Федеральное государственное автономное образовательное учреждение

высшего образования

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа «Киберфизические системы и управление»

**Отчет №13**

по дисциплине «Системный подход к разработке программного обеспечения»

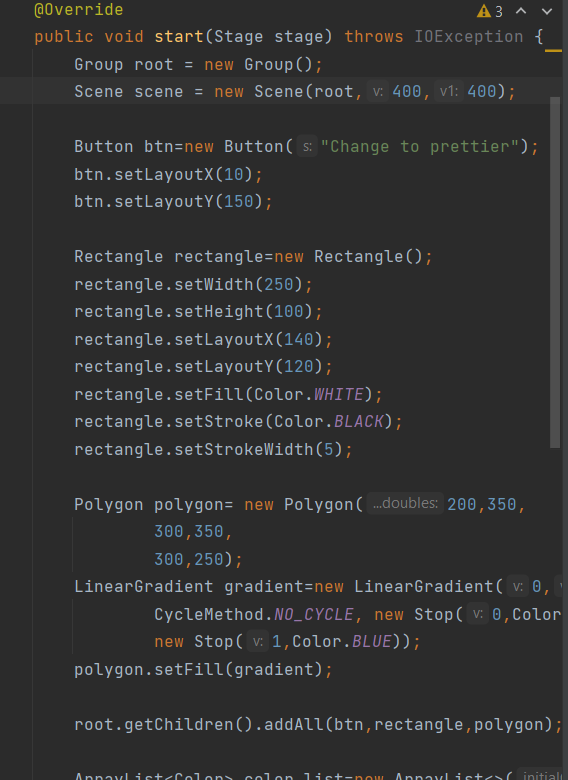
|  |  |  |
| --- | --- | --- |
| Выполнил:  студент гр. 3530902/00201 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Позолотин О. В. |
|  | <*подпись*> |  |
|  |  |  |
|  |  |  |
| Руководитель:  Кандидат т.н. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Нестеров С. А. |
|  | <*подпись*> |  |

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022 г.

Санкт-Петербург

2022

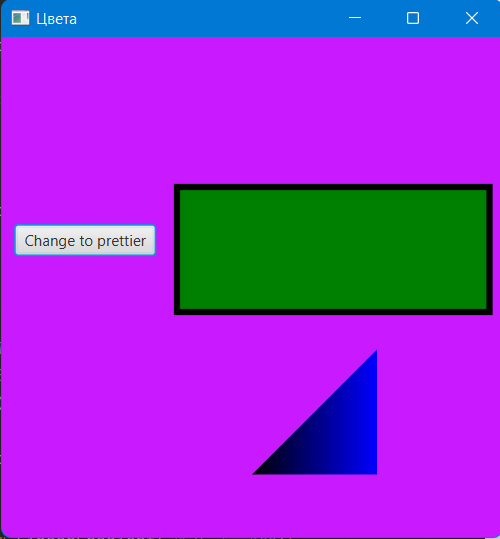
**Задания по уроку 9.2**



Изображение выглядит как текст

Автоматически созданное описание

Был написан код, используюший кастомные цвета. Добавлена кнопка, при нажатии на которую меняется цвет сцены, как показано на рисунке ниже.



Упражнение 2.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

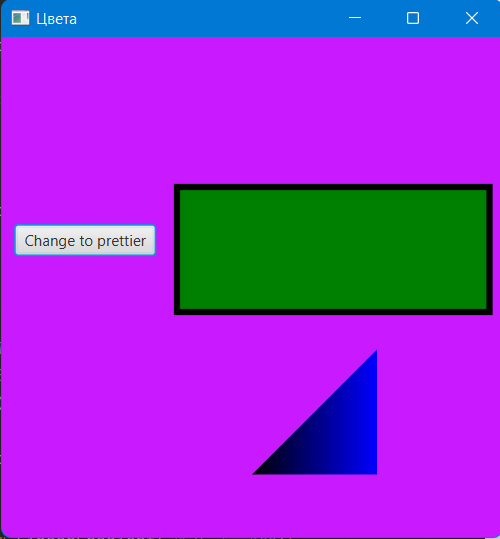
На сцену добавлена фигура – прямоугольник. Применены методы:

