2D Room Creation

Project, Advanced Object Oriented Programming

Authors

Abboud Afram

Danial Sarab Fard Sabet

**Table of content**

**1.1 Introduction**

1.2 Background

1.3 Prediction

1.4 Solutions

1.5 Current room sketchers

1.6 Requirements

**2.1 Design**

2.2 Program Design

2.3 Code Design

**3.1 Code Methods**

 3.2 Testing

3.3 Interesting parts

**4.1 Project Result**

4.2 Result

4.3 Discussion

**5.1 References**

**1.1 Introduction**

**1.2 Background**

The project was the last moment in the course Advanced Object Oriented Programming. The goal was to create an electronic circuit or a room sketcher which included 7.5 HP, the course consisted of lectures, three assignments and a project. In order to meet the requirements of the project, members had to use the various techniques taught during the course, for example implementing Jframe into a class. This included lectures, assignments and previous courses taught in the university.

In this case the room sketcher project was chosen to be created.

**1.3 Prediction**

The project was based on creating a room with the vision of furniture objects could be applied and placed in the room with constraints, for example a chair couldn't be in a specific shape or the wardrobe couldn’t be placed in front of a window. Different furniture could be placed in the room with different colours by choice. By pressing the furniture button left side of the frame a new frame would pop up with a label and buttons asking what shape the furniture should be, when the specific shape is chosen another frame would pop up where a specific colour could be chosen. After the chosen commands the furniture would appear on the mainPanel with specific coordinates and moved around as wished. When the furniture was added to the mainPanel, the furniture would be added to the shopping list section representing how many and which kind of furniture was added to the room. After the room is created it could be saved as a file and opened when necessary. By creating border around the mainPanel and constraints on the door and windows, the furniture object couldn’t be moved outside the mainPanel or placed in front of the door or windows. This concept was sketched with pen and paper to get the users perspective of the program. Everything was written and sketched down, where the buttons, tool kit (Save and open file), shopping list and the room panel should be placed in the program so that the user easily could navigate and understand how the room sketcher looks like and works.

**1.4 Solutions**

Difficulties occurred in the process of creating the program. The project consisted of delimitations that had to be interpreted correctly. The main concept of the program was to make it as similar as drawing a room containing different furniture with pen and paper. Being able to create such a program in java language required many steps.

* Creating a JFrame with components.
* Creating the furniture objects.
* Moving the objects inside the JPanel.
* Adding objects to shopping list.
* Opening and saving finishing room as a file.

Creating an abstract class which extends JFrame represents the frame of our program. In the abstract class different abstract methods are declared for the creation of different components for example, one abstract method represents the mainPanel which the furniture objects are placed in. With the help of Borderlayout() the layout of the frame could be decided and designed.

Creating the furniture objects consisted of 2 options, either using Graphics or implementing Icon interface. Comparing the two, Icon was a better choice to build the furniture objects. By implementing Icon, two classes were created that represented the shapes of the furniture objects and then easily be added to JLabels and JPanel.

To be able to move the furniture objects Mouselistener and Mousemotionlistener was implemented in a new class. By implementing the mousedragged method, the furniture objects coordinates could be identified in the JPanel and set in the new location the object was dragged to. The location was limited so that the mousedragged method couldn’t be operable outside the coordinates of the mainPanel.

When a new furniture object is added to the mainPanel with a specific colour the name of the furniture with the colour would be added to the shopping list with a specific number. This way it is easier to see which furniture are added to the panel and the amount of the furniture kind. By creating a linkedhashmap and adding every furniture to the list when the furniture object is created, the program would with the help of a counter add the type of furniture and the specific colour.

To open and save the file a new class was created with two different methods. The writeToFile() method which transfers all the component from the mainpanel and the shopping list to two created component arrays, one array saves the furniture objects and the other array saves the shopping list. The redFromFile() method consists of choosing the file and transferring all data from the chosen file to two new components arrays and adding the objects to the main panel and the shopping list.

