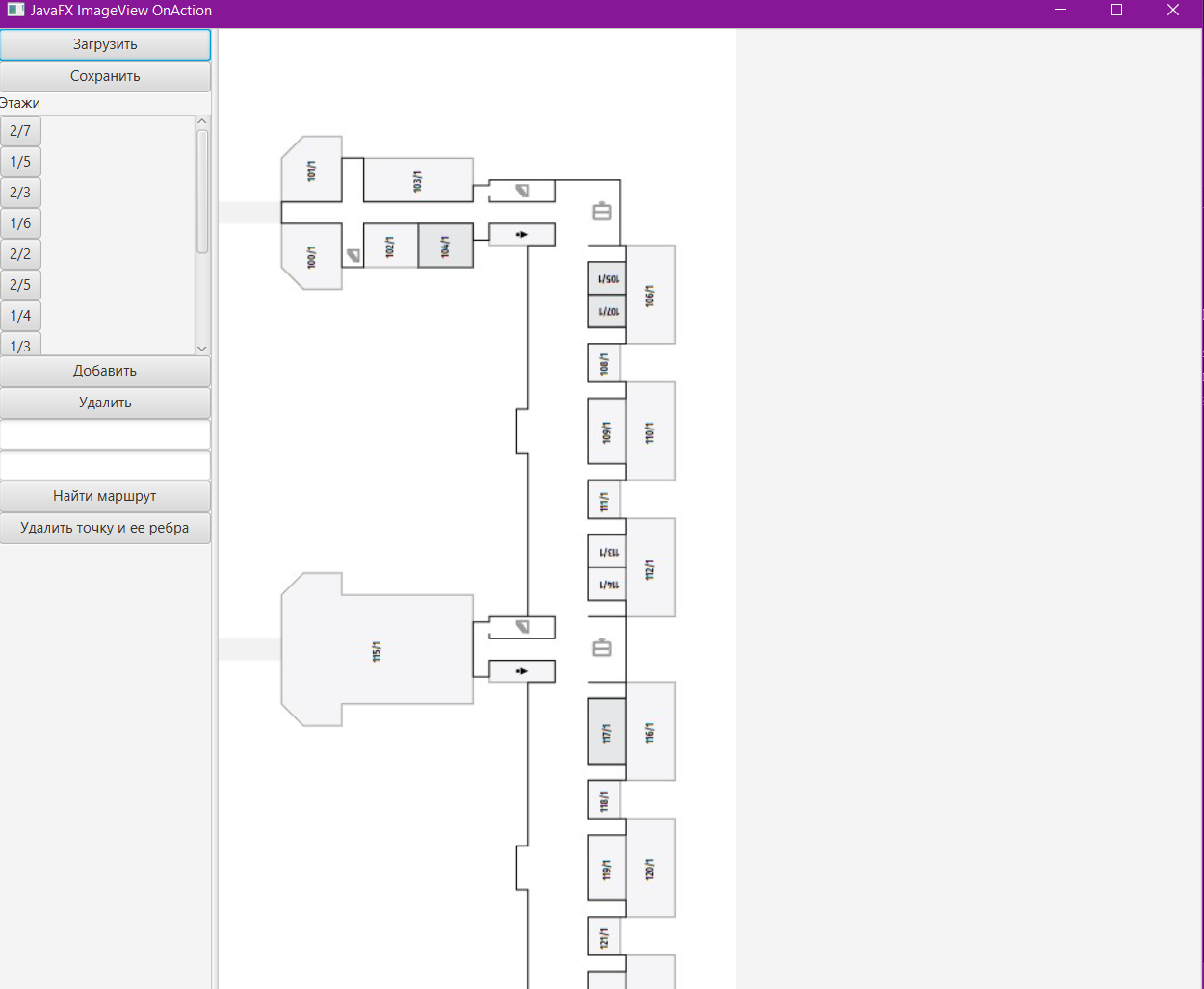


Рис.3. Интерфейс рабочей зоны для администрирования.

Клавиша “Загрузить” (рис.4) отвечает за подгрузку графа на изображение помещения и/или этажа здания, но стоит заметить, что если граф не был до этого построен, то и подгружать программе будет нечего. Пример с подгруженными графами см. на рис.5.

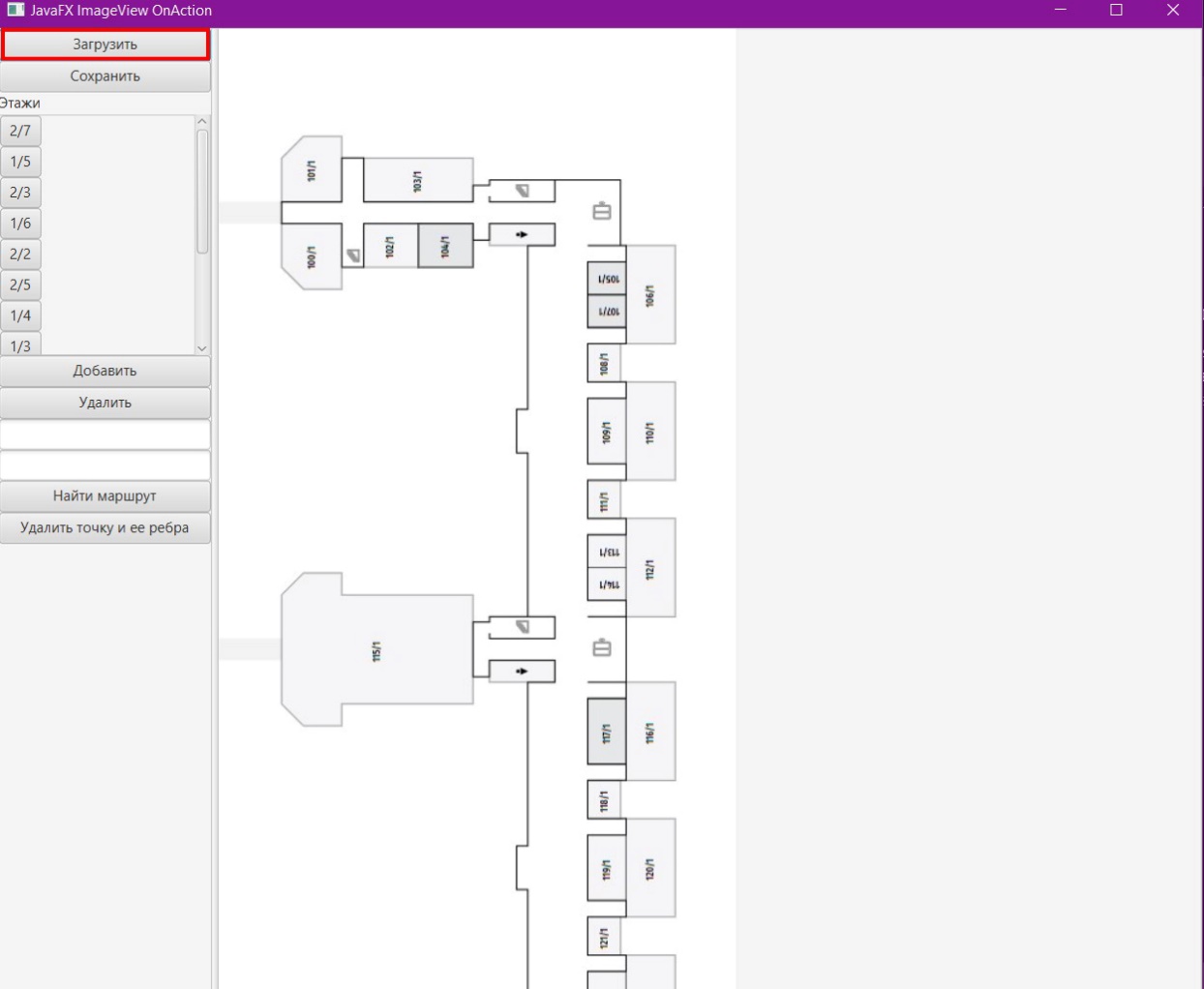


Рис.4. Клавиша “Загрузить”.

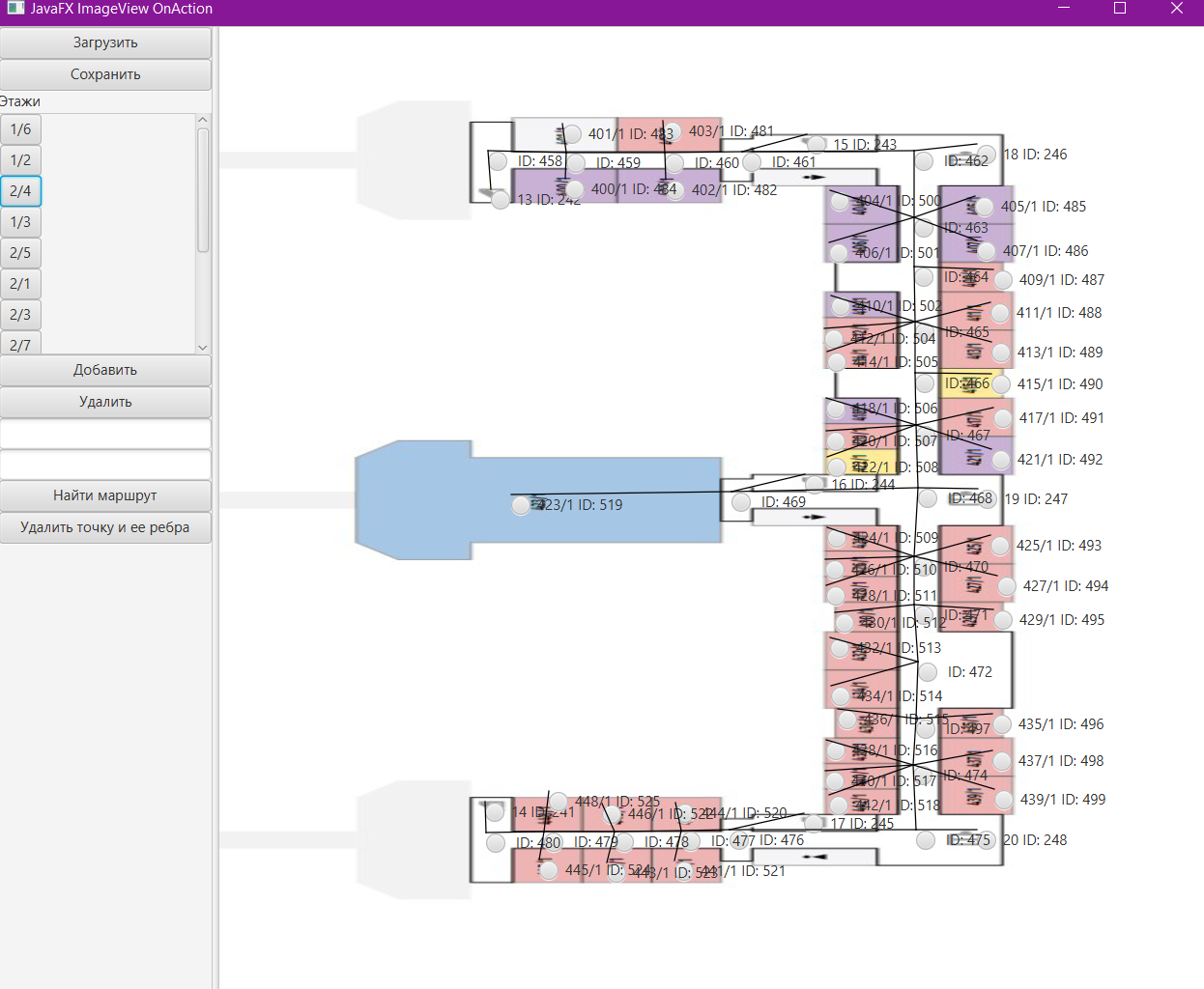


Рис.5. Граф успешно загружен.

