Sketcher Program
A way to draw a floor plan

Omar Alfakir

1.Introduction:

1.1 Requirements:

In this report we are going to explain our project of the Sketcher program that is a part of *Advanced Object-Oriented Programming* course. To begin with, there are some criterias and requirements that need to be achieved in order to implement a proper and usable program as described below:

1. In which way the components will be drawn and added to the program.
2. Saving and loading features should be provided to the program.
3. The shopping list and which tools should be used in our code in order to provide a proper shopping list for the consumers.
4. The component should be provided by a library of components in order to be easier to use.
5. The layout of the components in consideration of other components.
6. The proper use of design patterns in our program.

1.2 Design:

In terms of designing our program should be provided with an ability to draw a wall, window and door (construction tools) afterwards the user can be able to add furniture to those drawn constructors to represent a Room/Apartment floor plan. Beside that the user of the program should be able to do some modifications on the component for example rotate, remove and move. **1.3 Implementation:**

In our program we have achieved these mentioned criterias by implementing and building our program from the scratch. Furthermore, we have relied on our own construction and ideas to fix a problem that we have faced during the writing of this program. For instance, we started with a simple frame and we added some modified shapes to it so step by step a library that contains a huge amount of components has emerged.

1.4 Testing

In our program the testing by junit was difficult to achieve, so that is why we relied on testing our program by debugging and using the frame to see how our implementation is working and functioning without any issues.

1.5 inspiration

We have been inspired to draw the component in our program from an application on google play (Madrees). This app has inspired us on how to draw furniture shapes. The main difference in functionality between our program and this app is that this app is available on android phones and has a pro version that costs money to have, while our program is working on PC and beside that is still free to use.

2. Designing and Organization of the program

2.1. The first thing that the users are going to get once they run the program is the welcoming frame(figure 2.1) which contains two options: create a new project or open an existing project (create new or open), in addition there is an image of an example that illustrates how the result would be by using this program. We have provided this initial frame to give the user an ability and flexibility to choose between new projects or open an already existing one.

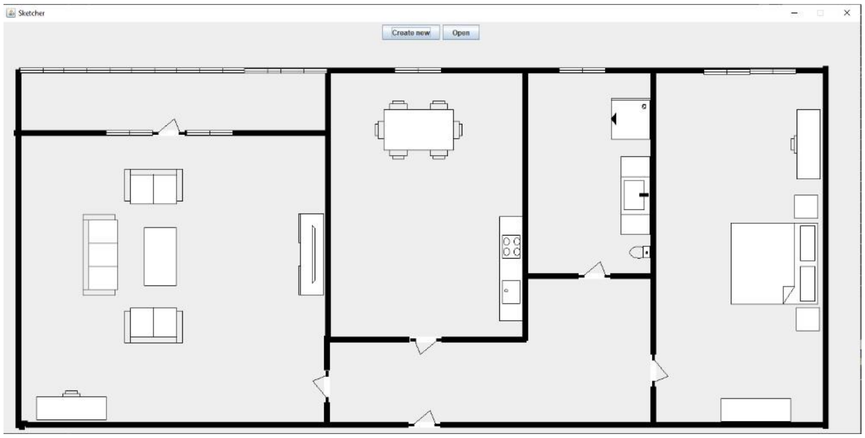


Figure 2.1

2.2. Once the user chooses one of the two choices from the initial window, the main frame will pop up with a constructor panel to start drawing the construction of the floor plan. In order to provide furniture to the construction shape that the user has been drawn in, the user should switch to the “Add furniture menu” under tools on the menuBar(figure 2.2). The reason behind why we have chosen to create two separate panels is because we wanted to separate the modifications features that would occur on the components according to which panel’s components will be modified. For example, if the user has construction of a room and furniture and needs to move the bed in the room then the construction component will not be affected by move feature.