SetWidth(), SetHeight() – установка размеров прямоугольника

SetFill() – установка цвета заполнения

SetStroke() и SetStrokeWidth() – установка цвета контура и его ширины

Теперь программа выглядит так:



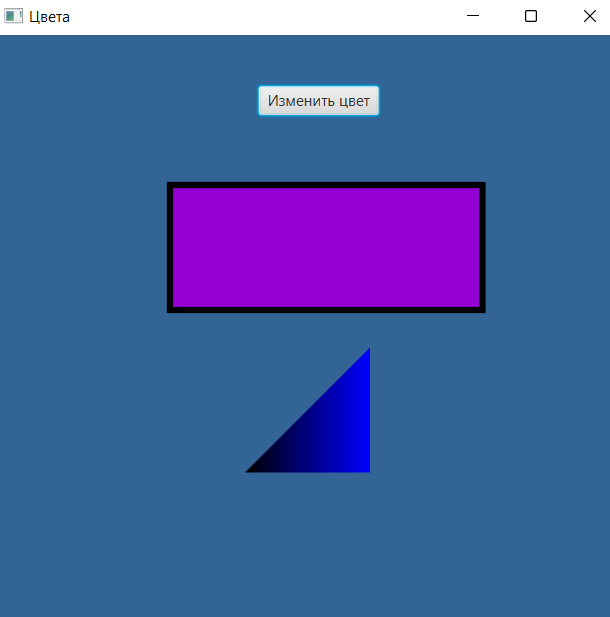
Упражнение 3.

Написан код, который добавляет на сцену треугольник:

Изображение выглядит как текст

Автоматически созданное описание

При нажатии на кнопку треугольник заполняется градиентным цветом.



Упражнения по уроку 9.3

Упражнение 1.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

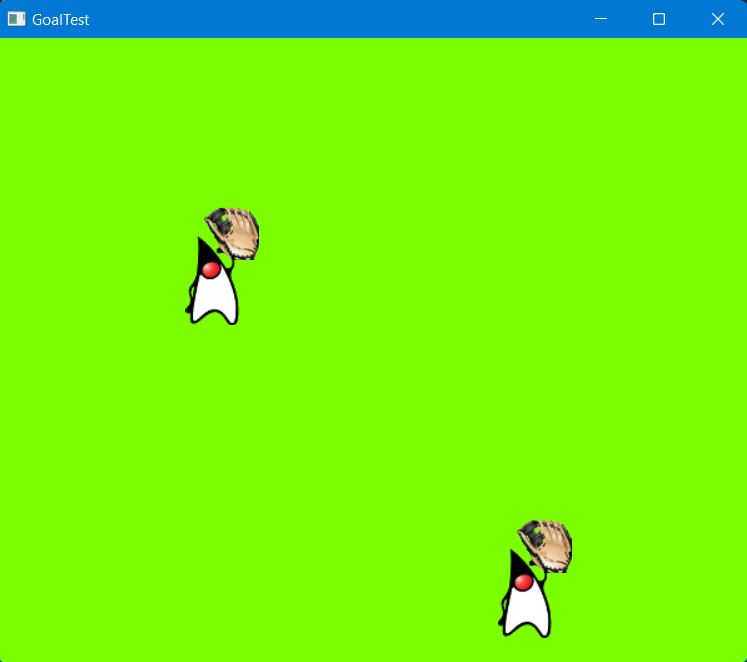
Был изменен конструктор, добавлены поля, и созданы экземпляры класса Goal в GoalTest. При запуске программы на сцене появляется 2 изображения Дюка.

Упражнение 2.

