**UC1 Edit Booth Floor Plan**

1. **Produce a description of the business process for the use case.**

**Creating a Booth Floor Plan:**

A designer begins by gathering the necessary tools: a sheet of graph paper (or plain paper), a ruler, a pencil, and an eraser. The designer first establishes the scale for the floor plan, such as 1 inch representing 5 feet, ensuring that the size of the paper can accommodate the entire space.

The designer starts by drawing the outline of the event space on the paper. This could be the dimensions of a room, a hall, or an open area, depending on the event's location. The outline is carefully measured and drawn using the ruler, with the dimensions clearly marked along the edges.

Next, the designer considers the number and sizes of booths needed. In this case, there are three different booth sizes to incorporate. The designer lightly sketches the booths within the outlined space, carefully arranging them to make the most of the available area. Booths of the same size might be grouped together for a more organized appearance, with walkways and access points planned out between them.

Once satisfied with the initial layout, the designer refines the booth placements, considering factors like traffic flow, accessibility, and visibility. Adjustments are made as needed, ensuring that the layout is functional and aesthetically pleasing. The designer might erase and redraw certain areas several times to optimize the arrangement.

After finalizing the booth locations, the designer darkens the lines with a pen or a darker pencil, making the floor plan more readable. Each booth is labeled according to its size and position, perhaps using a system like "S1" for the first small booth, "M1" for the first medium booth, and "L1" for the first large booth. If the designer wishes to highlight different booth sizes visually, they might use colored pencils or markers, assigning different colors to each size category.

In addition to booths, the designer includes other essential elements such as entrances, exits, stages, or seating areas. These are placed strategically within the layout to ensure they complement the booth arrangement without causing congestion.

**Editing the Booth Floor Plan:**

If the need arises to edit the floor plan, the designer begins by reviewing the entire layout. Changes might be required due to vendor requests, unexpected space constraints, or a desire to improve the flow of the event. The designer uses an eraser to remove or adjust the affected sections, being careful not to disrupt the overall balance of the layout.

The designer repositions booths, redraws boundaries, and relabels areas as necessary, continually checking that the modifications enhance the plan’s functionality. Once all changes are made, the designer retraces the lines, ensuring that the final version is clear and precise.

Finally, the designer adds any additional notes or labels, dates the plan, and stores it safely for use during the event setup. This manual process allows the designer to maintain a high level of control over the layout, making it possible to create a customized, well-considered floor plan that meets the event's specific needs.

1. Do domain model brainstorming on the description produced above, classify the brainstorming results into classes, attributes of classes and relationships between classes, and visualize the classification results in a UML class diagram (which must not show any operations).

**Part 1 for Step 2. Brainstorming**

The team members get together and identify & list

(*domain specific, domain specific, domain specific*)  
1. nouns / noun phrases  
2. "X of Y" expressions (e.g., color of car)  
3. transitive verbs  
4. adjectives, enumeration  
5. numeric, quantity  
6. possession expressions (has/have, possess, etc.)  
7. “Consist of / part of" expressions  
8. containment / containing expressions  
9. "X is a Y" expressions (generalization/specialization)  
 \*\* X, Y are nouns/noun phrases

1. **Domain-Specific Nouns / Noun Phrases: (Pratikshya Devkota)**

* Designer
* Floor plan
* Event space
* Booths
* Walkways
* Access points
* Traffic flow
* Entrances
* Exits
* Stages
* Seating areas
* Vendor requests
* Space constraints

1. **Domain-Specific "X of Y" Expressions: (Pratikshya Devkota)**

* Sheet of graph paper of the tools
* Ruler of the tools
* Pencil of the tools
* Eraser of the tools
* Dimensions of the room
* Booths of the event space
* Sizes of the booths
* Colors of the booth size categories
* Scale is an attribute of the floor plan
* Size is an attribute of the paper
* Dimensions are attributes of the room or hall
* Position is an attribute of the booth
* Accessibility is an attribute of the layout
* Ruler is part of the toolset
* Booths are part of the event space
* Walkways are part of the layout
* Entrances and exits are part of the event layout
* Designer is a role in the creation of the floor plan
* Booth is a role in the event space organization
* Traffic flow is a role in the layout optimization

1. **Domain-Specific Transitive verbs: (Steven Fitzgerald)**

* establishes (scale)
* drawing (outline)
* measured (outline)
* sketches (booths)
* arranging (booths)
* refines (placements)
* erase (areas)
* redraw (areas)
* darkens (lines)
* labeled (booths)
* repositions (booths)
* retraces (lines)
* marked (dimensions)

1. **Adjectives, Enumeration (Sanjana Ravi Prakash)**

* Initial layout, Final Layout
* carefully measured --- isMeasured() or Measure(values)
* functional and aesthetically pleasing.
* Booth Sizes - Small, Medium, Large --- Enumeration
* placed strategically
* Congestion

1. **Numeric, Quantity (Taksh Patel)**

* 1 inch representing 5 feet - scale
* Three different booth sizes - boothSize
* "S1" for the first small booth
* "M1" for the first medium booth
* "L1" for the first large booth

1. **Possession expressions (has/have, possess, etc.)**

* **"designer’s tools"** – Refers to the tools that belong to or are used by the designer.
* **"event's location"** – Refers to the location associated with the event.
* **"booth’s size"** – Implies the size that belongs to a specific booth.
* **"plan’s functionality"** – Refers to the functionality of the plan.
* **"event's specific needs"** – Refers to the specific needs of the event.

1. **“Consist of / part of" expressions (Jesse Chumo)**

* Floor plan consists of booths, walkways, and access points
* Event space consists of entrances, exits, stages, and seating areas
* Booth is part of the floor plan
* Walkway is part of the floor plan
* Stage is part of the event space
* Seating area is part of the event space

1. **Containment / containing expressions**

* "Floor plan consists of booths, walkways, and access points."
* "Event space consists of entrances, exits, stages, and seating areas."
* "Booth is part of the floor plan."
* "Walkway is part of the floor plan."
* "Stage is part of the event space."
* "Seating area is part of the event space."