Кнопка “Сохранить” (Рис.6), отвечает за сохранение построенного графа, поэтому после каждого действия необходимо её нажимать, чтобы сохранить прогресс.

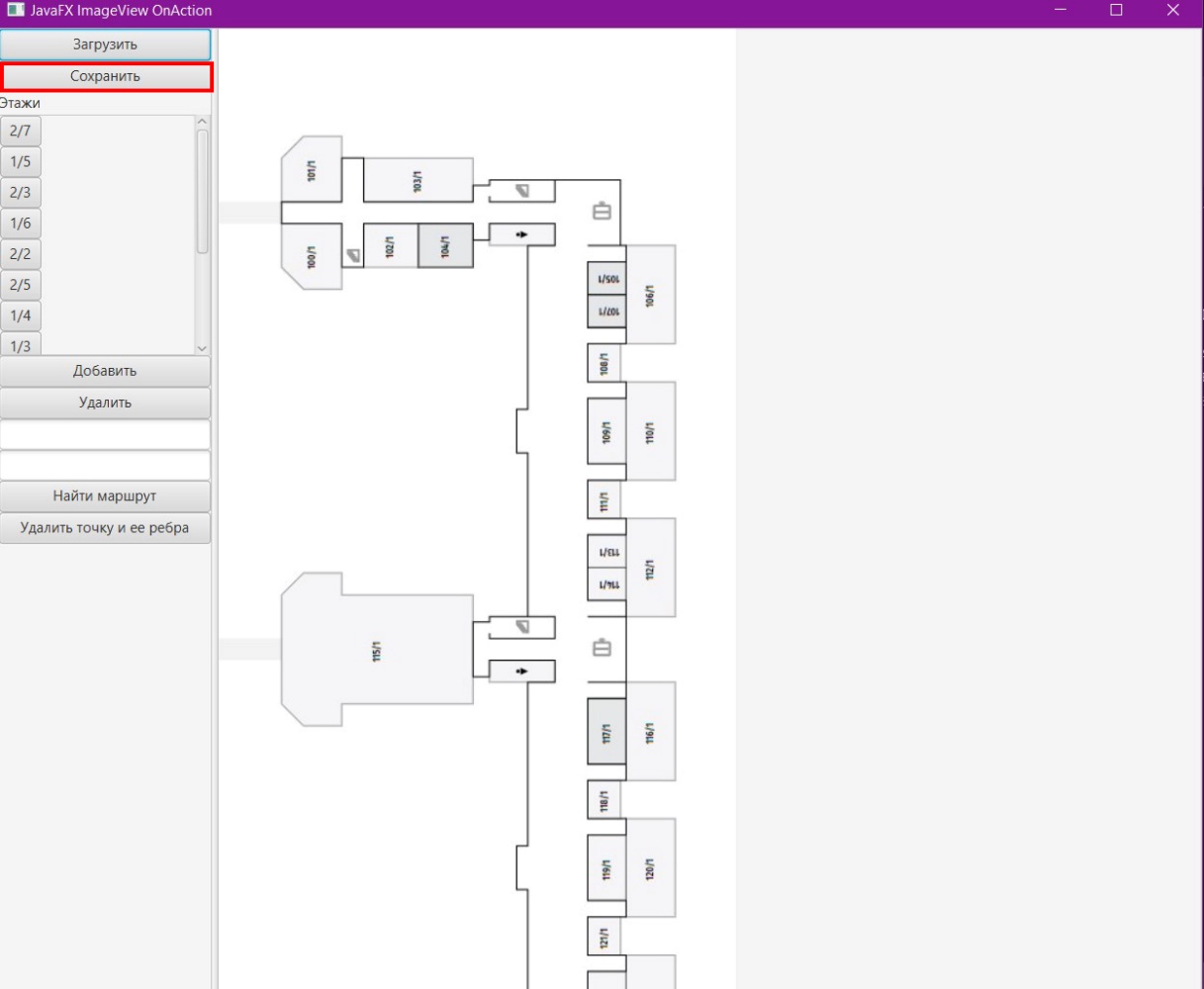


Рис.6. Кнопка “Сохранить”.

Область с надписью “Этажи” (рис.7) отвечает за отображение и переключение между этажами и/или отдельными помещениями (если таковые имеются), для дальнейшей проверки или построения графа.

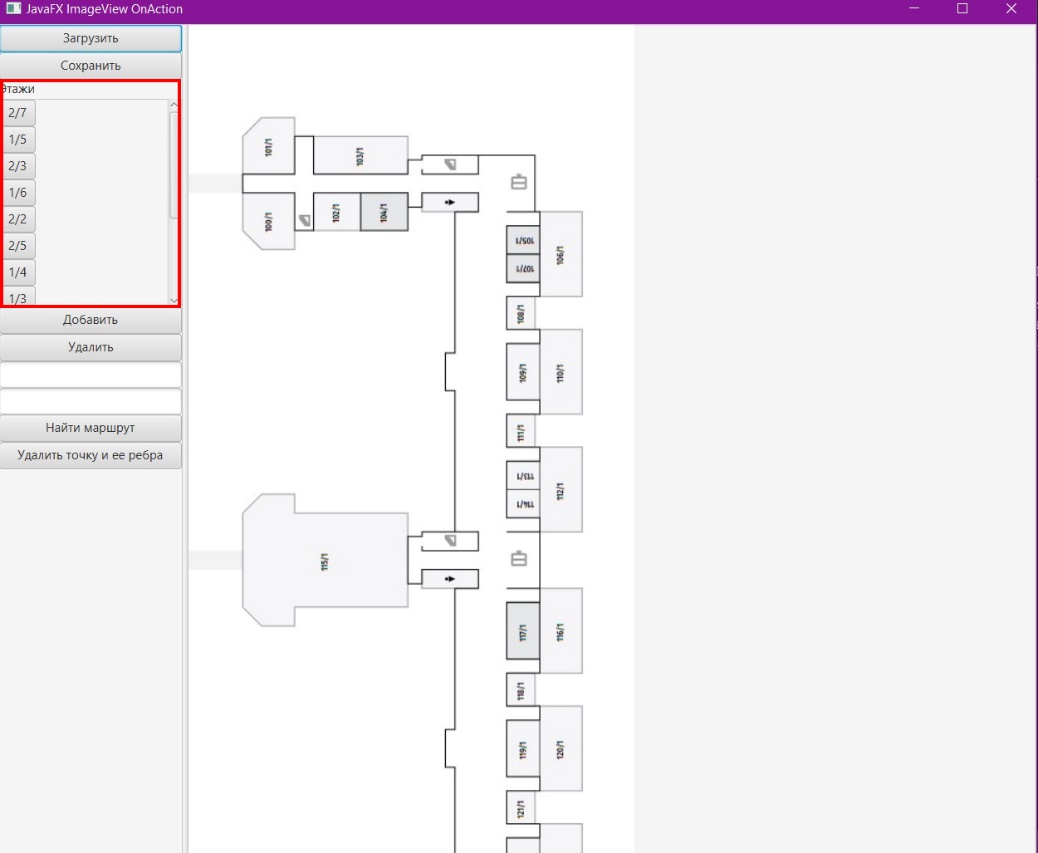


Рис.7. Область с надписью “Этажи”.

Функция “Добавить” (рис.8) имеет свой интерфейс (см. рис.9) и отвечает за функцию добавления изображения этажей и/или помещения в форматах Png или JPEG.

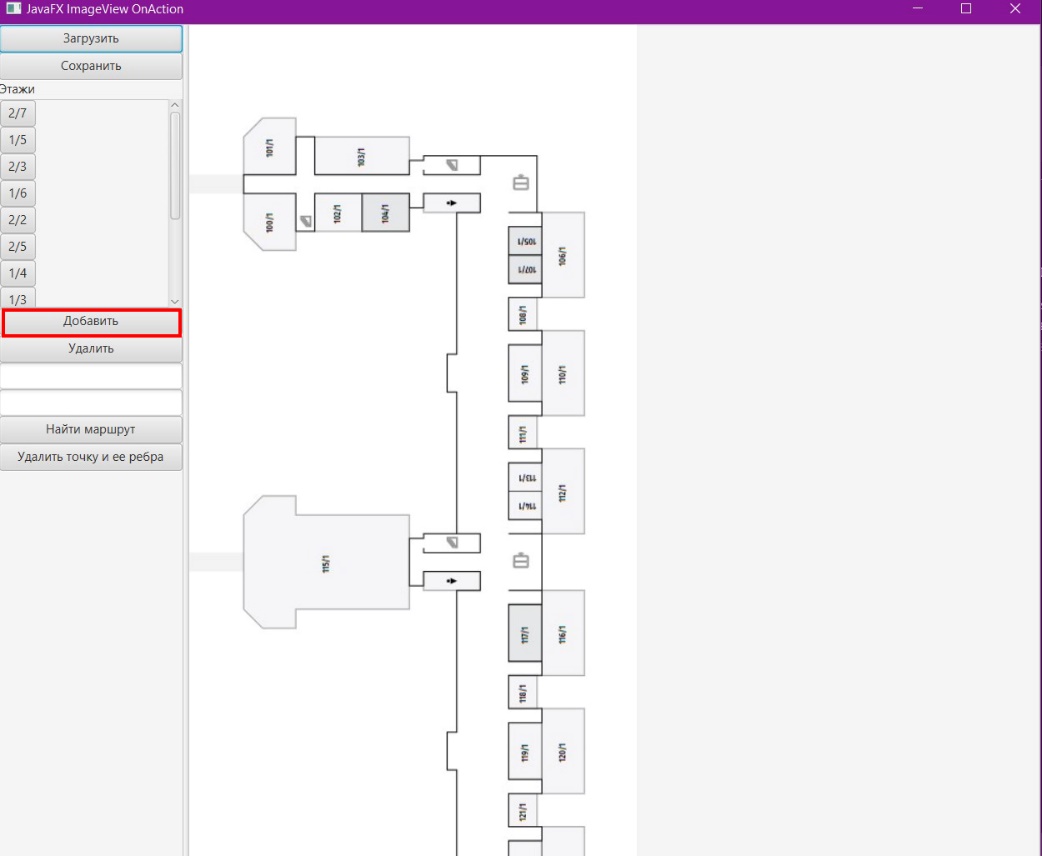


Рис.8. Функция “Добавить”.

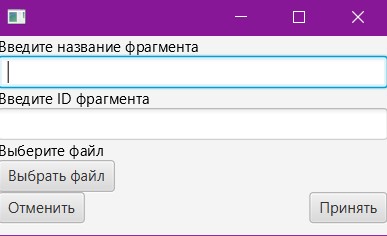


Рис.9. Интерфейс добавления.

В строке “Введите название фрагмента” пользователю необходимо указать уникальное название этажа и/или помещения, а также в строчке “Введите ID фрагмента” необходимо присвоить “Фрагменту” уникальный ID для опознавания его программой. Функция же кнопки “Выберите файл” предоставляет возможность пользователю выбрать необходимое заранее заготовленное изображение этажа и/или помещения для добавления его в рабочую область. Функционал кнопок “Отменить” и “Принять” в представлении не нуждается.

Клавиша “Удалить” (рис.10) служит для удаления из программы уже загруженных изображений этажей и/или помещений, для этого необходимо выбрать интересующий вас этаж в области “Этажи”, после чего щёлкнуть по кнопке “Удалить”.