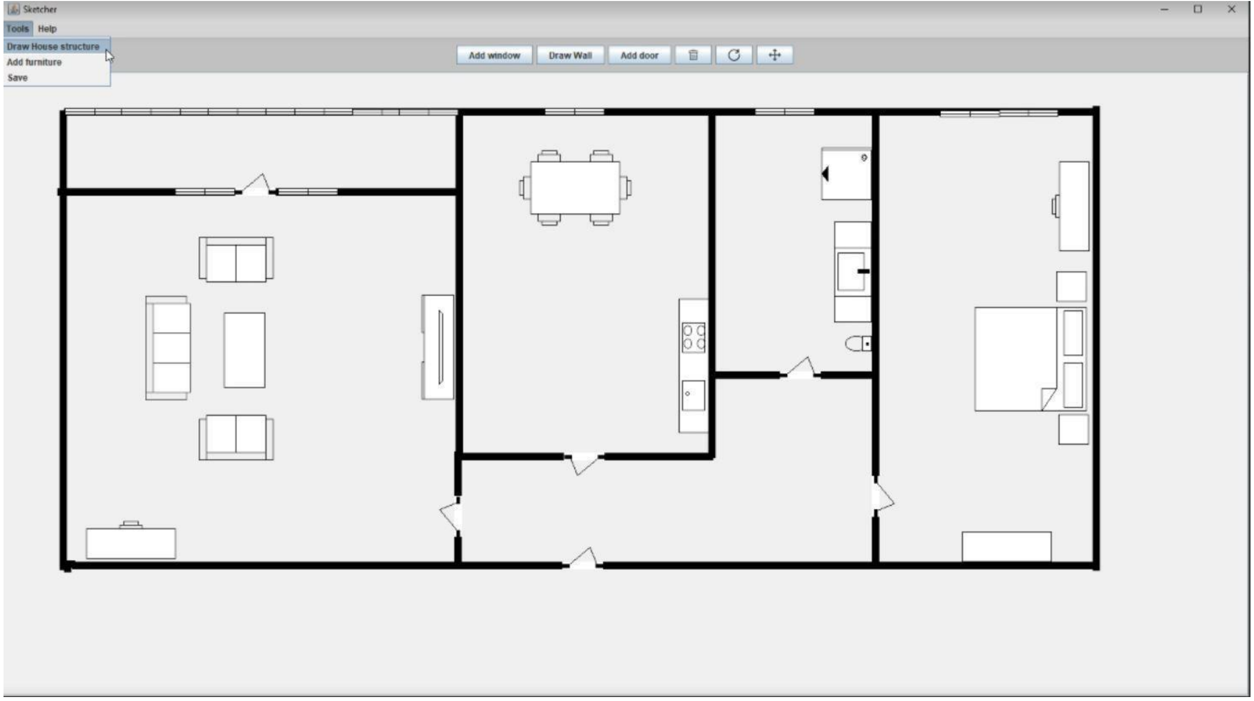
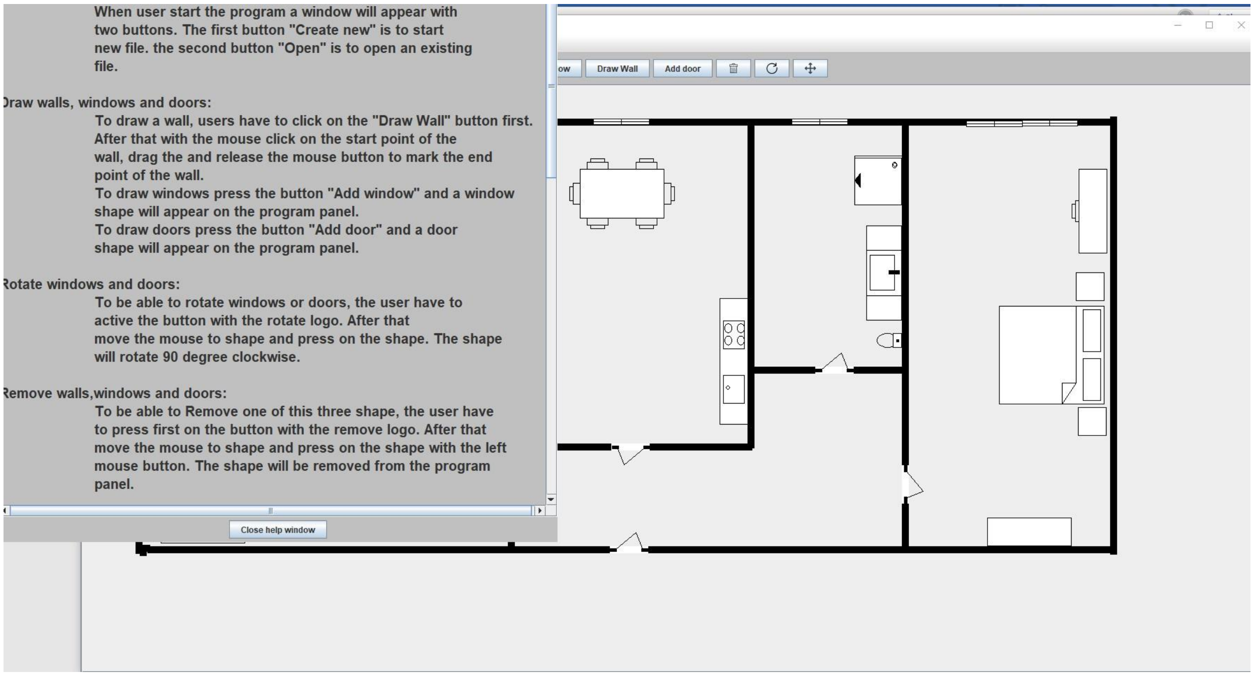