1. **"X is a Y" expressions (generalization/specialization)**

* "Booth Sizes - Small, Medium, Large"
* "Plan is a layout"
* "Event space is a type of space"
* "Walkway is a type of access point"

**Part 2 for Step 2. Classification**

1. **Nouns/Noun Phrases**

* Designer - Class
* Floor plan - Class
* Event space - Class
* Booths - Class
* Entrances - Attribute
* Exits - Attribute
* Stages - Attribute
* Seating areas - Attribute
* Space constraints - Attribute

1. **X of Y expressions – X is an attribute of Y, X is part of Y, X is a role in an association**

**X of Y Expressions (Domain-Specific):**

* Dimensions of the room
* Booths of the event space
* Sizes of the booths
* Colors of the booth size categories

### **X is an Attribute of Y (Domain-Specific):**

* Scale is an attribute of the floor plan
* Dimensions are attributes of the room or hall
* Position is an attribute of the booth

### **X is Part of Y (Domain-Specific):**

* Booths are part of the event space
* Walkways are part of the layout

### **X is a Role in an Association (Domain-Specific):**

* Booth is a role in the event space organization
* Traffic flow is a role in the layout optimization

1. **Transitive verbs (Association relationship) - Steven Fitzgerald**

* Drawing
* Marked
* Measured
* Arranging
* Refines
* Repositions
* Redraws
* Relabels
* Erase

1. **Adjectives, Enumeration - attribute value – Sanjana Ravi Prakash**

* Booth sizes – Small, Medium, Large
* isMeasured() or Measure(values)
* Congestion or strategically placed – isCongested()
* Layout – Initial, Final

1. **Numeric, Quantity – attribute value / multiplicity value**

* Size of booth
* 1 inch represents 5 feet - scale

1. **Possession expressions (has/have, possess, etc.) - aggregation, association or attribute**

* Number and sizes of booths - attribute
* Labels for booths according to size and location – attribute
* Designer refining booth placements – Association relationship
* Booths with planned walkways and access points between them – Association relationship

1. **“Consist of / part of" expressions – aggregation relationship - Jesse Chumo**

* "Floor plan consists of booths, walkways, and access points." - Aggregation
* "Event space consists of entrances, exits, stages, and seating areas." - Aggregation

1. **Containment / containing expressions - association or aggregation**

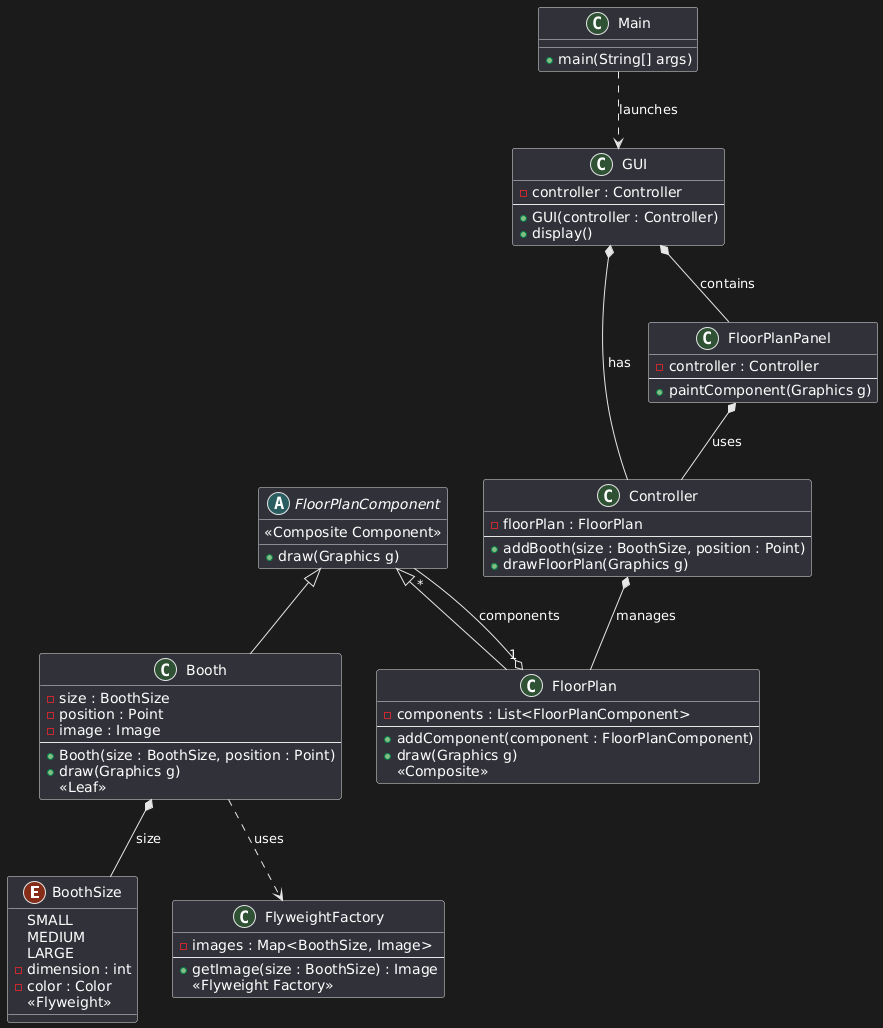
* "Booth is part of the floor plan." - Aggregation
* "Walkway is part of the floor plan." - Aggregation
* "Stage is part of the event space." - Aggregation
* "Seating area is part of the event space." - Aggregation

1. **"X is a Y" expressions (generalization/specialization) – inheritance**

* "Booth Sizes - Small, Medium, Large" - Inheritance
* "Plan is a layout" - Inheritance
* "Event space is a type of space" - Inheritance
* "Walkway is a type of access point" - Inheritance