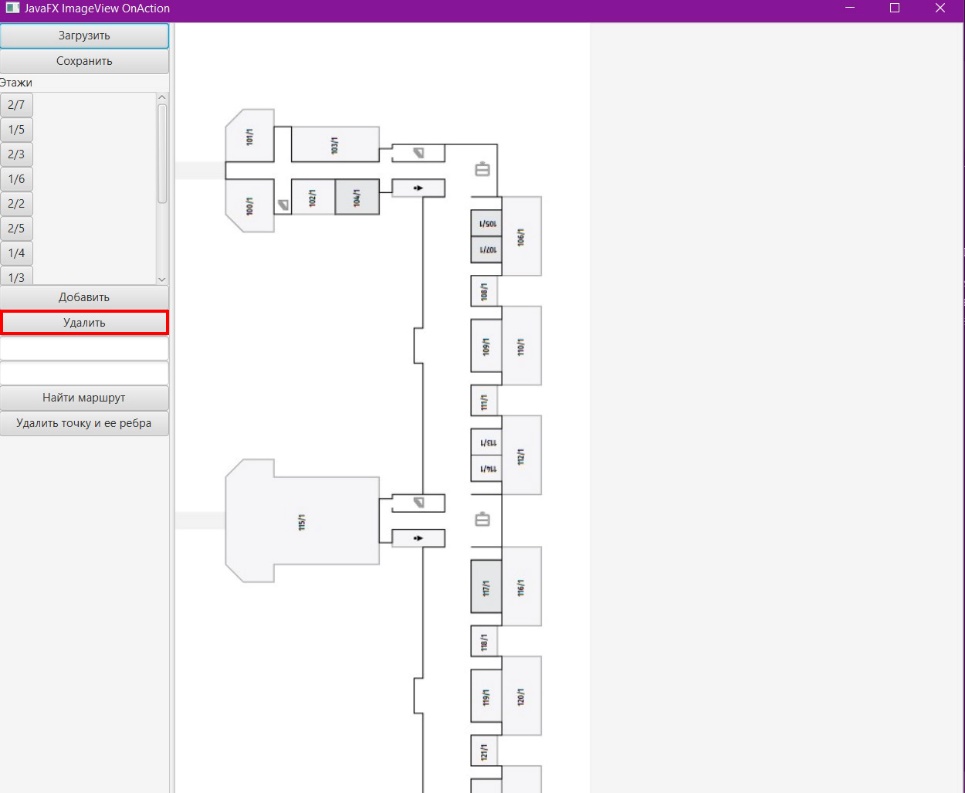


Рис.10. Клавиша “Удалить”.

Ниже кнопки “Удалить” располагаются две пустые зоны для ввода (рис.11), служащие для проверки работоспособности программы и правильности построения графа, верхняя область для отправной точки, а нижняя для конечного пункта.

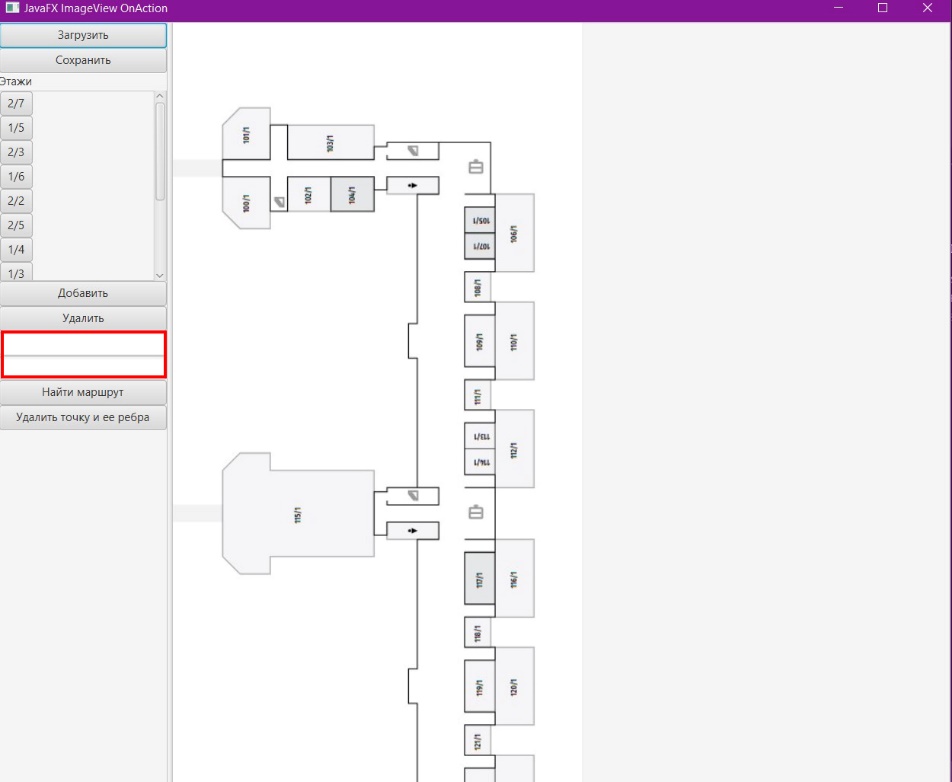


Рис.11. Зоны для ввода кабинетов.

Следом идёт клавиша “Найти маршрут” (рис.12), которая и является триггером для введенных выше значений “Начала” и “Конца” пути.

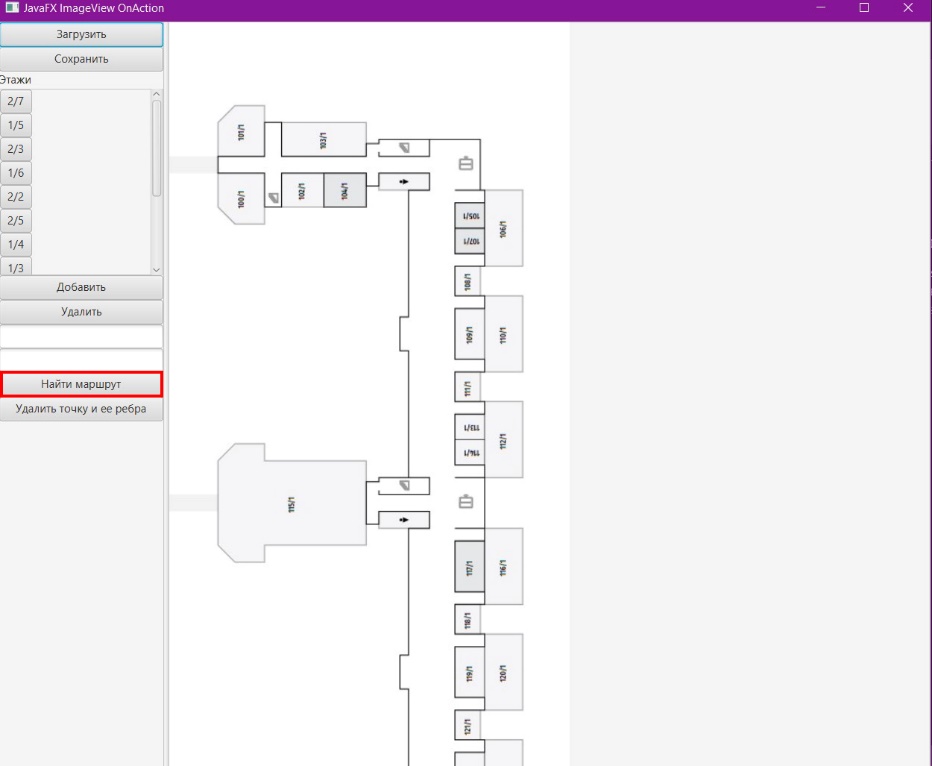


Рис.12. Клавиша “Найти маршрут”.

Кнопка “Удалить точку и её рёбра” (рис.13) необходима для построения графиков. Данная функция позволяет удалить точку графа, а также все соединения с другими точками (рёбра).

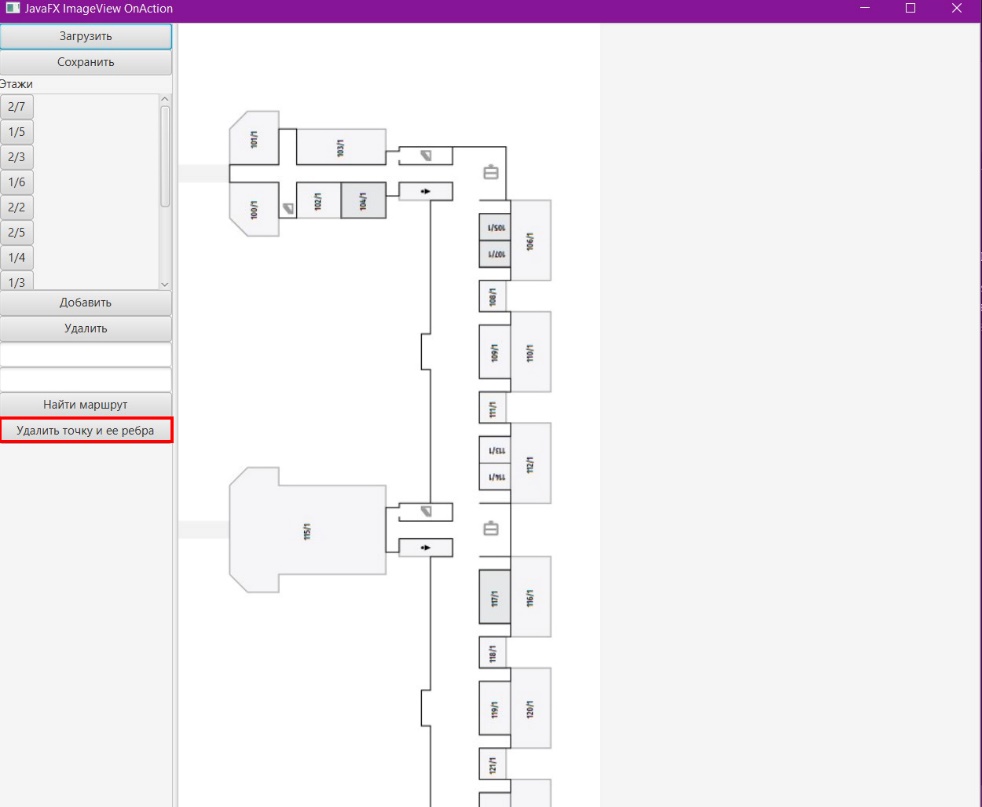


Рис.13. Кнопка “Удалить точку и её рёбра”.

Самой же большой областью программы является “Рабочая зона” (Рис.14) или же зона для визуального отображения действий пользователя, а также зона для построения графа и рёбер. В данной области и будет происходить большая часть работы с программой.

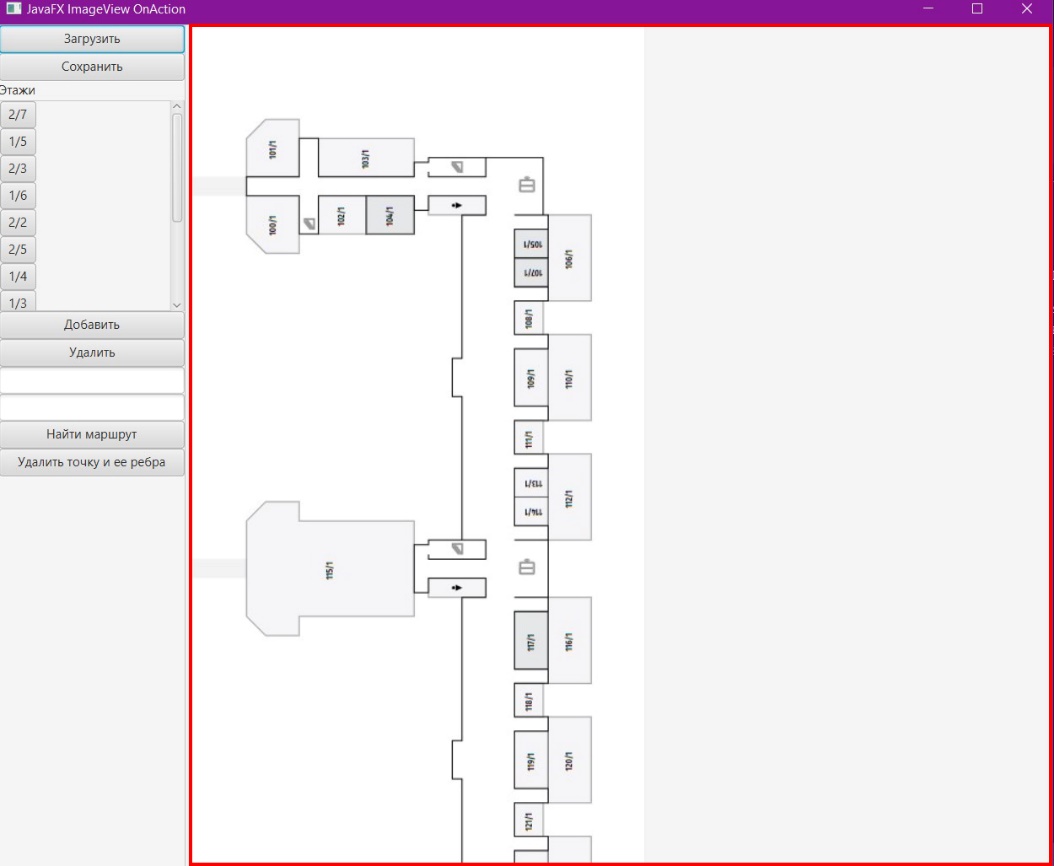


Рис.14. Рабочая зона.