Изображение выглядит как текст

Автоматически созданное описание

Конструктор изменен так, чтоб на руке Дюка оказалась перчатка.



Упражнение 3.

Изображение выглядит как текст

Автоматически созданное описание

Поле AudioClip добавлено.

Упражнение 4.

Изображение выглядит как текст

Автоматически созданное описание

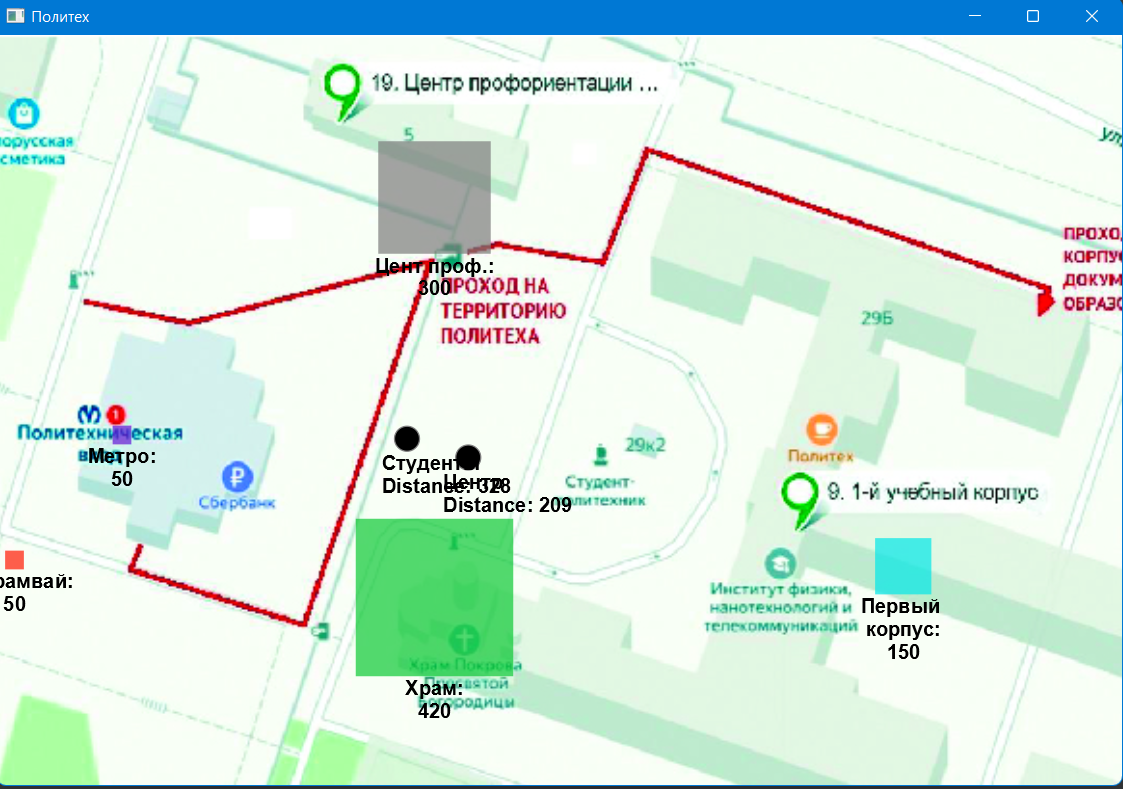
Так же в класс GoalTest был добавлен код, помогающий отслеживать положение мыши на сцене:

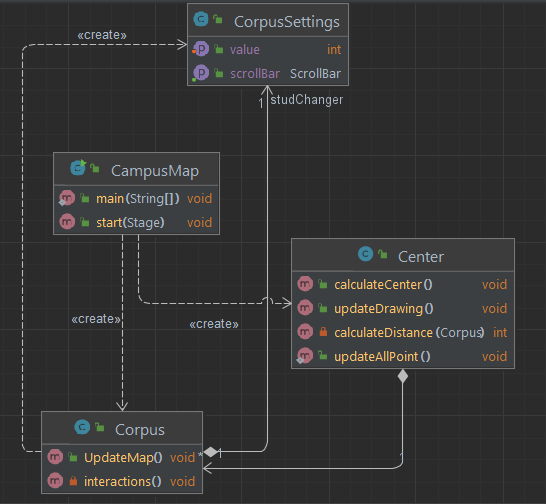
Изображение выглядит как текст

Автоматически созданное описание

P.S. курсор на скрине пропадает.