**Classify everything into classes and attributes**

**Rough UML class diagram**



Step 3. High Level Use Case

TUCBW -> The user clicks ‘Edit Booth Floor Plan’ button

TUCEW -> The user sees a confirmation message the edit has been saved

Step 4. Expanded Use Case (Identify Non-trivial Steps)

|  |  |
| --- | --- |
| Precondition: This use case assumes the user has selected the floor plan they wish to edit | |
| Actor: User | System: Floor Plan Designer |
|  | 1. FPD displays home page |
| 1. TUCBW -> The user clicks ‘Edit Booth Floor Plan’ button | 1. FPD displays the selected floor plan |
| 3. Drag the desired booth size from all the choices to the desired location  on the floor plan and release. | 4. FPD makes a visual copy of the booth at that location on the floor plan |
| 5. The user finalizes edits and clicks ‘Save floorplan’ button | 6. FPD stores the final document and displays confirmation message |
| 7. TUCEW -> The user sees a confirmation message that the edit has been saved |  |

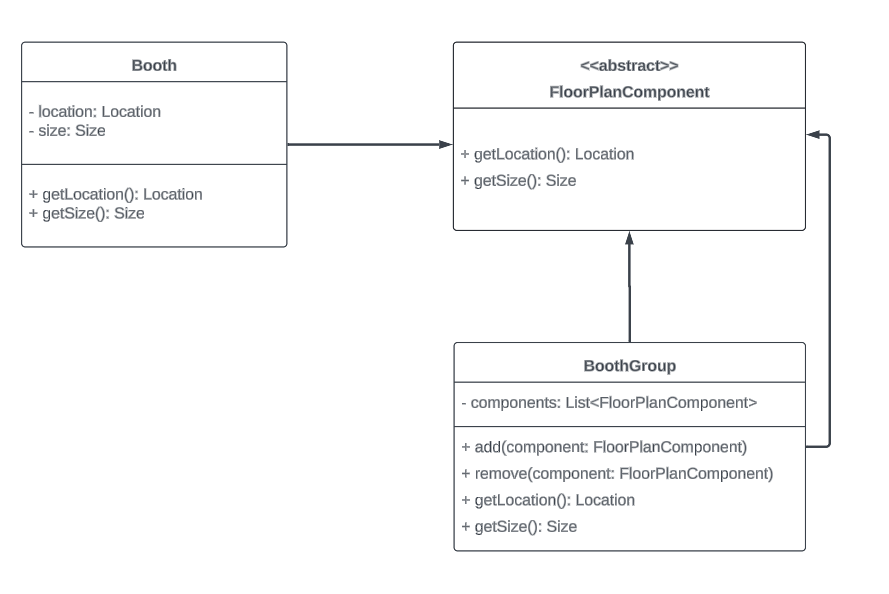
**Non-Trivial Steps:**

4: FPD makes a visual copy of the booth at that location on the floor plan

6: FPD saves the final document and displays confirmation message

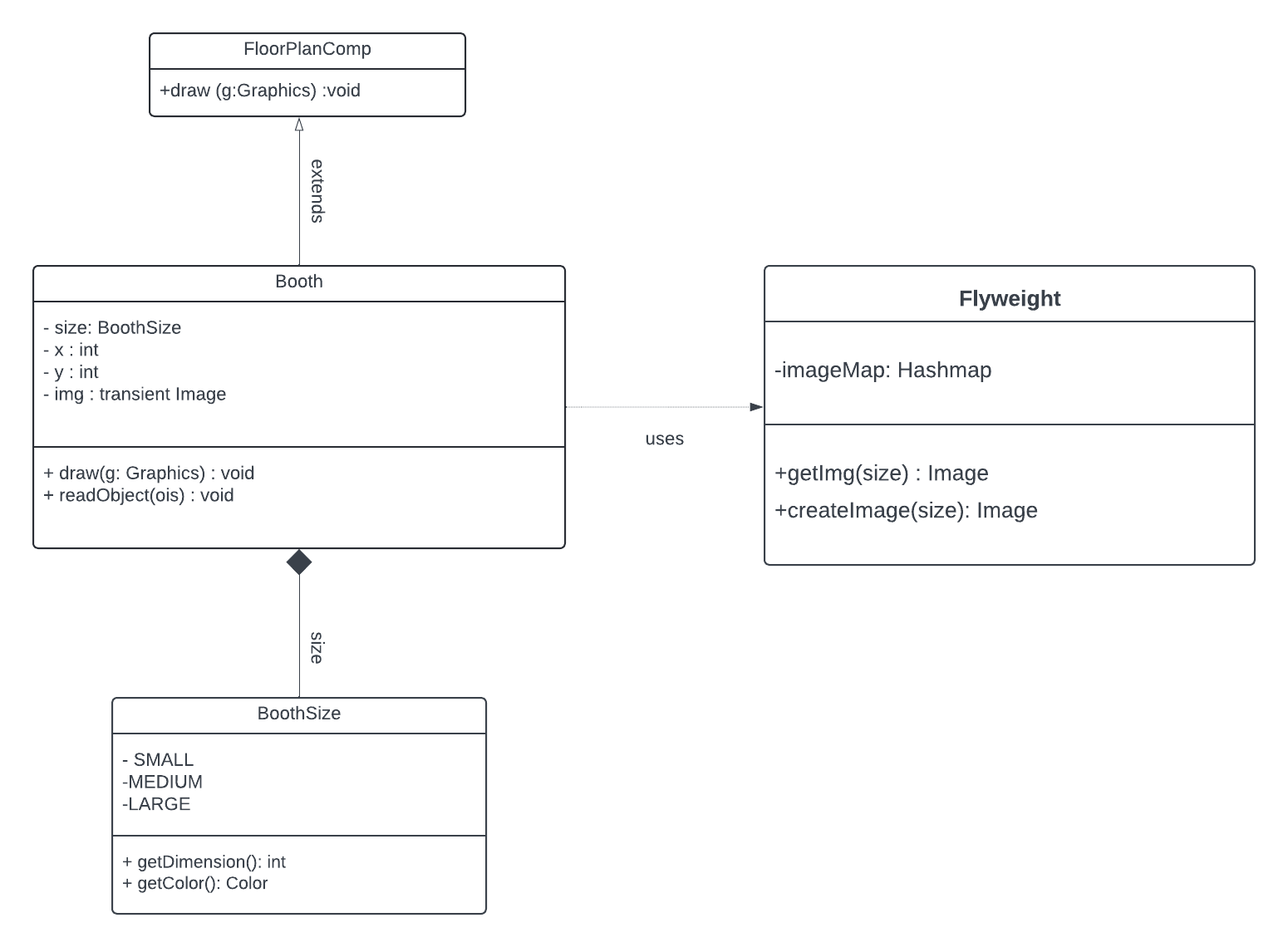
Step 5. Apply each of the required Gang of Four (GoF) patterns by copying the structural design of the pattern, and replacing the generic class names, method names and attribute names with application-specific classes, methods and attributes.