**1.5 Current room sketchers**

There are already many room sketchers programs with different softwares, in this project a lot of inspiration came from an specific websites [1] , how to think, create and use. The main difference between the website and the project is that the user can pick between 2D and 3D image shapes and instead of decorating a room the user can choose between a house or an apartment plan to decorate. In the project coordinates was used to apply the furniture with definite width and height to the panel unlike websites that use the metric system. The programs in general serve the same purpose but uses different graphics and tools. The project uses java language and most of the room sketcher programs use adapted software, for example CAD (Computer-Aided Design) software. CAD is used for designers, drafters, architects or engineers and is technology for design and technical documentation, which replaces manual drafting with an automated process [2]. While java is a programming language that is class based, object oriented and mainly used by programmers and engineers. By comparing the two software's one can clearly understand that most room sketchers are used by automated software while in java language the programmer writes code for every detail and idea. In conclusion, using java language to create a room sketcher is much more advanced than using an automated software that is designed to create rooms.

**1.6 Requirements**

The created program should be easy to operate and understand, just by running the program the user will directly see how easily operable it is with the flawless design. By clicking on the type of furniture to be created and by following the simple instructions that appears on the screen, the user will be able to create different shapes of furniture with the specific chosen colours. The furniture object can be moved around freely regardless the amount of furniture in the room and everything created will be added to the shopping list. With this straightforward functionality the program can be saved and opened as a file. Implementation of the code are written in java language and the program consists of different classes which are connected to each other. Some classes were created to design (the view) the program and some classes were created to give the functionality (the controller) of the program.

**2.1 Design**

**2.2 Program Design**

Coming up with how the room sketcher should be designed was decided with pen and paper before the coding started, this made the organizing part simple. Creating different classes with different tasks made the program more efficient to understand and run.

A picture containing screenshot

Description automatically generatedBy applying buttons, toolbar, shopping list and a main panel to the frame, the program could be designed and organized as sketched with pen and paper. The design works like an instruction to make it easier to navigate and use, as shown in figure 1.

Figure 1: Design of the finishing program, the design works like an instruction, navigate through the program is simple and without problems.

**2.3 Code Design**

To simplify the program, the code has been divided into one abstract class, one concrete class and multiple classes (Shown in figure 2).

A picture containing screenshot

Description automatically generated

Figure 2: A diagram representing the design pattern used in the program

**Frame**

The created abstract class is the frame of the program which contains the different abstract methods that represents different tasks in the program.

**Furniture**

Furniture objects are created in furniture class which contains square class and circle class, both classes implement Icon. A specific square furniture object contains width, height and colour. The square object represents both rectangle and squared furniture depending on width and height. A specific circle furniture object contains radius and colour.

**Handler**

The handler class implements both mouselistener and mousemotionlistener, these interfaces contains different methods. These methods are used to be able to perform different actions with the furniture objects, for example moving the furniture object to specific locations or removing a specific furniture object from the panel and the shopping list.

**Room sketcher**

The room sketcher class implements all the abstract methods from AbstractFurniture class. All the different buttons are declared and created in room sketcher, every significant button has a specific task. These tasks are decided by different methods, for example the furniture buttons are used to create different shapes of furniture objects with different size and colour.

**3.1 Code Methods**

**3.2 Testing**

With the limited time the group members didn’t have time to test the RoomSketcher program with any test tools. If there was time over, the group members would decide to test the program with Junit where different pre- /postconditions for every written method  would be tested with the help of assertions.

With the time that was provided the group members decided to write a pre- / postcondition for almost all the method. With the help of conditions, the methods requirements could be understood and how they were supposed to work, this will be shown in the code. Pre-conditions are the things that must be true before a method is called. The method tells clients “ this is what I expect from you”. Post- conditions are the thing that must be true after the method is complete. The method tells clients “this is what I promise to do for you”. Invariants are the things that are always true and won't change. The method tells clients “if this was true before you called me, I promise it’ll still be true when I’m done”.

In conclusion the code has been documented and pre- / postcondition are used.