Практика 9





Написана программа, позволяющая определить центр между студентами и центр, обозначающий «центр масс» кампуса. Все изменения, проводимые для мест, влияют на расположение центров.

Код программы представлен ниже.

**Класс Center**

package campusmap;  
  
import java.util.ArrayList;  
import javafx.scene.Node;  
import javafx.scene.paint.Color;  
import javafx.scene.shape.Circle;  
import javafx.scene.text.Font;  
import javafx.scene.text.FontWeight;  
import javafx.scene.text.Text;  
  
public class Center {  
 private static final ArrayList<Center> *allPoints* = new ArrayList();  
 Corpus[] corps;  
 public int xLocation;  
 public int yLocation;  
 public String text;  
 private int distance;  
 public Circle point = new Circle();  
 public Color fillColor;  
 public Color outlineColor;  
 public Text display;  
  
 //конструктор  
 public Center(String s, Corpus... dorms) {  
 fillColor = Color.*BLACK*;  
 outlineColor = Color.*GRAY*;  
 display = new Text();  
 text = s + "\nDistance: ";  
  
 corps = dorms;  
 display.setFill(Color.*BLACK*);  
 display.setFont(Font.*font*("Default", FontWeight.*BOLD*, 16.0D));  
 CampusMap.*root*.getChildren().addAll(new Node[]{point, display});  
 *allPoints*.add(this);  
 calculateCenter();  
 updateDrawing();  
 }  
  
 public void updateDrawing() {  
 point.setCenterX( xLocation);  
 point.setCenterY( yLocation);  
 point.setRadius(10.0D);  
 point.setFill(fillColor);  
 point.setStroke(outlineColor);  
 display.setText(text + distance);  
 display.setX( (xLocation - 20));  
 display.setY( yLocation + point.getRadius() + 15.0D);  
 }  
 public void calculateCenter() {  
 int stud = 0;  
 int xWeight = 0;  
 int yWeight = 0;  
 Corpus[] corp = corps;  
  
 for (int i = 0; i < corp.length; ++i) {  
 Corpus build\_ = corp[i];  
 xWeight += build\_.xLocation \* build\_.students;  
 yWeight += build\_.yLocation \* build\_.students;  
 stud += build\_.students;  
 }  
  
 xLocation = xWeight / stud;  
 yLocation = yWeight / stud;  
  
 distance = calculateDistance(corps[0]);  
 }  
  
 private int calculateDistance(Corpus build) {  
 return (int) Math.*sqrt*(Math.*pow*((double) (build.xLocation - xLocation), 2.0D) + Math.*pow*((double) (build.yLocation - yLocation), 2.0D));  
 }  
  
 public static final void updateAllPoint() {  
 for(Center point:*allPoints*) {  
 point.calculateCenter();  
 point.updateDrawing();  
 point.point.toFront();  
 point.display.toFront();  
 }  
 }  
}

**Класс Corpus**

package campusmap;  
  
import java.awt.geom.Point2D.Double;  
import javafx.scene.Node;  
import javafx.scene.paint.Color;  
import javafx.scene.shape.Rectangle;  
import javafx.scene.text.Font;  
import javafx.scene.text.FontWeight;  
import javafx.scene.text.Text;  
import javafx.scene.text.TextAlignment;  
  
public class Corpus {  
 public int xLocation;  
 public int yLocation;  
 public int students;  
 public String text;  
 private Double iconGrabPosition = new Double(0.0D, 0.0D);  
 private Double mouseGrabPosition = new Double(0.0D, 0.0D);  
 private Double mouseReleasePosition = new Double(0.0D, 0.0D);  
 CorpusSettings studChanger;  
 public Rectangle buildingRectangle = new Rectangle();  
 public Color fillColor;  
 public Color outlineColor;  
 public Text display;  
   