**Composite Pattern**



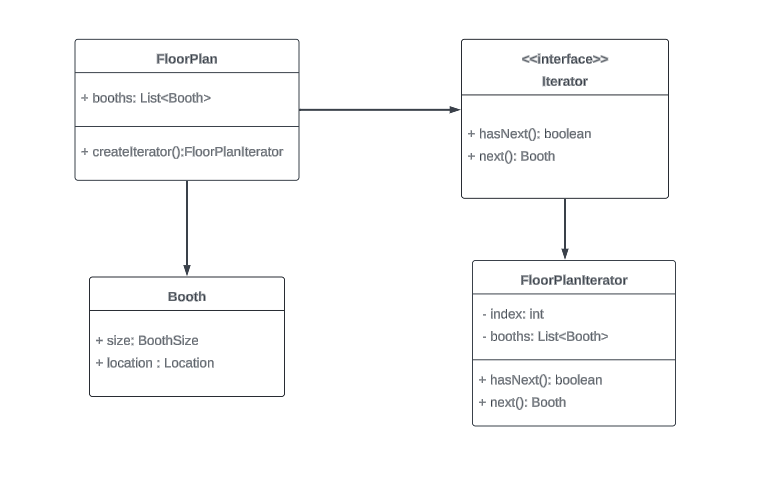
Link available [Composite Pattern](https://lucid.app/lucidchart/8f987704-5540-496c-9ec3-282890749bb4/edit?viewport_loc=-419%2C-260%2C1725%2C1371%2C0_0&invitationId=inv_4cf0df70-c6ab-4f1d-8983-57205c4924b8)

**Flyweight Pattern**



Link: [Flyweight Pattern design diagram](https://lucid.app/lucidchart/f557496c-0da6-4437-a3bf-af40ea4eee04/edit?viewport_loc=-239%2C-159%2C4183%2C2449%2C0_0&invitationId=inv_0489467c-540f-4bfa-b0bb-df1008ff6361)

**Iterator Pattern**

Link: [Iterator Pattern](https://lucid.app/lucidchart/9568751a-ceee-4902-b53a-8d5e8fdf0b2f/edit?view_items=ma3.EOhXMNEx%2C4b3.zj.6uCyM%2CZ72.Bc~mECB3%2C_82.S4XjiFmH%2Cvp3.HJlNoeX4%2CJg3.YmaURr4Z%2CCf3.wBjt0i7s&invitationId=inv_46abd23f-328b-451d-b6f8-b3f87dd46d5e)

Step 6. For each of the nontrivial steps of the expanded use case produce a scenario description to describe how software objects interact with each other to process the actor request to produce the system response (as specified in the nontrivial step). The scenario description must also apply the required design patterns including the controller, expert and creator patterns whenever applicable.

**Non-Trivial Steps:**

4: FPD makes a visual copy of the booth at that location on the floor plan

Make Booth Scenario Description:

3.) Drag the desired booth size from all the choices to the desired location

on the floor plan and release.

4.1) The MakeBooth GUI sends the booth size and location to the MakeBoothController.

4.2) The MakeBoothController calls Booth class to create Booth using the size and location.

4.3) The Booth class passes the argument size to the Flyweight Factory class

4.4) The Flyweight Factory class calls the Hashtable class by providing the size and calls the img function to create the image

4.5) FPD makes a visual copy of the booth at that location on the floor plan

6: FPD saves the final document and displays confirmation message

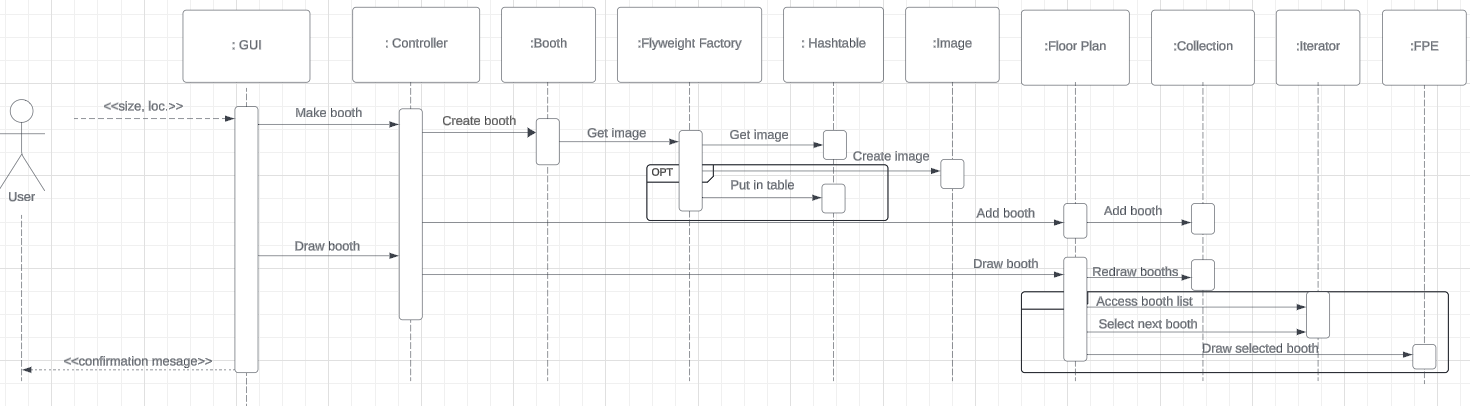
5. The user finalizes the edits and clicks the ‘Save Floorplan’ button.