Figure 2.2

2.3 MenuBar manages to switch between the two panels by choosing one of the two choices “Draw House structure” and “Add furniture” and it also provides a save feature under the “Tools->Save” option. Additionally, there are some hints under the “Help->Open help screen” that provide a guide information on how to use the program(figure2.3).



When user start the program a window will appear with two buttons. The first button "Create new" is to start new file. the second button "Open" is to open an existing file.

Draw walls, windows and doors:

To draw a wall, users have to click on the "Draw Wall" button first. After that with the mouse click on the start point of the wall, drag the and release the mouse button to mark the end point of the wall.

To draw windows press the button "Add window" and a window shape will appear on the program panel.

To draw doors press the button "Add door" and a door shape will appear on the program panel.

Rotate windows and doors:

To be able to rotate windows or doors, the user have to active the button with the rotate logo. After that move the mouse to shape and press on the shape. The shape will rotate 90 degree clockwise.

Remove walls,windows and doors:

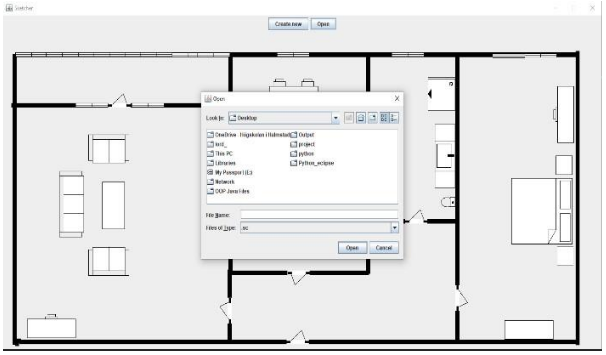
To be able to Remove one of this three shape, the user have to press first on the button with the remove logo. After that move the mouse to shape and press on the shape with the left mouse button. The shape will be removed from the program panel.

Figure 2.3

2.4 Save and Load

Load features can be accessed on the welcome frame under “open” which gives an ability to the user to load a saved project(figure2.4.1).

Save feature can be accessed on the runtime under “Tools->Save” (Figure2.4.2) or when the user wants to exit the main frame, thereby the user will not risk losing any data by accidentally exiting the main frame.



Look In: Desktop

File Name:

Files of Type: .sc

Figure 2.4.1

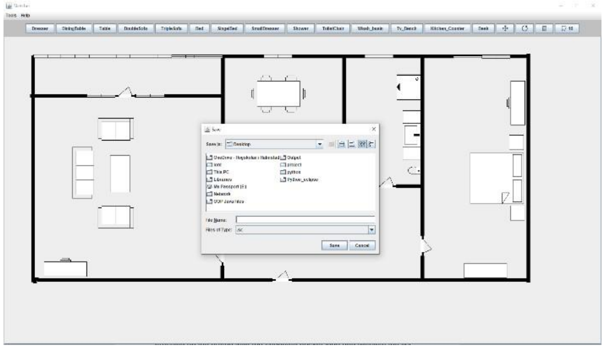
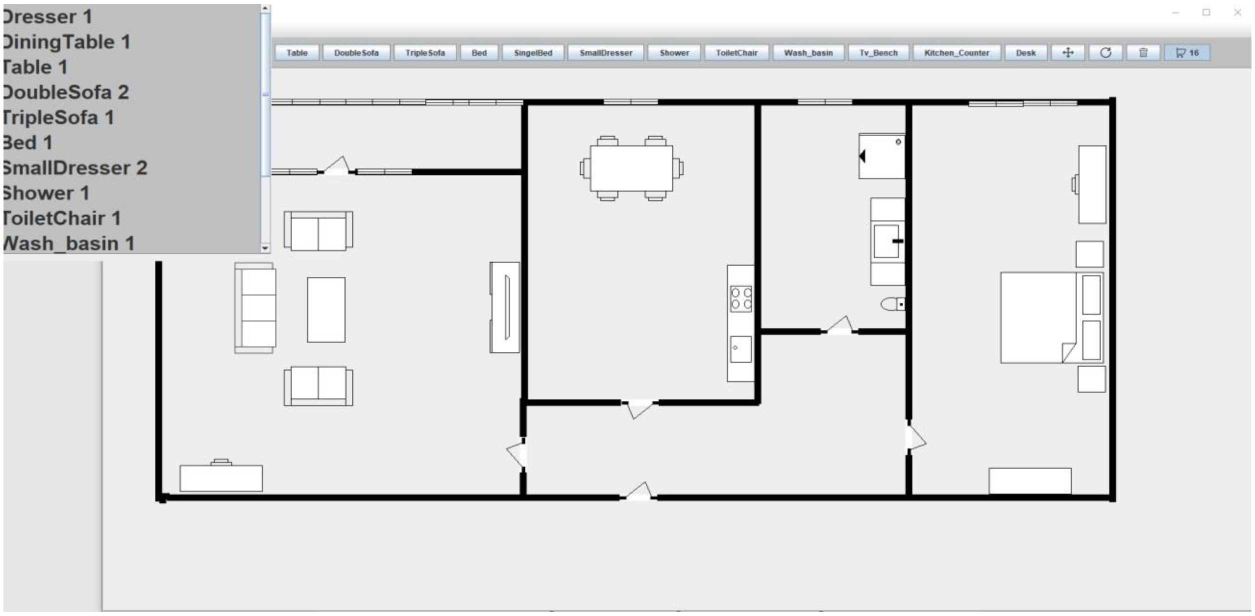


Figure 2.4.2

2.5 Shopping List

Shopping List features always get updated every time the user adds new furniture. This feature can be accessed any time at the running time by pressing on the button with the shopping basket logo (figure 2.5) and divisible the list by pressing again on the same button. We have also provided the button beside the logo with the quantity of the furniture. The reason behind designing the shopping list in this way is to provide an external frame with the information instead of printing the information on the main frame.

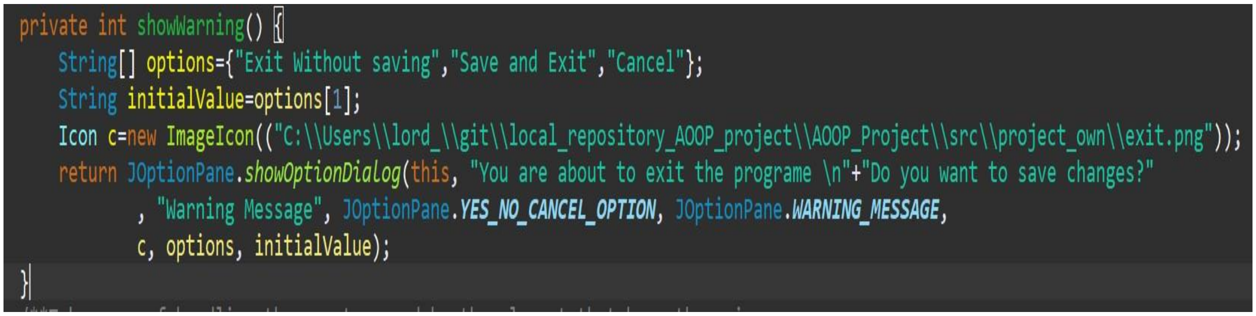


```
Dresser 1
DiningTable 1
Table 1
DoubleSofa 2
TripleSofa 1
Bed 1
SmallDresser 2
Shower 1
ToiletChair 1
Wash_basin 1
```