 //конструктор класса  
 public Corpus(int width, int height, int students\_quantity, String name, Color color) {  
 xLocation = width;  
 yLocation = height;  
 students = students\_quantity;  
 text = name;  
 fillColor = color;  
  
 display = new Text();  
 display.setFill(Color.*BLACK*);  
 display.setFont(Font.*font*("Default", FontWeight.*BOLD*, 16.0D));  
  
 CampusMap.*root*.getChildren().addAll(new Node[]{buildingRectangle, display});  
 interactions();  
 UpdateMap();  
 }  
 //добавление и расстановка   
 public void UpdateMap() {  
 buildingRectangle.setWidth(150.0D \* (1.0D \* (double)students / 500.0D));  
 buildingRectangle.setHeight(150.0D \* (1.0D \* (double)students / 500.0D));  
 buildingRectangle.setLayoutX((double)xLocation - buildingRectangle.getWidth() / 2.0D);  
 buildingRectangle.setLayoutY((double)yLocation - buildingRectangle.getHeight() / 2.0D);  
 buildingRectangle.setFill(fillColor);  
 buildingRectangle.setStroke(outlineColor);  
  
 display.setText(text + ":\n" + students);  
  
 display.setLayoutX(buildingRectangle.getLayoutX() - 40.0D);  
 display.setLayoutY(buildingRectangle.getLayoutY() + buildingRectangle.getHeight() + 15.0D);  
 display.setTextAlignment(TextAlignment.*CENTER*);  
 display.setWrappingWidth(buildingRectangle.getWidth() + 80.0D);  
 }  
 private void interactions() {  
 buildingRectangle.setOnMousePressed((me) -> {  
 mouseGrabPosition.x = me.getSceneX();  
 mouseGrabPosition.y = me.getSceneY();  
 iconGrabPosition.x = xLocation;  
 iconGrabPosition.y = yLocation;  
 buildingRectangle.toFront();  
 });  
 buildingRectangle.setOnMouseDragged((me) -> {  
 mouseReleasePosition.x = me.getSceneX();  
 mouseReleasePosition.y = me.getSceneY();  
 xLocation = (int)(me.getSceneX() - mouseGrabPosition.x + iconGrabPosition.x);  
 yLocation = (int)(me.getSceneY() - mouseGrabPosition.y + iconGrabPosition.y);  
 UpdateMap();  
 Center.*updateAllPoint*();  
 });  
  
 display.setOnMousePressed((me) -> {  
 if (stud Changer == null) {  
 mouseGrabPosition.x = me.getSceneX();  
 iconGrabPosition.x = (double)students;  
 studChanger = new CorpusSettings(students);  
 studChanger.getScrollBar().setLayoutX(buildingRectangle.getLayoutX());  
 studChanger.getScrollBar().setLayoutY(buildingRectangle.getLayoutY() + buildingRectangle.getHeight() + 40.0D);  
 CampusMap.*root*.getChildren().add(studChanger.getScrollBar());  
 }  
  
 });  
  
 display.setOnMouseDragged((me) -> {  
 mouseReleasePosition.x = me.getSceneX();  
 int newValue = (int)(2.0D \* me.getSceneX() - 2.0D \* mouseGrabPosition.x + iconGrabPosition.x);  
 if (newValue < 50) {  
 newValue = 50;  
 } else if (newValue > 600) {  
 newValue = 600;  
 }  
  
 studChanger.setValue(newValue);  
 students = newValue;  
 UpdateMap();  
 Center.*updateAllPoint*();  
 studChanger.getScrollBar().setLayoutY(buildingRectangle.getLayoutY() + buildingRectangle.getHeight() + 40.0D);  
 });  
  
 display.setOnMouseReleased((me) -> {  
 studChanger.getScrollBar().toBack();  
 if (studChanger != null) {  
 CampusMap.*root*.getChildren().remove(studChanger.getScrollBar());  
 studChanger = null;  
 }  
  