6.1) Save GUI captures the click of the ‘Save Floorplan’ button and sends a save request to the SaveController

6.2) SaveController sends a confirmation message to the Save GUI indicating the floor plan has been saved successfully.

6.3) Save GUI displays the confirmation message ("Floorplan saved successfully") to the user.

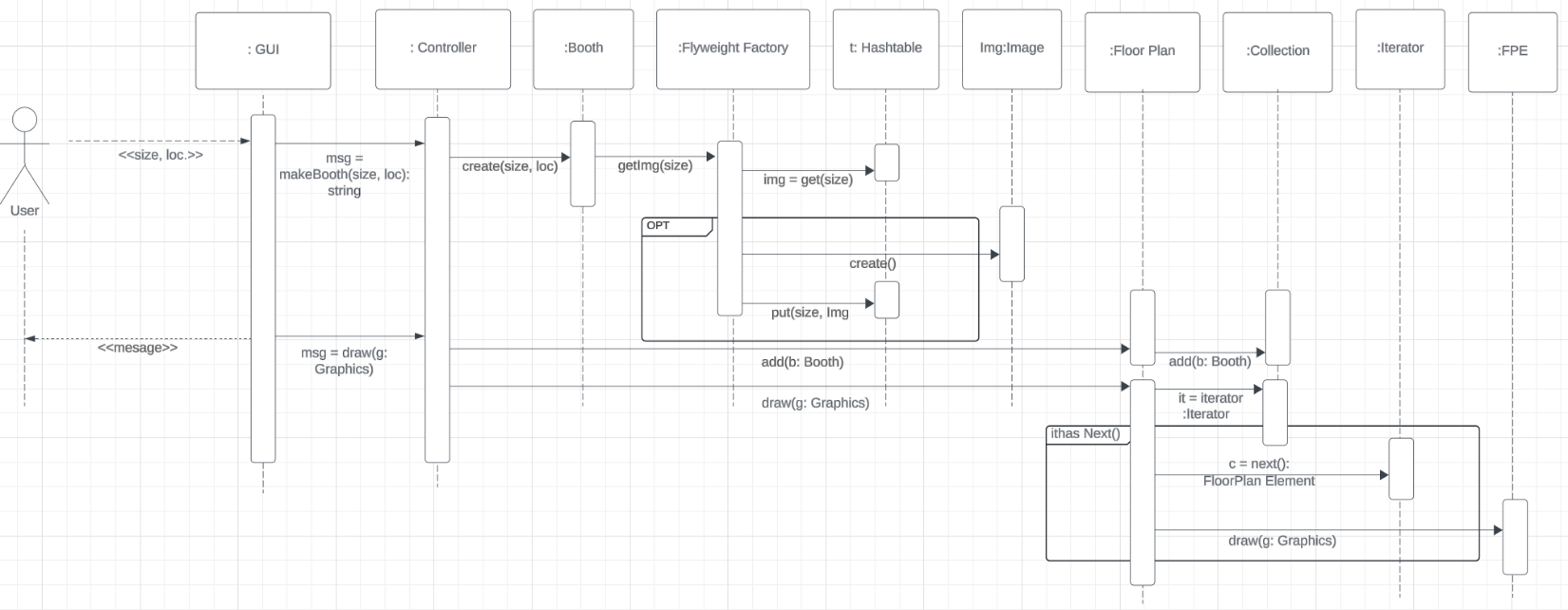
Step 7. Convert the scenario description to an informal sequence diagram in which the messages between the objects are labeled by English texts.



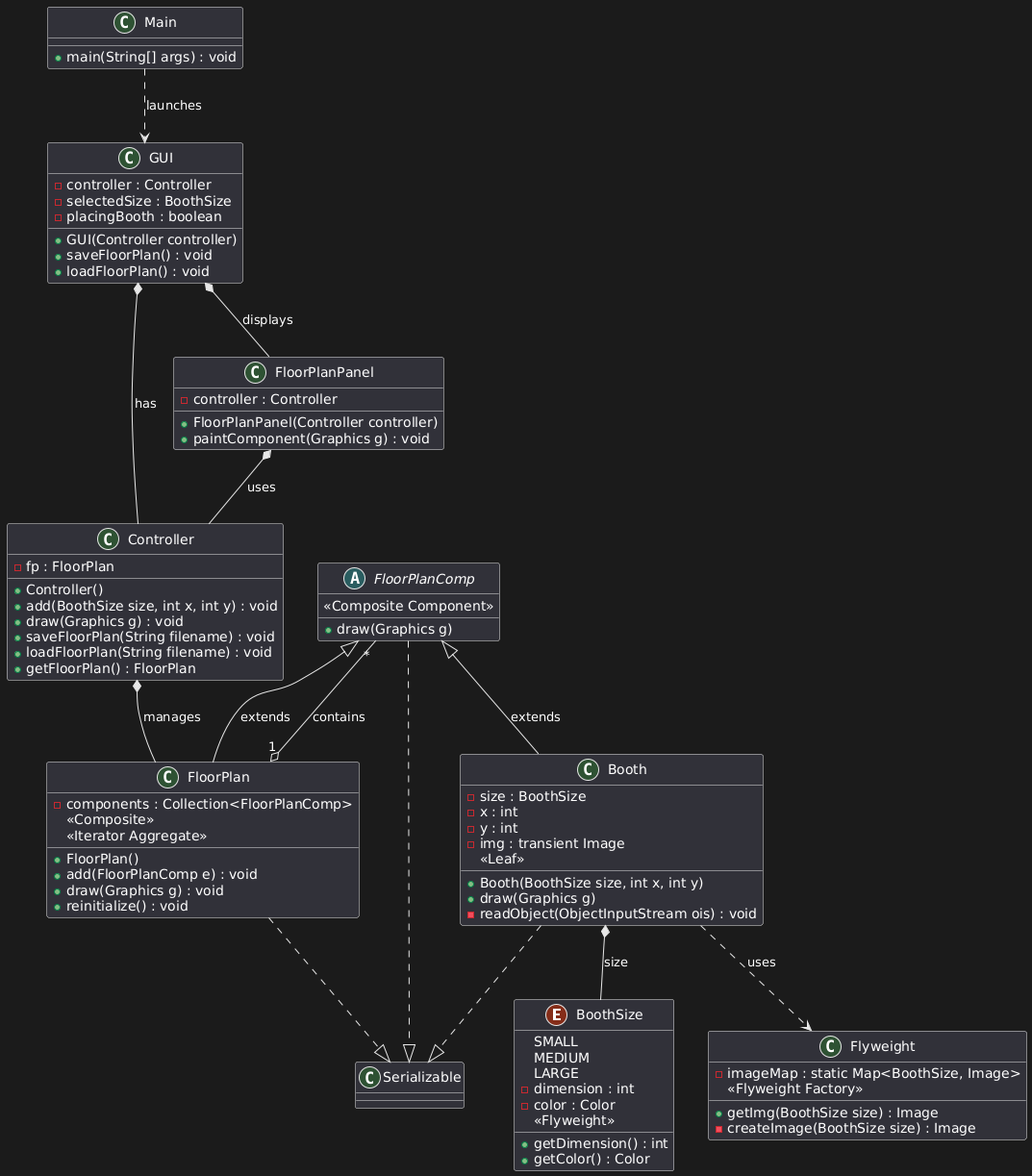
Finished: Link available [here](https://lucid.app/lucidchart/74c22401-a0bd-4ff7-a012-6ea922a1174f/edit?viewport_loc=-912%2C472%2C2491%2C1888%2C0_0&invitationId=inv_3bd0a7eb-39de-4b34-b457-1914fe704817)