**3.3 Interesting parts**

When writing the code in java language, the group members found ways to tweak the code and make it more efficient. When coding at first, Graphics objects were used to create furniture objects. This was successful, but the furniture objects couldn’t be seen or added to the mainPanel. The solution to the problem was to implement Icon instead of using Graphics. Where a new class could be created with subclasses as square class and circle class that implements icon interface adding the furniture objects to different JLabels and then adding it to mainPanel. This was the correct action to create the furniture objects and adding them.

To be able to move the furniture objects in the mainPanel, a handler class was created that implements both MouseMotionListener and MouseListener. The problem was that the furniture objects could be moved outside the mainPanel, because the different furniture objects had different sizes and shapes. To prevent the furniture objects from moving outside the mainPanel the method mousedragged() was used. This way the furniture objects became limited from moving outside the mainPanel.

**4.1 Project Result**

**4.2 Result**

The result of the project is presented below.

A picture containing screenshot

Description automatically generated

Shopping list Panel

mainPanel

Figure 3, The finished project, with the furniture options buttons to the left with the shopping list containing the added furniture that is displayed on the big panel in the middle. With a toolbar panel north west on the frame with the save and open option.

A screenshot of a cell phone

Description automatically generated

Figure 4, When one of the furniture object buttons is pressed this option will appear where the shape can be chosen.

A screenshot of a cell phone

Description automatically generated

Figure 5, when shape is chosen the colour option will appear where the colour can be chosen. After this option the furniture will be added to the pane and the shopping listl.

A screenshot of a cell phone

Description automatically generated

Figure 6, If a furniture object is placed in front of the door or an wardrobe is placed in front of a window this dialog will appear on the screen and the furniture object will be re-placed in the middle of the panel.

As explained and shown, the design of the project is already explanatory. The buttons to the left are where the furniture objects is created and the options shown in figure 4 and 5 will appear. When the furniture object is created it will be placed in the middle of the main panel, the furniture object is free to be moved and placed in the panel except in the constraint areas. If the furniture object is placed in a constraint area, the dialog in figure 6 will appear. The furniture is also able to be stacked on top of each other, because if many furniture objects are added to the room the users hands are not tied when moving the furniture objects around.

When the furniture is created, the name of the furniture will be added to the shopping list with the quantity of the furniture kind as shown in figure 3, this is created so that it is easy to keep track of the number of the same furniture object added. There is also an option to remove furniture objects both from the big panel and from the shopping list by right clicking.

When the room is finished and decorated, the room can be saved as a file by clicking on the save button on the toolbar and opened if changes are necessary.

**4.3 Discussion**

The project design and the functionality is as the group members intend to create, the sectioning of the panels on the placement of different buttons is planned as a sketch before starting with the project. In our opinion all the requirements were met with the program and the project. The only main thing that separates our project with most of the room sketchers is that there are no constraints between the furniture objects, this functionality was intended. We thought that the user should be free with the placement and the adding of the multiple furniture objects in the program. The advantage with our program compared with many other is that the user can add multiple furniture to the panel and the program will not complain. Another advantage is that our program is easy to understand and handle, every button and error have simple instructions the user can understand and follow.

Our program don’t have the functionality to decide the size of the room nor if it is an apartment or a house. If this function should be implemented most of the code would still work, but a part of the code  would have to been rewritten and re- implemented to work with the new functionalities.

Although, the current program can still implement more furniture objects ( for example carpet, fridge, etc) and more various colours by adding already written code with some small changes. New buttons can be easily added and various design. Instead of shapes we can implement BufferedImage and replace the shapes with images.

In conclusion we created the intended program as planned and met the requirements for the project. The program have space for improvement, but with high expectation and optimistic time we couldn't fully improve our program. Something we as group members did well is to use GUI and template pattern that was taught in this course and assignment.

**5.1 References**

[1] Inspiration taking from:<https://www.roomsketcher.se/>

[2] <https://www.autodesk.com/solutions/cad-software>