После формирования карты и ее сохранения, необходимо найти в директории программы папку files и перенести оттуда все файлы, где написано toandroid в ресурсы андроид-приложения, в папку raw(опционально ее надо будет создать). Кроме того, неплохо бы фрагменты карты, с которой работает программа перенести в папку files – pictures и брать их уже оттуда.

Также стоит отметить, что на последних этапах разработки для удобства пользователя была добавлена новая функция изменения размера текста (рис. 14.1).

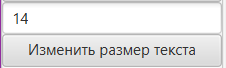


Рис.14.1. Изменение размера текста

Помимо текстового отчета существует обучающее видео: <https://drive.google.com/file/d/1tbTZRYvCIxbiws1Xf0a5id2SQmNvFK8c/view?usp=sharing>

Применение приложения “Навигатор” с точки зрения пользователя.

Второй частью программы является интерфейс, который видит пользователь (рис.15), а не администратор, его функционал куда меньше, но внешний вид гораздо более интуитивно понятен, для применения гораздо большим числом пользователей, а также данная часть программы используется уже не на OC Windows, а на платформе Android.

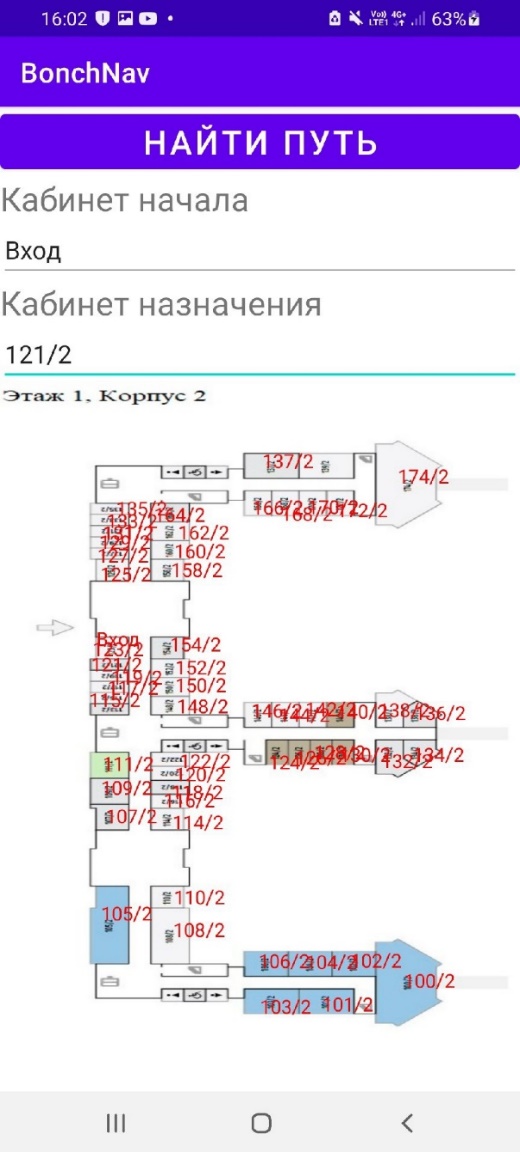


Рис.15. Интерфейс мобильного приложения.

Интерфейс программы интуитивно понятен, поля для ввода уже были рассмотрены выше, но, если повторяться, они служат для определения начального и конечного пункта назначения, но также здесь представлена кнопка “Найти путь” расположенная в верхней части приложения прямо под названием, она является триггером для нахождения программой кратчайшего пути из начальной точки в конечную, заданных в поле для ввода. Введя начальный и конечный в данном случае кабинеты и нажав клавишу найти путь вам будет показан маршрут (рис.16,17,18).

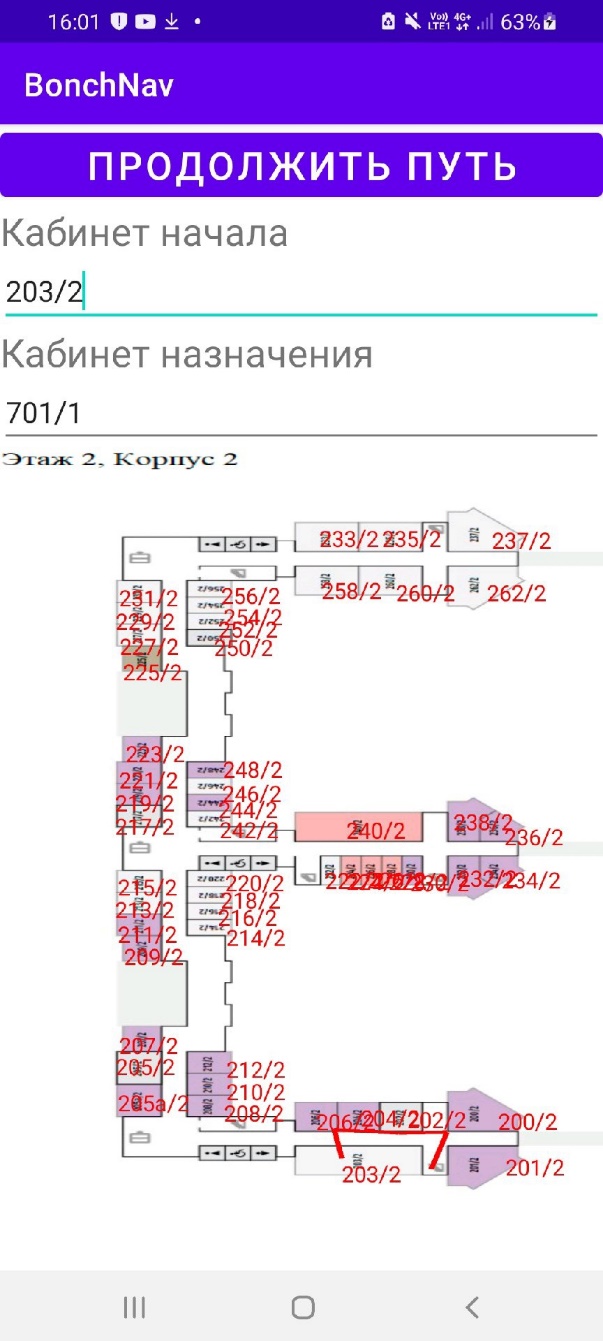


Рис.16. Показанный в приложении маршрут.

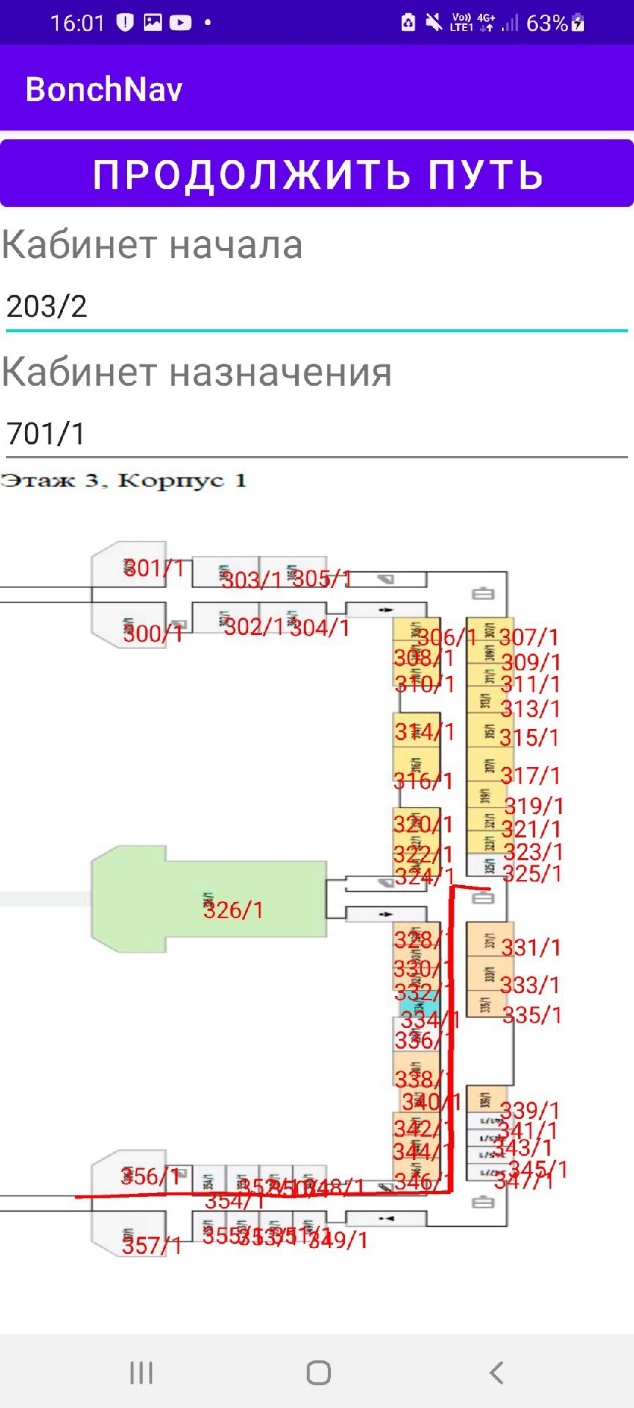


Рис.17. Показанный в приложении маршрут.