Figure 2.5

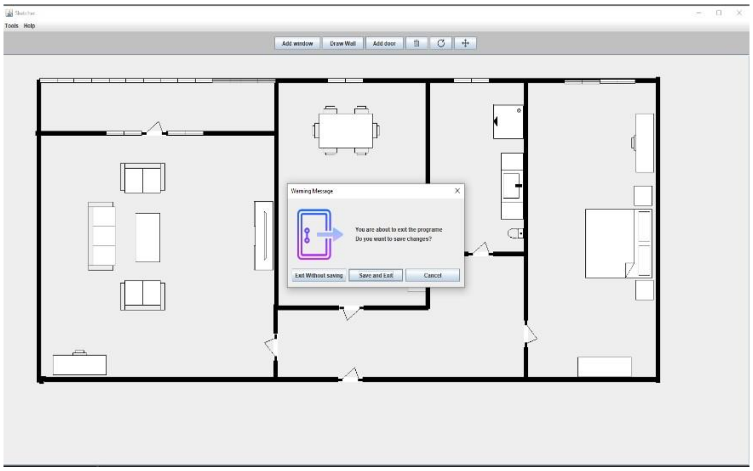
3.Interesting part Exit and Save

One of the interesting things that we wanted our program to have was a way to notify or warn the users that they are about to exit the program and there is some data that needs to be saved (figure 3.2). Therefore, we have provided our main program “Sketcher.java” with a function called “showWarning()” (figure3.1) which shows option dialog and we have added a windows listener to the main frame. So, each time the user is about to exit the program this window will appear with three choices “Exit Without saving”, “Save and Exit” and “Cancel”. The returning of option dialog is defining whether the program will be exited without saving, invoke the saving feature or this dialog will be exited.



```java
private int showWarning() {
    String[] options={"Exit Without saving","Save and Exit","Cancel"};
    String initialValue=options[1];
    Icon c=new ImageIcon(("C:\\Users\\lord_\\git\\local_repository_AOOP_project\\AOOP_Project\\src\\project_own\\exit.png"));
    return JOptionPane.showOptionDialog(this, "You are about to exit the programe \n"+"Do you want to save changes?"
            , "Warning Message", JOptionPane.YES_NO_CANCEL_OPTION, JOptionPane.WARNING_MESSAGE,
            c, options, initialValue);
}
```

Figure 3.1



Warning Message

You are about to exit the programe
Do you want to save changes?

Figure 3.2

4.Result:

In our project we satisfy all the requirements for the project that we have been asked for. In addition, we have added some other features because we felt that they would improve the project to function in a more efficient way. Some of these features are the welcome frame and option dialog. By the implementation of the last-mentioned features and the other features that we have asked for, we achieved program work as intended.

Our program can be modified to get real time data of furniture components like prices from marketing websites and also our shopping list can link those furniture to real pricing data. That can be done by providing the shopping with a list of URL links that can be linked to the furniture list by the name. Another example that can extend our program is by easily adding a new shape of furniture by implementing the interface (Fshape).

5.How classes are related

5.1 Sketcher.java

Sketcher class contains four variables from Drawing.java, MenuBar.java, OutputWriter.java and ReadFromFile.java. So, the main objective of Sketcher class is to manage the connection between those four classes as explained below:

Drawing.java this class will get updated when the user chooses to open a saving file. And share panels with the MenuBar class.

MenuBar.java will be initialized and will be provided with two panels by this class (Sketcher).

OutputWriter.java will be invoked in case the main frame is closed to save data on an external file.

ReadFromFile.java will be invoked in case the user has chosen to open an existing project.

5.2 Drawing.java

This class will handle progress of adding and notify the observers with updates. The main object of this class is to draw the component provided by the “component library”. And this library contains all the shapes needed for this program.

5.3 OutputWriter.java

This class has the responsibility of writing out data to files when its method "writeToFolder" is invoked by other classes. The data that exists in this class is provided by UpdateObserver.