Step 8. Convert the informal sequence diagram to a design sequence diagram, which is the same as the informal sequence diagram except that the messages between the objects are converted to function calls with parameters and parameter types, and possibly return values and return types.

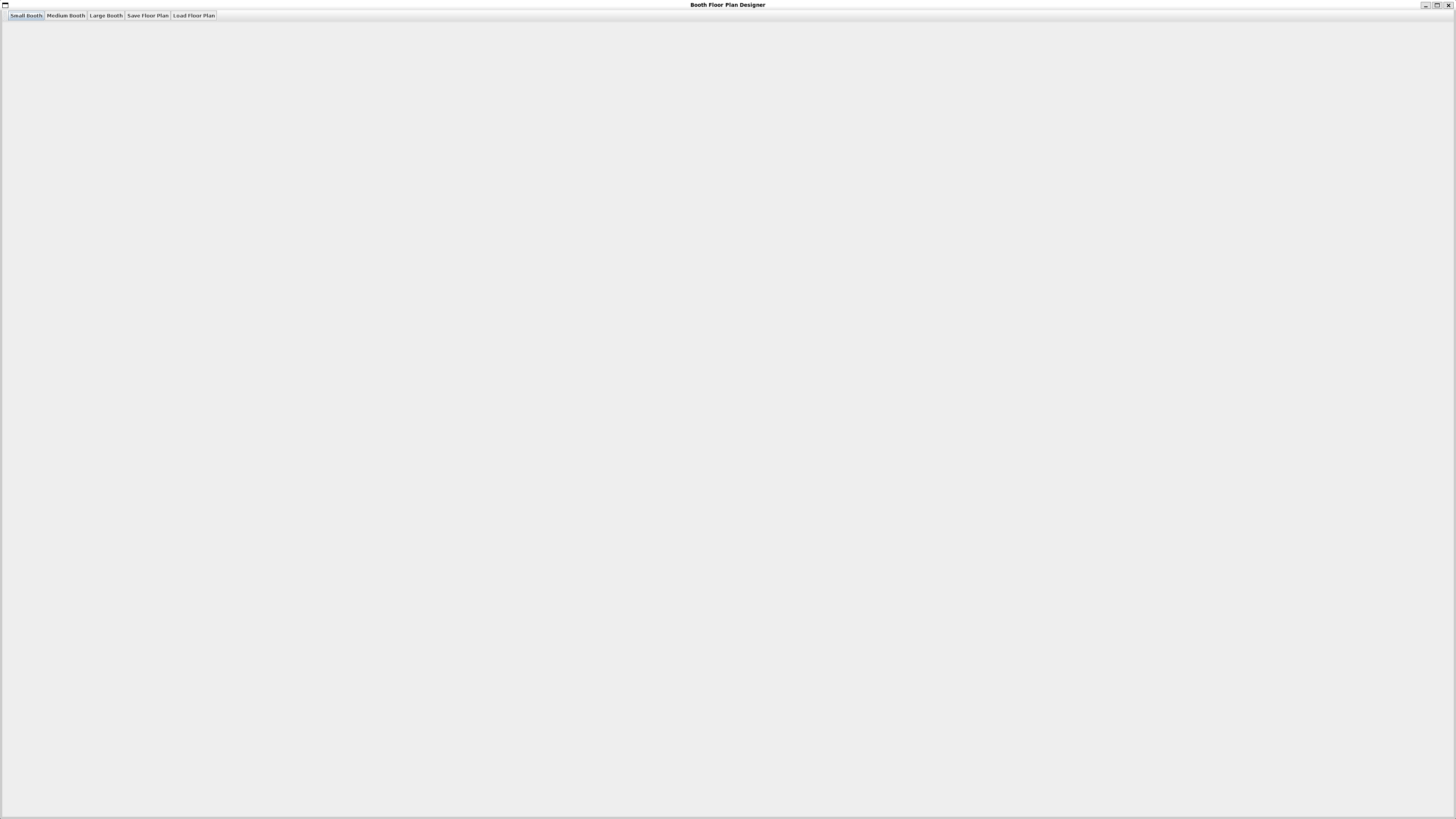