 });  
 }  
   
}

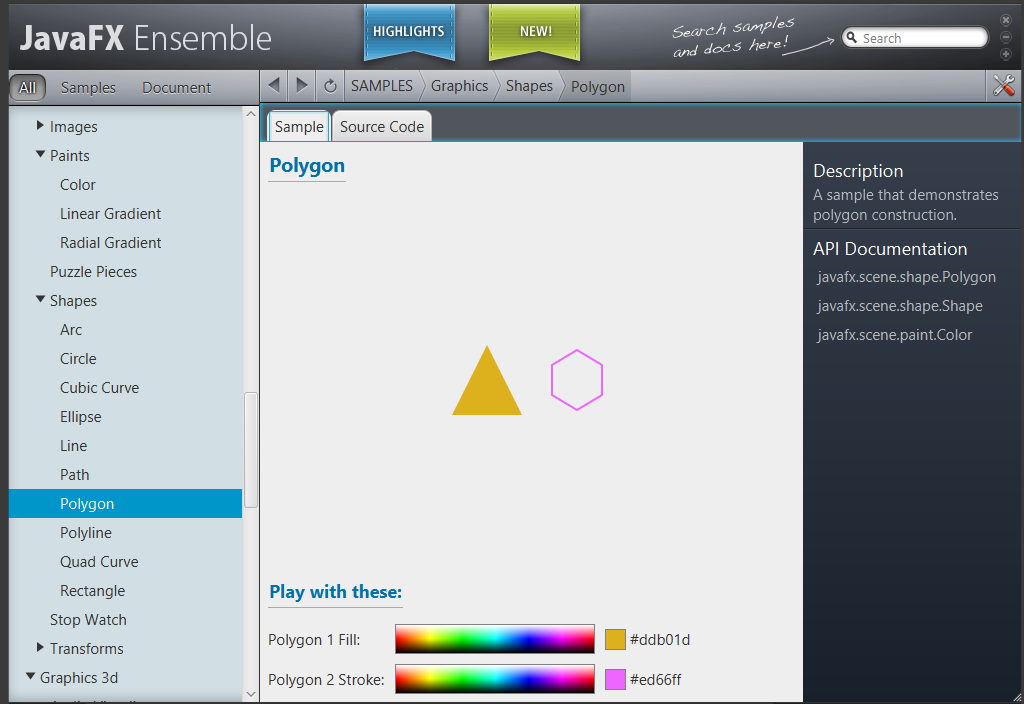
**Класс CampusMap**

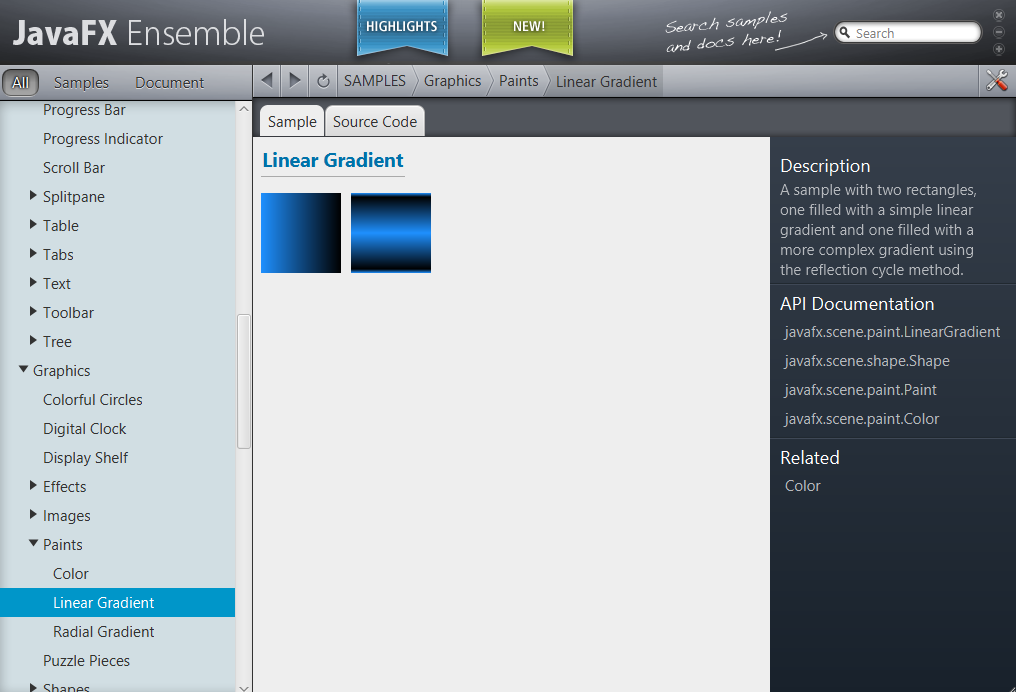
package campusmap;  
  
import javafx.application.Application;  
import javafx.scene.Group;  
import javafx.scene.Scene;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.paint.Color;  
import javafx.stage.Stage;  
  
public class CampusMap extends Application {  
 public static Group *root* = new Group();  
@Override  
 public void start(Stage primaryStage) {  
 Scene scene = new Scene(*root*, 900, 850, Color.*WHITE*);  
 primaryStage.setTitle("Политех");  
 primaryStage.setScene(scene);  
 primaryStage.show();  
  
 Image background = new Image(this.getClass().getResourceAsStream("maps/campus\_map.jpg"));  
 ImageView backgroundView = new ImageView(background);  
 backgroundView.setPreserveRatio(true);  
 backgroundView.setFitWidth(1024);  
  
 *root*.getChildren().add(backgroundView);  
 backgroundView.toBack();  
  
 Corpus main\_building = new Corpus(400, 190, 300, "Главное здание", Color.*rgb*(88, 255, 182, 0.7D));  
 Corpus Corpus1 = new Corpus(725, 425, 150, "Корпус 1", Color.*rgb*(250, 130, 23, 0.7D));  