5.4 MenuBar.java

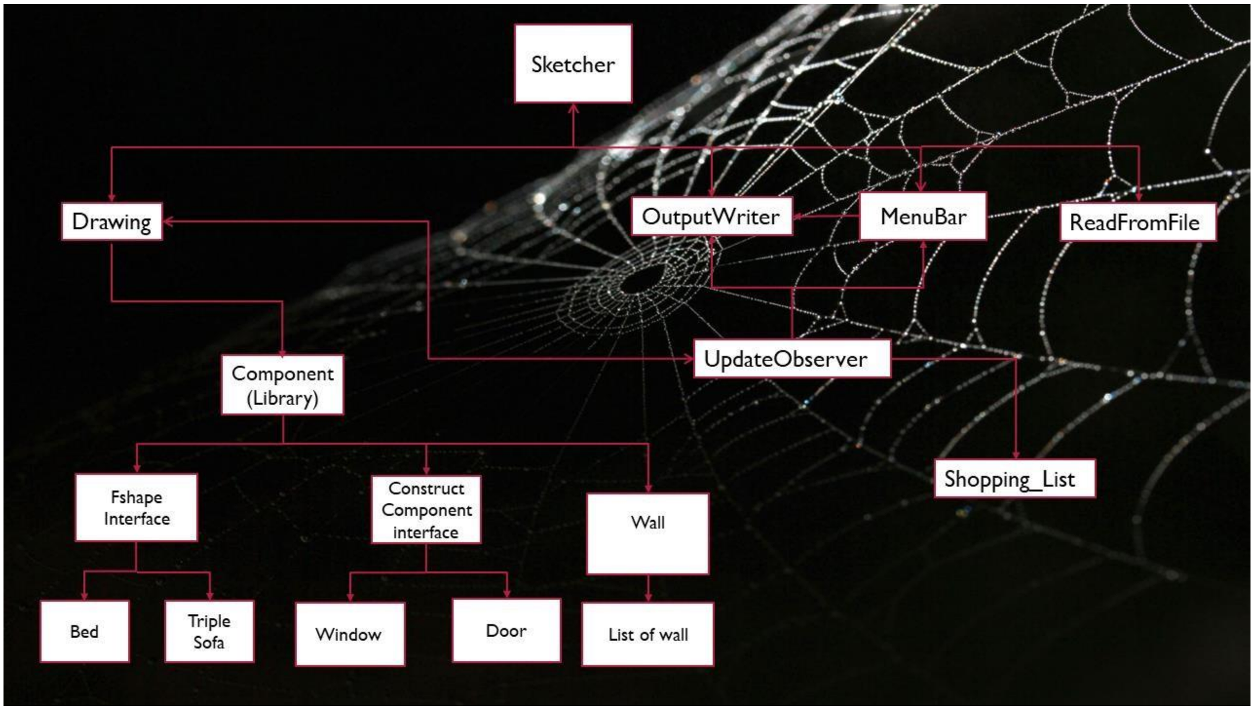
This class will control the swapping between panels as well as invoking the "writeToFolder" method that exists in "Outputwriter" in case the user chooses to save the project.

5.5 ReadFromFile.java.

Take care of extracting data from the saved file and provide Drawing class with the data in order to be drawn.

5.6 UpdateObserver interface

Handle the progress of notifying any class that implements this interface with the updated data.



6.Our experience with Version control:

Initially it was very difficult to start using version control because it was our first time using any of git-platform. However, it became easier after we watched videos on YouTube and surfed some websites, then we chose to use GitHub to upload our repository and gitkraken to stay connected with the update that any of us commit to the repository.

7.References:

1.google play app,
Madrees,*https://play.google.com/store/apps/details?id=com.anadreline.android.madrees*

2.Protottype pattern,

https://refactoring.guru/design-patterns/prototype/java/example

3.Preventing GFrame exit,

https://alvinalexander.com/blog/post/jfc-swing/closing-your-java-swingapplication-when-user-presses-close-but/?page=2

https://stackoverflow.com/questions/23092794/how-to-stop-jfilechooser-frombeing-closed-when-approve-or-cancel-button-is-pres/23093436
https://www.codejava.net/java-se/swing/preventing-jframe-window-fromclosing

4.Shape Rotation
https://stackoverflow.com/questions/7517688/rotate-a-java-graphics2drectangle

5. How to check if two given line segments intersect

https://www.geeksforgeeks.org/check-if-two-given-line-segments-intersect/