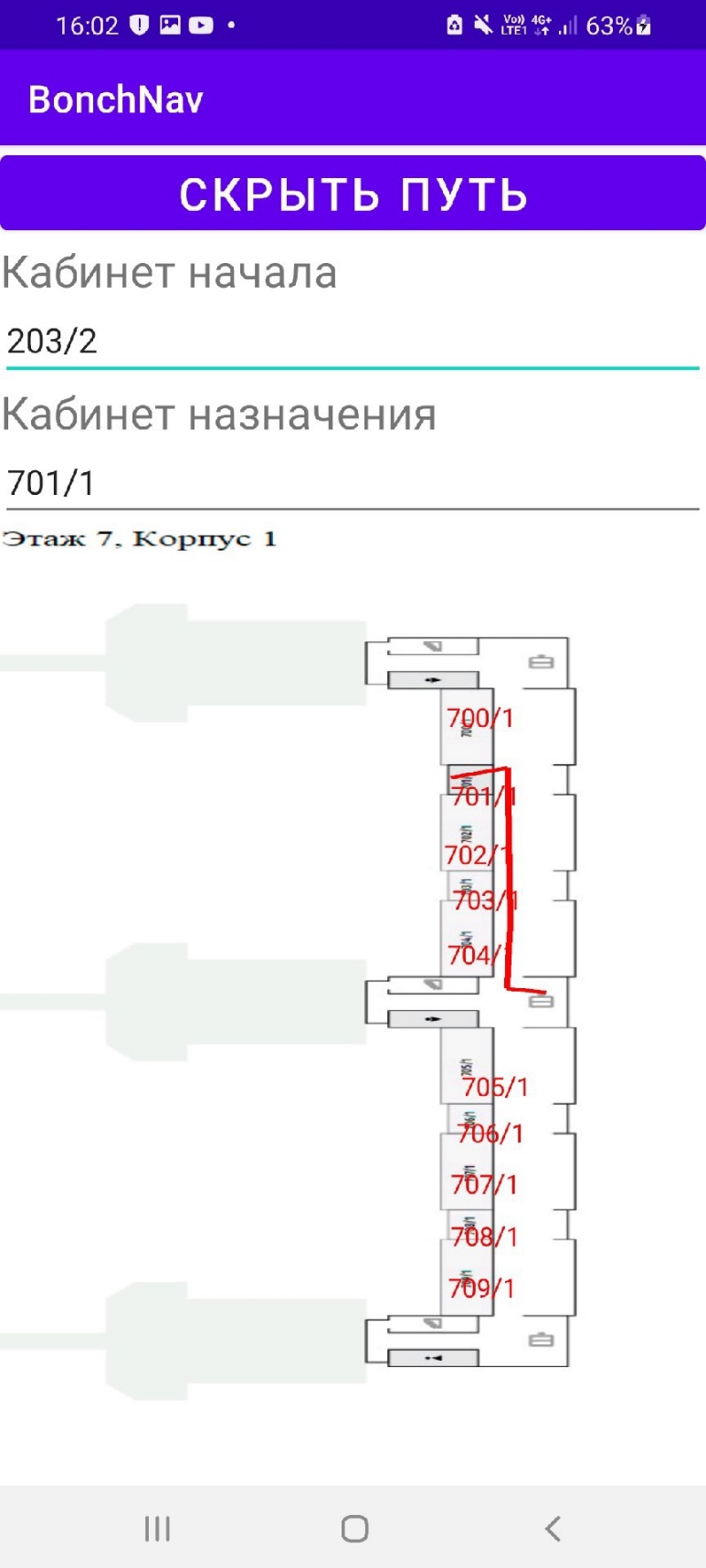


Рис.18. Показанный в приложении маршрут.

Как мы можем заметить, кнопка “Найти путь” сменилась на “Продолжить маршрут”, а затем и вовсе на “скрыть путь”, данные клавиши отвечают за “перелистывание” этажа и сигнала программе о том, что вы добрались до пункта назначения соответственно. Таким образом программа привела нас от входа в кабинет 701 первого корпуса СПБГУТ им. Профессора М. А. Бонч-Бруевича, по кратчайшему пути используя алгоритм Дейкстры.

# Код программы на ПК

Starter.java:

Очень важный файл, main-метод которого отмечен, как точка входа в программу. Необходимо для обхода бага IntelliJ IDEA(которых там, справедливости ради, великое множество) и для того, чтобы программа могла запускаться везде, где есть Java, а не Java и JavaFX. По-простому – костыль, без которого не работало.

public class Starter {  
 public static void main(String[] args){  
 HelloApplication.*main*(args); //Вызываем main-метод полноценного главного класса  
 }  
}

HelloApplication.java:

Главный класс программы, из которого она и должна запускаться. Впрочем, работа все будет идти в следующем классе, потому что попытка в MVC(настоятельно рекомендую ознакомиться с этой статьей <https://habr.com/ru/post/520172/> чтобы дальше стало понятнее) не увенчалась успехом.

import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.stage.Stage;//подключаем необходимые классы  
import java.io.IOException;  
  
public class HelloApplication extends Application {  
  
 public static HelloController *hc*; //ссылка на объект контроллера окна программы. Нужна для взаимодействия разных классов друг с другом  
  
 @Override  
 public void start(Stage stage) throws IOException {  
 Parent root = FXMLLoader.*load*(getClass().getResource("hello-view.fxml")); //загрузка .fxml файла  
 Scene = new Scene(root);   
 stage.setTitle("MapBuilder");  
 stage.setScene(scene);  
 stage.sizeToScene();//размер stage будет соответствовать размеру сцены  
 stage.show();  
 HelloController.*setOnClose*(stage); //установка обработчика события окна(метод setOnClose() реализован в контроллере.  
 }  
  
 public static void main(String[] args) {  
 *launch*();//запуск JavaFX приложения  
 }  
}

HelloController.java:

Класс, в котором происходит работа программы. Главный класс. Тут стоит внести некоторую ясность: карта в данном приложении состоит из двух слоев. Первый слой – это графическое изображение, а второй – это слой точек, которые соответствуют различным элементам графического изображения(кабинеты, коридоры, повороты, переходы). Помимо этого на этом слое располагаются линии, соединяющие эти точки. Точки в программе хранятся в виде графа, в узлах которого хранятся объекты класса Point, но о нем речь пойдет дальше, а пока ясность внесена успешно.

import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.fxml.Initializable;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.TextField;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.input.MouseEvent;  
import javafx.scene.layout.AnchorPane;  
import javafx.scene.layout.VBox;  
import javafx.scene.paint.Color;  
import javafx.scene.shape.Line;  
  
import java.io.\*;  
import java.net.URL;  
import java.util.\*;  
  
import javafx.scene.text.Text;  
import javafx.stage.Stage;  
import javafx.stage.WindowEvent;  
import org.jgrapht.Graph;  
import org.jgrapht.GraphPath;  
import org.jgrapht.alg.shortestpath.DijkstraShortestPath;  
import org.jgrapht.alg.util.Pair;  
import org.jgrapht.graph.DefaultWeightedEdge;  
import org.jgrapht.graph.SimpleWeightedGraph;  
  
public class HelloController implements Initializable {  
   
 @FXML  
 private AnchorPane anchorPane; //layout на котором будет отображаться imageView c графическим изображением карты и view, соответствующие точкам графа   
 @FXML  
 private VBox floors; //layout в который будут выводиться кнопки, соответствующие различным фрагментам карты  
 @FXML  
 private Button path; //кнопка для поиска пути  
 @FXML  
 private TextField cabfrom; //текстовое поле, для ввода названия точки, от которой будет строиться маршрут  
 @FXML  
 private TextField cabto; //текствое поле, для ввода названия точки, к которой будет строиться маршрут  
 @FXML  
 private TextField textSize; //иногда возникает проблема, связанная с тем, что view обозначающие точки перекрывают место, в которое вы хотите добавить новую точку, поэтому размер текста в них можно уменьшить или наоборот увеличить  
  
 double fX = -1;  
 double fY = -1;  
 boolean flag = true; //3 переменные необходимые для добавления связей между точками. Дальше по коду станет понятнее   
  
 double pX, pY; // Я реально не помню, что это и зачем оно тут  
 ImageView imageView;  
  
 private EventHandler<MouseEvent> eventEventHandler = new EventHandler<MouseEvent>() { //обработчик нажатия на view точки  
 @Override//у которого есть 2 варианта развития событий  
 public void handle(MouseEvent mouseEvent) {  
 if (flag) {//на view нажимают впервые   
 fX = ((Text) mouseEvent.getSource()).getLayoutX();  
 fY = ((Text) mouseEvent.getSource()).getLayoutY();  
 prevText = ((Text) mouseEvent.getSource());//сохраняются координаты этой точки и указатель на нее(хотя можно было бы сохранять только указатель, думаю, но проверку этого оставлю на усмотрение читателя)  
 flag = false;   
 } else {//на view нажимают второй раз  
 double nX = ((Text) mouseEvent.getSource()).getLayoutX();  
 double nY = ((Text) mouseEvent.getSource()).getLayoutY();  
 anchorPane.getChildren().add(new Line(fX, fY, nX, nY));//сохраняются координаты второй точки, добавляется графическое отображение нового ребра  
 flag = true;  
 DefaultWeightedEdge dwe = new DefaultWeightedEdge();  
 g.addEdge(hashMap.get(prevText), hashMap.get((Text) mouseEvent.getSource()), dwe);  
 g.setEdgeWeight(dwe, Math.*sqrt*((fX - nX) \* (fX - nX) + (fY - nY) \* (fY - nY)));//добавляется новое ребро между точками  
 System.*out*.println(g.getEdgeWeight(dwe));  
 System.*out*.println(g);  
 }  
 }  
 };  
 private int currentID;  
 Graph<Point, DefaultWeightedEdge> g = new SimpleWeightedGraph<>(DefaultWeightedEdge.class); //граф, хранящий карту   
 Map<Text, Point> hashMap = new HashMap<>(); //карта, хранящая связь графических элементов с точками  
 static Map<Button, FragmentInfo> *buttonImageMap* = new HashMap<>(); //карта, хранящая связь кнопок с фрагментами карты  
 int c = 0;  
 Text prevText;  
 static String *deWay* = System.*getProperty*("user.dir") +"\\files\\";//директория, в которой работает программа  
 String fontStyle = "-fx-font: 7 arial;"; // стиль текста по умолчанию. Теоретически сюда можно сделать css-инъекцию и что-то взломать, но зачем?..   
  
 public static void setOnClose(Stage stage) {//тут устанавливается событие закрытия окна. При закрытии окна он сохраняет какие фрагменты, где использовались  
 stage.setOnCloseRequest(new EventHandler<WindowEvent>() {  
 @Override  
 public void handle(WindowEvent windowEvent) {  
 try {  
 ObjectOutputStream objectOutputStream = new ObjectOutputStream(new FileOutputStream(*deWay* + "bim.ser"));  
 ArrayList<FragmentInfo> arrayList = new ArrayList(*buttonImageMap*.values());  
 objectOutputStream.writeObject(arrayList);  
 objectOutputStream.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 }  
  
  
  
 @Override  
 public void initialize(URL url, ResourceBundle rb) {//при открытии окна происходит процесс, обратный тому, что происходит при закрытии – загружается список фрагментов, которые использовались в программе  
 HelloApplication.*hc* = this;  
 File theDir = new File(*deWay*);  
 if (!theDir.exists()){  
 theDir.mkdirs();  
 }  
 try {  
 boolean flag = (new File(*deWay* + "bim.ser")).exists();  
 if(flag) {  
 ObjectInputStream objectInputStream = new ObjectInputStream(new FileInputStream(*deWay* + "bim.ser"));  
 ArrayList<FragmentInfo> arrayList = (ArrayList<FragmentInfo>) objectInputStream.readObject();  
 for (FragmentInfo fragment :  
 arrayList) {  
 addFragment(fragment);//вот тут они добавляются на экран  
 }  
 objectInputStream.close();  
 }  
 } catch (Exception e) {  
 e.getMessage();  
 }   
 }  
  
 @FXML  
 public void LoadFromFile(MouseEvent event) {//метод, вызываемый при нажатии на кнопку Загрузить. Загружает ранее сохраненный граф  
 FileInputStream inputStream = null;  
 try {  
 ObjectInputStream objectInputStream = new ObjectInputStream(new FileInputStream(*deWay* + "gr.ser"));  
 g = (Graph<Point, DefaultWeightedEdge>) objectInputStream.readObject();  
 objectInputStream.close();  
 hashMap = new HashMap<>();  
 c = 0;  
 for (Point p:  
 g.vertexSet()) {  
 if(Integer.*parseInt*(p.getID()) > c) c = Integer.*parseInt*(p.getID()) + 1;  
 }  
 showPoints();  
 System.*out*.println(g);  
 System.*out*.println(hashMap);  
 objectInputStream.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 showPoints();  
 }  
  
 public void showPoints(){//отображение кнопок, принадлежащих данному фрагменту карты  
 for (Point p :  
 g.vertexSet()) {  
 if(p.getPos(currentID) != null) {  
 Text text = new Text(p.getName() + " ID: " + p.getID());  
 text.setStyle(fontStyle);  
 text.setLayoutX(p.getPos(currentID).getFirst() \* imageView.getFitWidth());  
 text.setLayoutY(p.getPos(currentID).getSecond() \* imageView.getFitHeight());  
 text.setId(p.getID());  
 text.setOnMouseClicked(eventEventHandler);  
 anchorPane.getChildren().add(text);  
 hashMap.put(text, p);  
 }  
 }  
  
 for (DefaultWeightedEdge defaultEdge :  
 g.edgeSet()) {  
 Point source = g.getEdgeSource(defaultEdge);  
 Point target = g.getEdgeTarget(defaultEdge);  
 if((source.getPos(currentID) != null) && (target.getPos(currentID) != null))  
 anchorPane.getChildren().add(new Line(source.getPos(currentID).getFirst() \* imageView.getFitWidth(), source.getPos(currentID).getSecond() \* imageView.getFitHeight(), target.getPos(currentID).getFirst() \* imageView.getFitWidth(), target.getPos(currentID).getSecond() \* imageView.getFitHeight()));  
 }  
 }  
  
 @FXML  
 public void SaveToFile(MouseEvent event) {//метод, вызываемый при нажатии на кнопку Сохранить. Сохраняет граф карты в файл.  
 try {  
 ObjectOutputStream objectOutputStream = new ObjectOutputStream(new FileOutputStream(*deWay* + "gr.ser"));  
 objectOutputStream.writeObject(g);  
 objectOutputStream.close();  
 objectOutputStream = new ObjectOutputStream(new FileOutputStream(*deWay* + "toandroid.ser"));  
 objectOutputStream.writeObject(g.toString());  
 objectOutputStream.close();  
 objectOutputStream = new ObjectOutputStream(new FileOutputStream(*deWay* + "toandroidpoints.ser"));  
 HashMap<String, Point> androidHashMap = new HashMap<>();  
  
 for (Point p:  
 g.vertexSet()) {  
 androidHashMap.put(p.getID(), p);  
 }//из-за того, что используемая библиотека Jgraph имеет какие-то проблемы с сериализацией, граф для передачи в приложение превращается в строку, состоящую из id точек, которая потом парсится в приложении. Помимо этого приходится сохранять очередную карту, в которой хранится связь ID точек и самих объектов класса Point   
  
 objectOutputStream.writeObject(androidHashMap);  
 objectOutputStream.close();  
 objectOutputStream = new ObjectOutputStream(new FileOutputStream(*deWay* + "toandroidfragments.ser")); //файлы, которые в названии содержат toandroid впоследствии нужно копировать в ресурсы андроид-приложения  
 ArrayList<FragmentInfo> arrayList = new ArrayList<FragmentInfo>(*buttonImageMap*.values());  
 objectOutputStream.writeObject(arrayList);  
 objectOutputStream.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 private void setImageView(String url) {//установка фонового изображения карты  
 anchorPane.getChildren().clear();  
 imageView = new ImageView(new Image(url));  
 imageView.setFitWidth(anchorPane.getWidth());  
 imageView.setFitHeight(anchorPane.getHeight());  
 imageView.setPickOnBounds(true);  
 imageView.setOnMouseClicked(new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {//обработчик события нажатия на imageView. Выводит на экран диалог, который просит задать параметры новой точки или отметить уже существующую точку на другом фрагменте(например если точке соответствует переход между этажами)  
 pX = mouseEvent.getX() / imageView.getFitWidth();  
 pY = mouseEvent.getY() / imageView.getFitHeight();  
 Parent root;  
 try {  
 FXMLLoader fxmlLoader = new FXMLLoader();  
 URL url = getClass().getResource("dialog.fxml");  
 root = fxmlLoader.*load*(url);  
 Stage stage = new Stage();  
 stage.setTitle("My New Stage Title");  
 stage.setScene(new Scene(root));  
 stage.sizeToScene();  
 stage.show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 anchorPane.getChildren().add(imageView);  
 }  
  
 @FXML  
 public void OpenFloor(MouseEvent event) {//метод, вызываемый при нажатии на кнопку Добавить. Создает диалоговое окно, в котором задаются параметры нового фрагмента  
 FXMLLoader fxmlLoader = new FXMLLoader();  
 Parent root = null;  
 try {  
 root = fxmlLoader.*load*(getClass().getResource("dialogFragment.fxml"));  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 Stage stage = new Stage();  
 stage.setScene(new Scene(root));  
 stage.sizeToScene();  
 stage.show();  
 }  
  
 public void addFragment(FragmentInfo fragmentInfo) {//Тут новый фрагмент добавляется в программу  
 fragmentInfo.URL = *deWay* + fragmentInfo.URL.substring(fragmentInfo.URL.indexOf("pictures\\"));  
 setImageView(fragmentInfo.URL);  
 Button btn = new Button(fragmentInfo.name);  
 btn.setId(fragmentInfo.ID);  
 currentID = Integer.*parseInt*(fragmentInfo.ID);  
 *buttonImageMap*.put(btn, fragmentInfo);  
 btn.setOnMouseClicked(new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent event) {  
 Button me = (Button) event.getSource();  
 anchorPane.getChildren().removeAll();  
 currentID = Integer.*parseInt*(me.getId());  
 setImageView(*buttonImageMap*.get(me).URL);  
 showPoints();  
 }  
 });  
 floors.getChildren().add(btn);  
 }  
  
 @FXML  
 public void RemoveFloor(MouseEvent event) {//а тут при желании выбранный фрагмент можно удалить  
 Button btn = (Button) floors.lookup("#" + currentID);  
 *buttonImageMap*.remove(btn);  
 floors.getChildren().remove(btn);  
 currentID = -1;  
 anchorPane.getChildren().clear();  
 imageView = null;  
 }  
  
 public void addPoint(Point p) {//тут в программу добавляются новые точки  
 p.addXY(currentID, new Pair<Double, Double>(pX, pY));  
 p.setID(c);  
 c++;  
 g.addVertex(p);  
 Text rd = new Text();  
 rd.setId(p.getID());  
 rd.setStyle(fontStyle);  
 rd.setText(p.getName() + " ID: " + p.getID());  
 hashMap.put(rd, p);  
 rd.setLayoutX(pX \* imageView.getFitWidth());  
 rd.setLayoutY(pY \* imageView.getFitHeight());  
 rd.setOnMouseClicked(eventEventHandler);  
 anchorPane.getChildren().add(rd);  
 }  
  
 List<Line> lineList = new ArrayList<>();  
  
 @FXML  
 public void FindPath(MouseEvent event) {//метод, в котором происходит поиск из пункта А в пункт Б. Предназначен для проверки правильности работы программы  
 if (lineList.size() == 0) {  
 Point from = findPoint(cabfrom.getText());  
 Point to = findPoint(cabto.getText());  
 DijkstraShortestPath dijkstraShortestPath = new DijkstraShortestPath(g);  
 GraphPath<Point, DefaultWeightedEdge> graphPathList = dijkstraShortestPath.getPath(from, to);  
 System.*out*.println(graphPathList.getWeight());  
 for (DefaultWeightedEdge defaultEdge :  
 graphPathList.getEdgeList()) {  
 Point source = g.getEdgeSource(defaultEdge);  
 Point target = g.getEdgeTarget(defaultEdge);  
  
 if((source.getPos(currentID) != null) && (target.getPos(currentID) != null)) {  
 Line line = new Line(source.getPos(currentID).getFirst() \* imageView.getFitWidth(), source.getPos(currentID).getSecond() \* imageView.getFitHeight(), target.getPos(currentID).getFirst() \* imageView.getFitWidth(), target.getPos(currentID).getSecond() \* imageView.getFitHeight());  
 line.setStroke(Color.*RED*);  
 anchorPane.getChildren().add(line);  
 lineList.add(line);  
 }  
 }  
 path.setText("Скрыть маршрут");  
 } else {  
 anchorPane.getChildren().removeAll(lineList);  
 lineList.removeAll(lineList);  
 path.setText("Найти маршрут");  
 }  
 }  
  
 private Point findPoint(String name) {//две функции, одна ищет точку по имени, другая по ID  
 for (Point point :  
 g.vertexSet()) {  
 if (point.getName().equals(name)) return point;  
 }  
 return null;  
 }  
  
 private Point findPointByID(String name) {  
 for (Point point :  
 g.vertexSet()) {  
 if (point.getID().equals(name)) return point;  
 }  
 return null;  
 }  
  
 @FXML  
 public void RemovePoint(){//точки можно удалять. Там остаются объекты Line на экране, но они ничего не означают  
 anchorPane.getChildren().remove(prevText);  
 Point p = hashMap.get(prevText);  
 p.removeXY(currentID);  
 if(p.positions.size() == 0) {  
 g.removeVertex(p);  
 hashMap.remove(prevText);  
 }  
 prevText = null;  
 flag = true;  
 }  
  
 @FXML  
 public void ChangeTextSize(){//при желании текст, обозначающий названия точек и кабинетов можно увеличить или уменьшить  
 fontStyle = "-fx-font: " + textSize.getText() + " arial;";  
 }  
  
 public void addNewPosToExistPoint(String text) {//добавление нового положения точки, если она уже существует  
 Point p = findPointByID(text);  
 p.addXY(currentID, new Pair<>(pX, pY));  
 Text rd = new Text();  
 rd.setId(p.getID());  
 rd.setText(p.getName() + " ID: " + p.getID());  
 rd.setStyle(fontStyle);  
 rd.setLayoutX(pX \* imageView.getFitWidth());  
 rd.setLayoutY(pY \* imageView.getFitHeight());  
 rd.setOnMouseClicked(eventEventHandler);  
 hashMap.put(rd, p);  
 anchorPane.getChildren().add(rd);  
 }  
}

Point.java:

Тут все просто – это класс точки

import org.jgrapht.alg.util.Pair;  
  
import java.io.Serializable;  
import java.util.HashMap;  
import java.util.Map;  
  
public class Point implements Serializable {  
 private static final long *serialVersionUID* = -3496931855932152057L;  
 Map<Integer, Pair<Double, Double>> positions = new HashMap<>();//карта хранящая записи ID фрагмента – положение на нем. Если точка не принадлежит фрагменту, записи нет;  
 private double x;  
 private double y;//рудиментарые образования   
 private String name = new String();  
  
 int type;  
 private int id;  
  
 public String getName() {//геттеры и сеттеры, разных форм и размеров.  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public void addXY(Integer id, Pair<Double, Double> pos) {positions.put(id, pos);}  
  
 public void removeXY(Integer id) {positions.remove(id);}  
  
 public Pair<Double, Double> getPos(Integer id){  
 return positions.get(id);  
 }  
  
 public void setID(int c) {  
 id = c;  
 }  
  
 public String getID() {  
 return String.*valueOf*(id);  
 }  
  
 @Override  
 public String toString(){//преобразование точки в строку.  
 return getID();  
 }  
}

FragmentInfo.java:

import java.io.Serializable;  
  
public class FragmentInfo implements Serializable {  
 private static final long *serialVersionUID* = -3496931855932152057L;  
 public String name;  
 public String ID;  
 public String URL;  
 public String fileName;  
 public FragmentInfo(String \_name, String \_ID,String \_URL){  
 name = \_name;  
 URL = \_URL;  
 fileName = URL.substring(URL.lastIndexOf('\\') + 1);//нужно для андроид-приложения, поскольку там файлы лежат в ресурсах  
 ID = \_ID;  
 }  
}

dialogFragmentController.java

import javafx.fxml.FXML;  
import javafx.fxml.Initializable;  
import javafx.scene.control.TextField;  
import javafx.scene.input.MouseEvent;  
import javafx.scene.text.Text;  
import javafx.stage.FileChooser;  
import javafx.stage.Stage;  
  
import java.io.File;  
import java.net.URL;  
import java.util.ResourceBundle;  
  
public class dialogFragmentController implements Initializable {  
  
 @FXML  
 private TextField name;  
 @FXML  
 private TextField ID;  
 @FXML  
 private Text path;  
  
 private String url;  
  
 public void ChooseFile(MouseEvent event){//вызов диалога выбора файла  
 FileChooser chooser = new FileChooser();  
 chooser.setTitle("Open File");  
 File file = chooser.showOpenDialog(path.getScene().getWindow());  
 if (file != null) {  
 url = file.getPath();  
 path.setText("Выбранный файл: " + url);  
 }  
 }  
 @FXML  
 public void Accept(MouseEvent event){//Добавить фрагмент   
 HelloApplication.*hc*.addFragment(new FragmentInfo(name.getText(),ID.getText() , url));  
 Decline(event);  
 }  
 @FXML  
 public void Decline(MouseEvent event){//не добавлять его  
 ((Stage)path.getScene().getWindow()).close();  
 }  
  
 @Override  
 public void initialize(URL url, ResourceBundle resourceBundle) {  
 }  
  
}

DialogController.java:

import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.fxml.Initializable;  
import javafx.scene.control.\*;  
import javafx.scene.input.MouseEvent;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
  
import java.net.URL;  
import java.util.ResourceBundle;  
  
public class DialogController implements Initializable {  
 @FXML  
 private VBox l1;  
  
 @FXML  
 private RadioButton r1;  
 @FXML  
 private RadioButton r2;  
 @FXML  
 private RadioButton r3;  
 @FXML  
 private RadioButton r4;  
 @FXML  
 private RadioButton r5;  
  
 @FXML  
 private TextField t1;  
  
 @FXML  
 private CheckBox doesItExist;  
  
 @FXML  
 private Button save;  
 @FXML  
 private Button decline;  
  
 Point p = new Point();  
  
 @Override  
 public void initialize(URL url, ResourceBundle rb){  
  
 r1.setOnMouseClicked(new EventHandler<MouseEvent>() {//выбор типа точки, были кое-какие планы на это, но в реальности не используется. Возможно потом использоваться будет.  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 p.type = 1;  
 }  
 });  
  
  
 r2.setOnMouseClicked(new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 p.type = 2;  
 }  
 });  
  
  
 r3.setOnMouseClicked(new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 p.type = 3;  
 }  
 });  
  
  
 r4.setOnMouseClicked(new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {   
 p.type = 4;  
 }  
 });  
 }  
 @FXML  
 public void Save(MouseEvent event){//добавление новой либо уже существующей точки  
 if(!(doesItExist.isSelected())) {  
 p.setName(t1.getText());  
 HelloApplication.*hc*.addPoint(p);  
 ((Stage) save.getScene().getWindow()).close();  
 } else {  
 HelloApplication.*hc*.addNewPosToExistPoint(t1.getText());  
 ((Stage) save.getScene().getWindow()).close();  
 }  
 }  
 @FXML  
 public void Decline(MouseEvent event){//отмена добавления точки  
 ((Stage)save.getScene().getWindow()).close();  
 }  
}

Некоторые моменты из файла pom.xml:

Плагин, предназначенный для сборки проекта в самостоятельный jar-файл

<plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-shade-plugin</artifactId>  
 <version>3.3.0</version>  
 <executions>  
 <execution>  
 <phase>package</phase>  
 <goals>  
 <goal>shade</goal>  
 </goals>  
 <configuration>  
 <transformers>  
 <transformer implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">  
 <mainClass>com.example.demo.Starter</mainClass>  
 </transformer>  
 </transformers>  
 </configuration>  
 </execution>  
 </executions>  
</plugin>

Зависимость от библиотеки Jgrapht

<dependency>  
 <groupId>org.jgrapht</groupId>  
 <artifactId>jgrapht-core</artifactId>  
 <version>1.5.1</version>  
</dependency>

fxml и pom файлы можно посмотреть в Приложении 1.

# Код android-приложения

MainActivity:

Как следует из названия, очень важный файл. В принципе, аналогичен HelloController

import androidx.appcompat.app.AppCompatActivity;  
  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.graphics.Canvas;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.graphics.Rect;  
import android.graphics.drawable.BitmapDrawable;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.View;  
import android.view.ViewTreeObserver;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
  
import org.jgrapht.Graph;  
import org.jgrapht.GraphPath;  
import org.jgrapht.alg.shortestpath.DijkstraShortestPath;  
import org.jgrapht.graph.DefaultWeightedEdge;  
import org.jgrapht.graph.SimpleWeightedGraph;  
  
import com.example.demo.FragmentInfo;  
import com.example.demo.Point;  
  
import java.io.ObjectInputStream;  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.LinkedHashSet;  
  
public class MainActivity extends AppCompatActivity {  
  
 Graph<Point, DefaultWeightedEdge> g = new SimpleWeightedGraph<>(DefaultWeightedEdge.class);  
 private HashMap<String, Point> hashMap;  
 private HashMap<String, Bitmap> fragments = new HashMap<>();//тут все в принципе, аналогично программе  
 private Integer firstID = 9999;//первый ID явялется самым маленьким ID. Поиск самого маленького ID держится на утверждении, что оно будет точно меньше 9999  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {//приложение запущено  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 try {  
 ObjectInputStream objectInputStream = new ObjectInputStream(this.getResources().openRawResource(R.raw.*toandroid*));  
 String graph = (String) objectInputStream.readObject();  
 objectInputStream.close();  
 objectInputStream = new ObjectInputStream(this.getResources().openRawResource(R.raw.*toandroidpoints*));  
 hashMap = (HashMap<String, Point>) objectInputStream.readObject();//загрузка сериализованных на пк объектов  
 objectInputStream.close();  
 objectInputStream = new ObjectInputStream(this.getResources().openRawResource(R.raw.*toandroidfragments*));  
 ArrayList<FragmentInfo> set = (ArrayList<FragmentInfo>) objectInputStream.readObject();  
 for (FragmentInfo fragment:  
 set) {  
 String name = "p" + fragment.fileName.substring(0, fragment.fileName.length() - 4).replace(".", "");  
 Bitmap bmp = BitmapFactory.*decodeResource*(getResources(), getResources().getIdentifier(name, "drawable", this.getPackageName()));//создание карты Имя-Фрагмент карты  
 if(bmp != null)//если что-то пошло не так и нет нужного изображения фрагмента, то нам и добавлять нечего  
 fragments.put(fragment.ID, bmp);  
 if(Integer.*parseInt*(fragment.ID) < firstID)//небольшая неопределенность с типом ID  
 firstID = Integer.*parseInt*(fragment.ID);  
 }  
 objectInputStream.close();  
 int i = 0;  
 int j = 0;  
 do {//парсер строки, в которую превратился граф for (; !(Character.*isDigit*(graph.charAt(i))); i++);  
 for (j = i; (Character.*isDigit*(graph.charAt(j))); j++);  
 String str = graph.substring(i, j);  
 try {  
 g.addVertex(hashMap.get(graph.substring(i, j)));  
 } catch (Exception e){}  
 i = j + 1;  
 } while (graph.charAt(j) != ']');  
 do {  
 for (; !(Character.*isDigit*(graph.charAt(i))); i++);  
 for (j = i; (Character.*isDigit*(graph.charAt(j))); j++);  
 Point first = (hashMap.get(graph.substring(i, j)));  
 i = j + 1;  
 for (; !(Character.*isDigit*(graph.charAt(i))); i++);  
 for (j = i; (Character.*isDigit*(graph.charAt(j))); j++);  
 Point second = (hashMap.get(graph.substring(i, j)));  
 i = j + 1;  
 if(first != null && second != null) {  
 DefaultWeightedEdge dwe = new DefaultWeightedEdge();  
 g.addEdge(first, second, dwe);  
 int ID = -1;  
 for (Integer fID :  
 first.positions.keySet()) {  
 for (Integer sID :  
 second.positions.keySet()) {  
 if (fID.equals(sID))  
 ID = fID;  
 }  
 }  
 if (ID != -1)//не всегда входной файл идеален и не имеет каких-либо проблем, поэтому тут мы пытаемся допилить его до того самого идеала  
 g.setEdgeWeight(dwe, Math.*sqrt*((first.getPos(ID).getFirst() - second.getPos(ID).getFirst()) \* (first.getPos(ID).getFirst() - second.getPos(ID).getFirst()) + (first.getPos(ID).getSecond() - second.getPos(ID).getSecond()) \* (first.getPos(ID).getSecond() - second.getPos(ID).getSecond())));  
 }  
 } while (graph.charAt(i) != ']');  
 String str = g.toString();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
  
 }  
  
 @Override  
 public void onStart() {  
 super.onStart();  
 ImageView imageView = findViewById(R.id.*imageView*);  
 imageView.getViewTreeObserver().addOnGlobalLayoutListener(new ViewTreeObserver.OnGlobalLayoutListener() {  
 @Override  
 public void onGlobalLayout() {  
 showFragment();//когда будут получены размеры imageView, вывести первый фрагмент и его точки  
 if (Build.VERSION.*SDK\_INT* < Build.VERSION\_CODES.*JELLY\_BEAN*) {  
 imageView.getViewTreeObserver().removeGlobalOnLayoutListener(this);  
 } else {  
 imageView.getViewTreeObserver().removeOnGlobalLayoutListener(this);  
 }  
 }  
 });  
 }  
  
 int flag = 1;  
 LinkedHashSet<Integer> fragmentsList = new LinkedHashSet<>();  
 public void OnClickWay(View view){//обработчик события нажатия на кнопку  
 if(flag == 1){//если нажимают впервые   
 FindPath();//найти путь  
 showDeWay();//показать путь на первом фрагменте пути  
 } else  
 if(flag == 2){//если путь пройден, скрыть его с экрана, обнулить состояние кнопки  
 ImageView imageView = findViewById(R.id.*imageView*);  
 imageView.setImageBitmap(fragments.get(String.*valueOf*(firstID)));  
 ((Button)view).setText("Найти путь");  
 flag = 1;  
 } else  
 if(flag == 3){  
 showDeWay();//если есть другие фрагменты, по которым проходит путь, показать следующий   
 }  
 }  
  
 ArrayList<TextView> arrayListTextViev = new ArrayList<>();  
 public void showFragment(){//вывод на экран фрагмента и точек, лежащих на нем  
 ImageView imageView = findViewById(R.id.*imageView*);  
 for (TextView text:  
 arrayListTextViev) {  
 ((RelativeLayout) findViewById(R.id.*relativeLayout*)).removeView(text);  
 }  
 arrayListTextViev.clear();  
 for (Point p:  
 hashMap.values()) {  
 if(p.getPos(firstID) != null) {  
 RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.*WRAP\_CONTENT*, RelativeLayout.LayoutParams.*WRAP\_CONTENT*);  
 params.leftMargin = (int) (p.getPos(firstID).getFirst() \* imageView.getWidth());  
 params.topMargin = (int) (p.getPos(firstID).getSecond() \* imageView.getHeight());  
 TextView textView = new TextView(MainActivity.this);  
 if(p.getName().length() > 3)  
 textView.setText(p.getName());  
 textView.setTextColor(Color.*RED*);  
 ((RelativeLayout) findViewById(R.id.*relativeLayout*)).addView(textView, params);  
 arrayListTextViev.add(textView);  
 }  
 }  
 Bitmap bmp = fragments.get(String.*valueOf*(firstID));  
 imageView.setImageBitmap(bmp);  
 }  
  