 Corpus Corpus2 = new Corpus(725, 250, 50, "Корпус 2", Color.*rgb*(0, 0, 255, 0.7D));  
 Corpus tram = new Corpus(140, 420, 50, "Трамвай", Color.*rgb*(255, 255, 0, 0.7D));  
 Corpus stadium = new Corpus(350, 700,420, "Стадион", Color.*rgb*(255, 255, 255, 0.7D));  
  
 Center centerPoint = new Center("Центр", new Corpus[]{main\_building, Corpus1, Corpus2,tram , stadium});  
 Center centerPoint1 = new Center("Студенты", new Corpus[]{tram, stadium, main\_building});  
 }  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
   
}

**Класс СorpusSettings**

package campusmap;  
import javafx.scene.control.ScrollBar;  
  
public class CorpusSettings {  
 private final ScrollBar xscrollBar = new ScrollBar();  
 public CorpusSettings(int value) {  
 xscrollBar.setMin(50.0D);  
 xscrollBar.setMax(600.0D);  
 setValue(value);  
 xscrollBar.setMinSize(100.0D, 15.0D);  
 xscrollBar.setPrefSize(100.0D, 15.0D);  
 xscrollBar.setMaxSize(100.0D, 15.0D);  
 }  
   
 public ScrollBar getScrollBar() {  
 return xscrollBar;  
 }  
 public void setValue(int value) {  
 xscrollBar.setValue((double)value);  
 }  
   
}

**Пресеты из Ensemble.jar:**

****

****