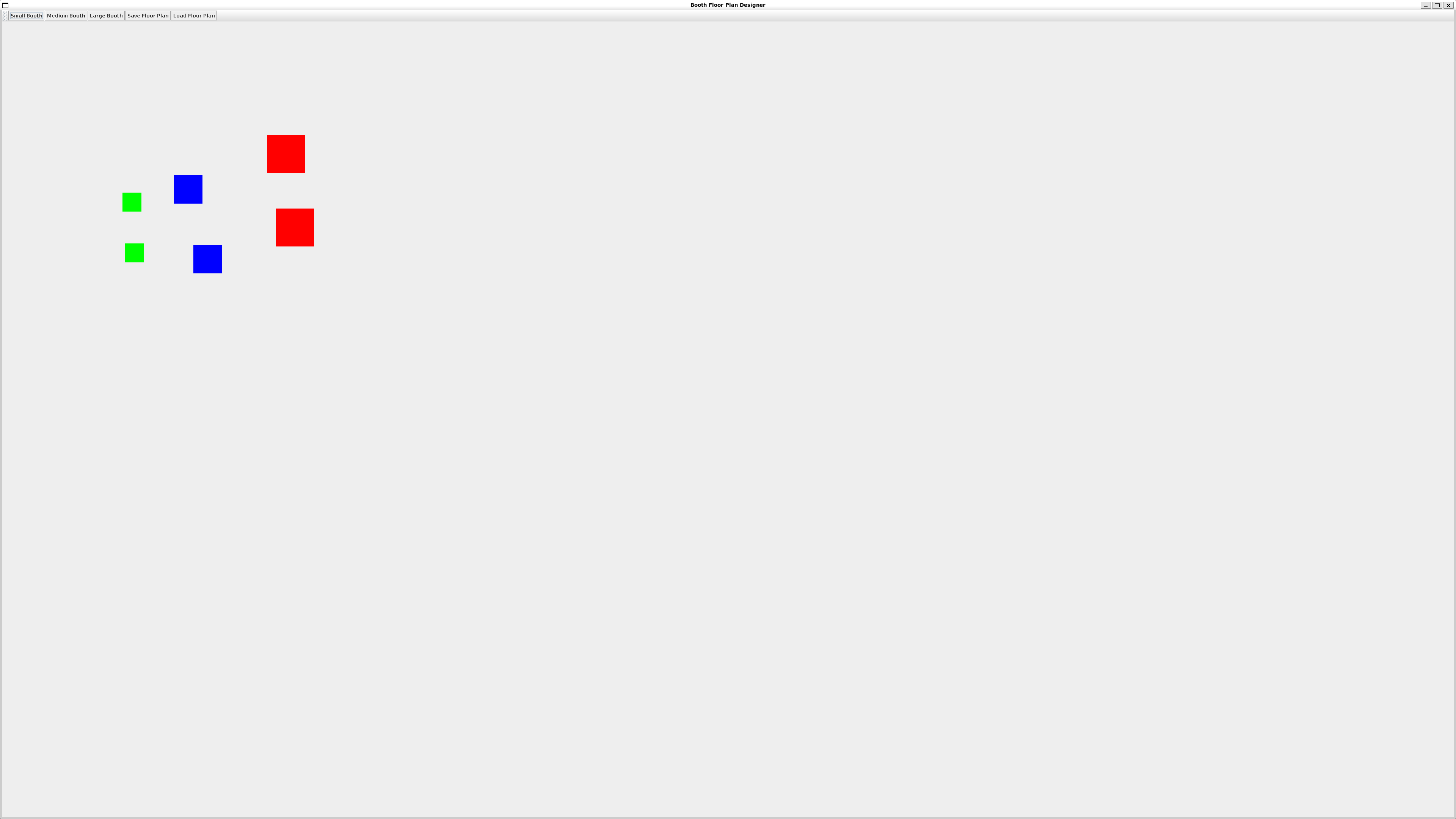
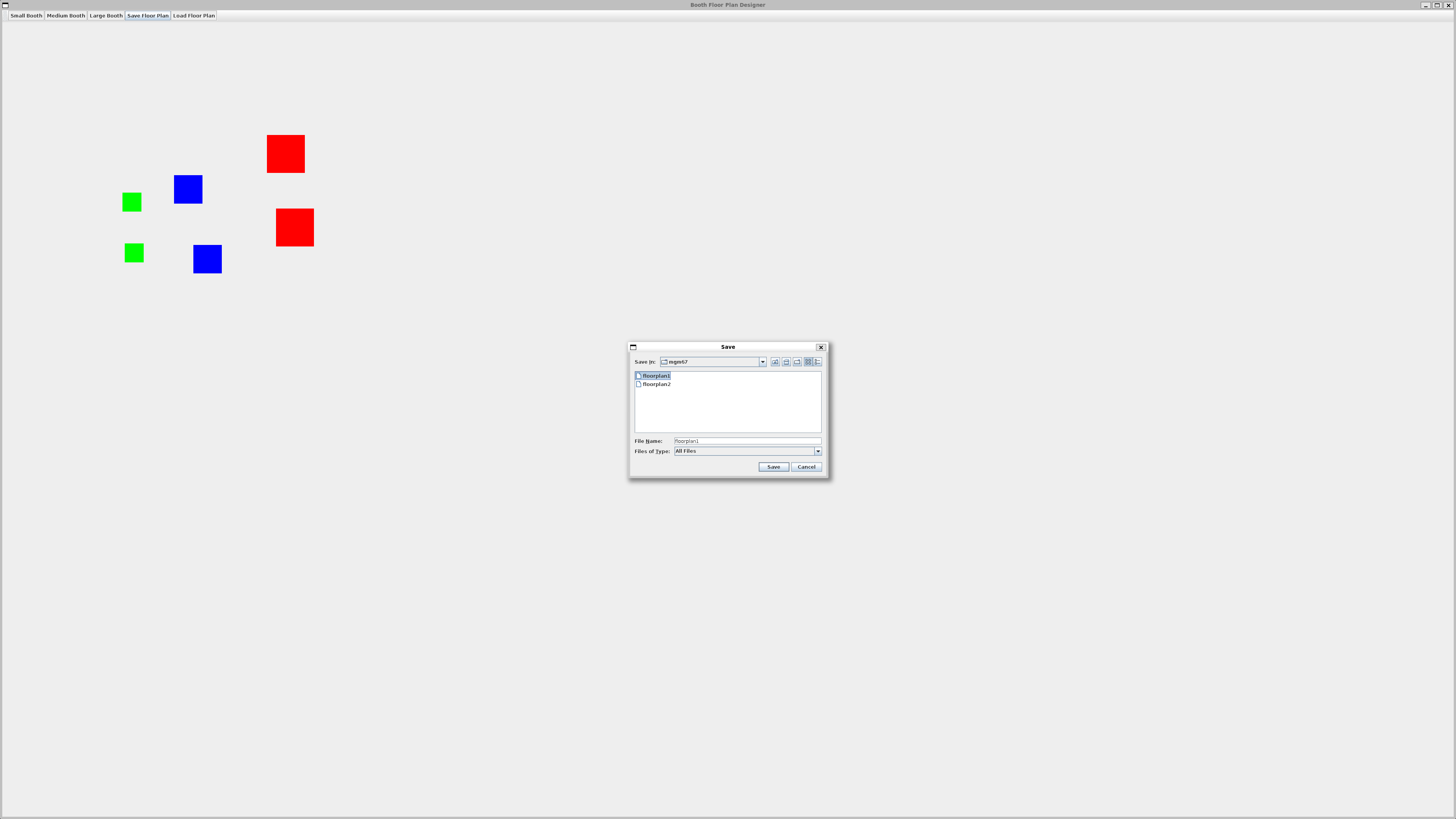
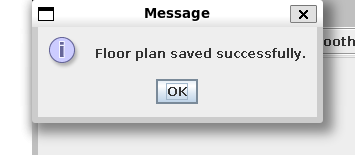
Link available [here](https://lucid.app/lucidchart/74c22401-a0bd-4ff7-a012-6ea922a1174f/edit?viewport_loc=-912%2C472%2C2491%2C1888%2C0_0&invitationId=inv_3bd0a7eb-39de-4b34-b457-1914fe704817)