 GraphPath<Point, DefaultWeightedEdge> graphPathList = null;  
 public void FindPath() {//поиск пути. Аналогичен программе на пк  
 Point from = findPoint((String) ((TextView) findViewById(R.id.*cabfrom*)).getText().toString());  
 Point to = findPoint((String) ((TextView) findViewById(R.id.*cabto*)).getText().toString());  
 DijkstraShortestPath dijkstraShortestPath = new DijkstraShortestPath(g);  
 graphPathList = dijkstraShortestPath.getPath(from, to);  
 for (DefaultWeightedEdge defaultEdge :  
 graphPathList.getEdgeList()) {  
 Point source = g.getEdgeSource(defaultEdge);  
 Point target = g.getEdgeTarget(defaultEdge);  
 if(source.positions.size() == 1) fragmentsList.add((Integer) (source.positions.keySet().toArray())[0]);//добавление фрагментов, по которым идет путь  
 if(target.positions.size() == 1) fragmentsList.add((Integer) (source.positions.keySet().toArray())[0]);  
 }  
 System.*out*.println(graphPathList.getWeight());  
 }  
  
 public void showDeWay(){//отображение пути  
 firstID = fragmentsList.iterator().next();//получить первый фрагмент по которому проходит путь  
 fragmentsList.remove(firstID); //удалить его из пути  
 showFragment();//показать этот фрагмент  
 ImageView imageView = findViewById(R.id.*imageView*);  
 Bitmap bitmap = ((BitmapDrawable) imageView.getDrawable()).getBitmap();  
 Bitmap tempBitmap = Bitmap.*createBitmap*(bitmap.getWidth(), bitmap.getHeight(), Bitmap.Config.*RGB\_565*);  
 Canvas canvas = new Canvas(tempBitmap);  
 canvas.drawBitmap(bitmap, new Rect(0, 0,bitmap.getWidth(), bitmap.getHeight()), new Rect(0, 0,bitmap.getWidth(), bitmap.getHeight()), null);  
 Paint paint = new Paint();  
 paint.setColor(Color.*RED*);  
 paint.setStrokeWidth(10);  
  
 for (DefaultWeightedEdge defaultEdge :  
 graphPathList.getEdgeList()) {  
 Point source = g.getEdgeSource(defaultEdge);  
 Point target = g.getEdgeTarget(defaultEdge);  
  
 if ((source.getPos(firstID) != null) && (target.getPos(firstID) != null)) {  
 canvas.drawLine((float) (source.getPos(firstID).getFirst()\*bitmap.getWidth()),(float) (source.getPos(firstID).getSecond()\*bitmap.getHeight()), (float) (target.getPos(firstID).getFirst()\*bitmap.getWidth()),(float) (target.getPos(firstID).getSecond()\*bitmap.getHeight()), paint); //нарисовать линию, между точками, если они принадлежат этому фрагменту  
 }  
 if(fragmentsList.size() > 0) {  
 flag = 3; //если еще остались фрагменты  
 ((Button)findViewById(R.id.*PathWay*)).setText("Продолжить путь");  
 } else {  
 flag = 2; //если фрагментов больше нет  
 ((Button)findViewById(R.id.*PathWay*)).setText("Скрыть путь");  
 }  
 }  
 imageView.setImageBitmap(tempBitmap);  
  
  
  
 }  
  
 private Point findPoint(String name) {//поиск точки  
 for (Point point :  
 hashMap.values()) {  
 if (point.getName().equals(name))  
 return point;  
 }  
 return null;  
 }  
  
}

Подключение библиотеки графа в зависимостях в файле build.gradle:

implementation 'org.jgrapht:jgrapht-core:1.5.1'

FragmentInfo и Point аналогичны тем, которые были на пк. xml и build.gradle файлы и прочее можно посмотреть в Приложении 2.

# Источники:

[https://habr.com](https://habr.com/ru)/

<https://stackoverflow.com/>

<https://qna.habr.com/>

<https://www.cyberforum.ru/>

<https://docs.oracle.com>/

<http://developer.alexanderklimov.ru/android/>

<https://developer.android.com/>

<https://startandroid.ru/>

<https://docs.microsoft.com/>

# Приложение 1:

dialog.xml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.CheckBox?>  
<?import javafx.scene.control.RadioButton?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.Pane?>  
<?import javafx.scene.layout.VBox?>  
<?import javafx.scene.text.Text?>  
  
<VBox fx:id="l1" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="196.0" prefWidth="235.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.demo.DialogController">  
 <children>  
 <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Выберите тип точки" />  
 <RadioButton fx:id="r5" mnemonicParsing="false" text="Коридор" />  
 <RadioButton fx:id="r1" mnemonicParsing="false" text="Кабинет" />  
 <RadioButton fx:id="r2" mnemonicParsing="false" text="Лестница" />  
 <RadioButton fx:id="r3" mnemonicParsing="false" text="Туалет" />  
 <RadioButton fx:id="r4" mnemonicParsing="false" text="Лифт" />  
 <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Введите имя точки" />  
 <TextField fx:id="t1" />  
 <CheckBox fx:id="doesItExist" mnemonicParsing="false" text="Эта точка уже существует" />  
 <Pane prefHeight="200.0" prefWidth="200.0">  
 <children>  
 <Button fx:id="save" layoutX="151.0" layoutY="2.0" mnemonicParsing="false" onMouseClicked="#Save" prefHeight="25.0" prefWidth="84.0" text="Сохранить" />  
 <Button fx:id="decline" layoutY="2.0" mnemonicParsing="false" onMouseClicked="#Decline" text="Отменить" />  
 </children>  
 </Pane>  
 </children>  
</VBox>

dialogFragment.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.HBox?>  
<?import javafx.scene.layout.Pane?>  
<?import javafx.scene.layout.VBox?>  
<?import javafx.scene.text.Text?>  
  
<VBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="159.0" prefWidth="312.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.demo.dialogFragmentController">  
 <children>  
 <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Введите название фрагмента " />  
 <TextField fx:id="name" />  
 <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Введите ID фрагмента" />  
 <TextField fx:id="ID" />  
 <Text fx:id="path" strokeType="OUTSIDE" strokeWidth="0.0" text="Выберите файл" />  
 <Button mnemonicParsing="false" onMouseClicked="#ChooseFile" text="Выбрать файл" />  
 <Pane prefHeight="200.0" prefWidth="200.0">  
 <children>  
 <Button alignment="BOTTOM\_LEFT" mnemonicParsing="false" onMouseClicked="#Decline" text="Отменить" />  
 <Button layoutX="249.0" mnemonicParsing="false" onMouseClicked="#Accept" text="Принять" />  
 </children>  
 </Pane>  
 <HBox prefHeight="100.0" prefWidth="200.0" />  
 </children>  
</VBox>

hello-view.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.ScrollPane?>  
<?import javafx.scene.control.SplitPane?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.layout.VBox?>  
  
<SplitPane dividerPositions="0.17948717948717946" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-Infinity" minWidth="-Infinity" prefHeight="800.0" prefWidth="1000.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.demo.HelloController">  
 <items>  
 <VBox prefHeight="200.0" prefWidth="100.0">  
 <children>  
 <Button mnemonicParsing="false" onMouseClicked="#LoadFromFile" prefHeight="26.0" prefWidth="192.0" text="Загрузить" />  
 <Button mnemonicParsing="false" onMouseClicked="#SaveToFile" prefHeight="26.0" prefWidth="201.0" text="Сохранить" />  
 <Label prefHeight="18.0" prefWidth="176.0" text="Этажи" />  
 <ScrollPane prefHeight="200.0" prefWidth="200.0">  
 <content>  
 <VBox fx:id="floors" prefHeight="201.0" prefWidth="158.0" />  
 </content>  
 </ScrollPane>  
 <Button mnemonicParsing="false" onMouseClicked="#OpenFloor" prefHeight="26.0" prefWidth="191.0" text="Добавить" />  
 <Button mnemonicParsing="false" onMouseClicked="#RemoveFloor" prefHeight="26.0" prefWidth="183.0" text="Удалить" />  
 <TextField fx:id="cabfrom" />  
 <TextField fx:id="cabto" />  
 <Button fx:id="path" mnemonicParsing="false" onMouseClicked="#FindPath" prefHeight="26.0" prefWidth="184.0" text="Найти маршрут" />  
 <Button mnemonicParsing="false" onMouseClicked="#RemovePoint" prefHeight="26.0" prefWidth="192.0" text="Удалить точку и ее ребра" />  
 <TextField text="7" fx:id="textSize"/>  
 <Button mnemonicParsing="false" prefHeight="26.0" prefWidth="176.0" text="Изменить размер текста" onMouseClicked="#ChangeTextSize"/>  
 </children>  
 </VBox>  
 <AnchorPane fx:id="anchorPane" minHeight="0.0" minWidth="0.0" prefHeight="640.0" prefWidth="627.0" />  
 </items>  
</SplitPane>

pom.xml:

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.example</groupId>  
 <artifactId>demo</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <name>demo</name>  
  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <junit.version>5.8.1</junit.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-controls</artifactId>  
 <version>17.0.1</version>  
 </dependency>  
 <dependency>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-fxml</artifactId>  
 <version>17.0.1</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>${junit.version}</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-engine</artifactId>  
 <version>${junit.version}</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.jgrapht</groupId>  
 <artifactId>jgrapht-core</artifactId>  
 <version>1.5.1</version>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <resources>  
 <resource>  
 <directory>src/main</directory>  
 <includes>  
 <include>\*\*/\*.fxml</include>  
 </includes>  
 </resource>  
 </resources>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-shade-plugin</artifactId>  
 <version>3.3.0</version>  
 <executions>  
 <execution>  
 <phase>package</phase>  
 <goals>  
 <goal>shade</goal>  
 </goals>  
 <configuration>  
 <transformers>  
 <transformer implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">  
 <mainClass>com.example.demo.Starter</mainClass>  
 </transformer>  
 </transformers>  
 </configuration>  
 </execution>  
 </executions>  
 </plugin>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.8.1</version>  
 <configuration>  
 <source>17</source>  
 <target>17</target>  
 </configuration>  
 </plugin>  
 <plugin>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-maven-plugin</artifactId>  
 <version>0.0.8</version>  
 <executions>  
 <execution>  
 <!-- Default configuration for running with: mvn clean javafx:run -->  
 <id>default-cli</id>  
 <configuration>  
 <mainClass>com.example.demo/com.example.demo.HelloApplication</mainClass>  
 <launcher>app</launcher>  
 <jlinkZipName>app</jlinkZipName>  
 <jlinkImageName>app</jlinkImageName>  
 <noManPages>true</noManPages>  
 <stripDebug>true</stripDebug>  
 <noHeaderFiles>true</noHeaderFiles>  
 </configuration>  
 </execution>  
 </executions>  
 </plugin>  
 </plugins>  
 </build>  
</project>

# Приложение 2:

activity\_main.xml:

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <Button  
 android:id="@+id/PathWay"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:onClick="OnClickWay"  
 android:text="Найти путь"  
 android:textSize="24sp" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Кабинет начала"  
 android:textSize="24sp" />  
  
 <EditText  
 android:id="@+id/cabfrom"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Кабинет назначения"  
 android:textSize="24sp" />  
  
 <EditText  
 android:id="@+id/cabto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName" />  
  
 <RelativeLayout  
 android:id="@+id/relativeLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:scaleType="fitXY"  
 app:srcCompat="@drawable/p12" />  
  
 </RelativeLayout>  
  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>

build.gradle

plugins **{** id 'com.android.application'  
**}**android **{** compileSdk 31  
  
 defaultConfig **{** applicationId "com.example.bonchnav"  
 minSdk 21  
 targetSdk 31  
 versionCode 1  
 versionName "1.0"  
  
 testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
 **}** buildTypes **{** release **{** minifyEnabled false  
 proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
 **}  
 }** compileOptions **{** sourceCompatibility JavaVersion.*VERSION\_1\_8* targetCompatibility JavaVersion.*VERSION\_1\_8* **}  
}**dependencies **{** implementation 'org.jgrapht:jgrapht-core:1.5.1'  
 implementation 'androidx.appcompat:appcompat:1.4.1'  
 implementation 'com.google.android.material:material:1.6.0'  
 implementation 'androidx.constraintlayout:constraintlayout:2.1.3'  
 testImplementation 'junit:junit:4.+'  
 androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
 androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
**}**