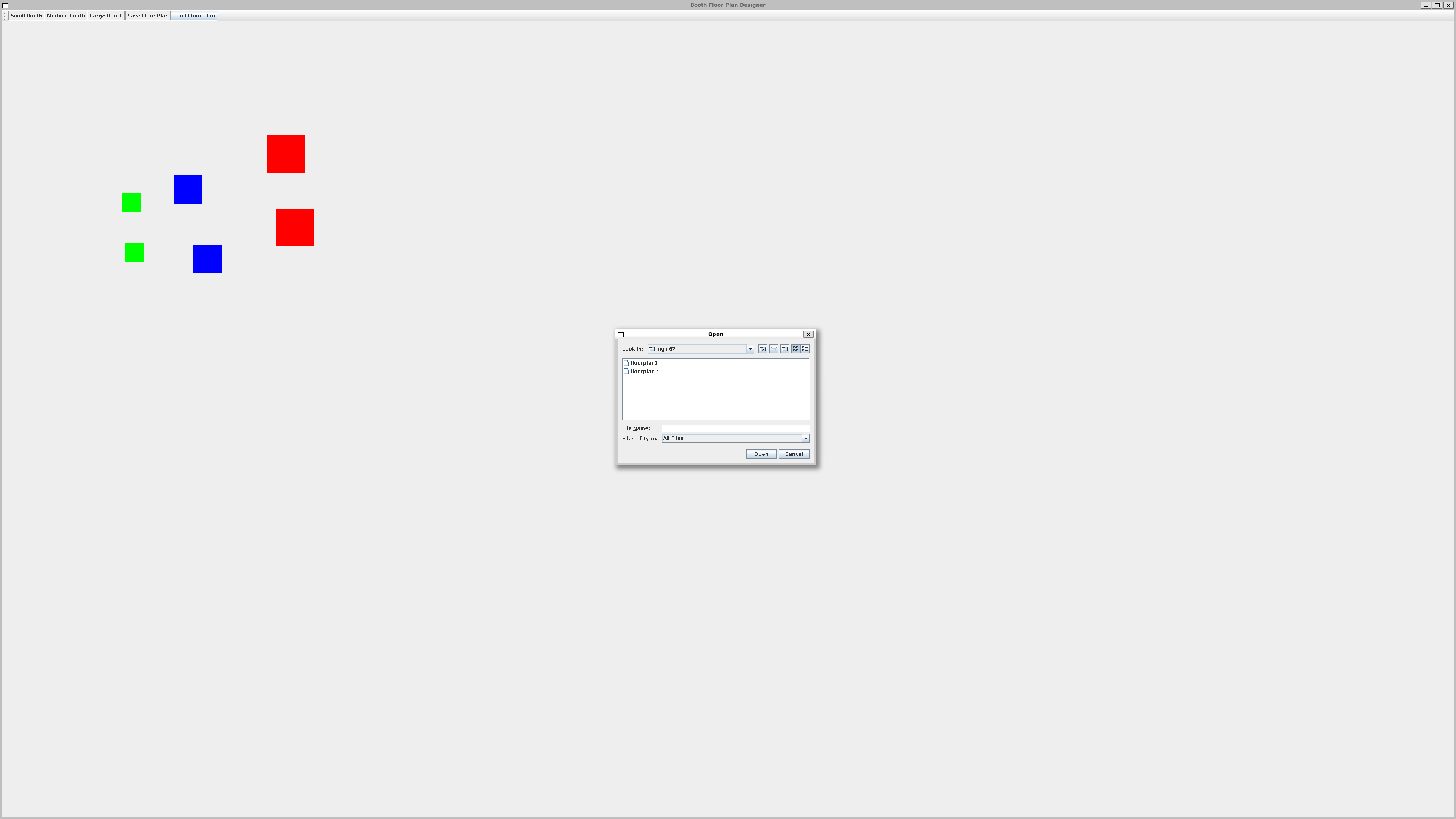


Step 9. Derive a design class diagram (DCD) from the design sequence diagram. That is, derive classes, methods and attributes of classes, and relationships between the classes.

  
Step 10. Implement all classes and all methods in the DCD. Compile, run and debug the software. Produce screen shots showing the working of the software.

GitHub: <https://github.com/mgm67671/booth-floor-plan>

Empty Floor Plan: Examples of all 3 booth sizes:Saving window:Saving confirmation:

Loading window:Loading Successful and confirmed: