Università degli Studi di Salerno

.DIEM

Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata

Corso di Laurea Magistrale in Ingegneria Informatica

Software Architecture Design 2024/2025
Canale I-Z

Project Work

Gruppo n. 02 – IZ

Cognome e Nome	Matricola	e-mail
Iannone Davide	0622702572	d.iannone19@studenti.unisa.it
Lomazzo Noemi	0622702663	n.lomazzo@studenti.unisa.it
Sirica Simone	0622702627	s.sirica5@studenti.unisa.it
Petrone Gianluca	0622702642	g.petrone27@studenti.unisa.it

Anno accademico 2024-2025

Summary

1. Pre-game

1.1. User stories, Acceptance Criteria & Tasks

US1

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** an empty window to be displayed when I start the application **So that** I can make new drawings	**Given** the installed application **When** I start the application **Then** an empty window is displayed	MUST	1	• Create an empty page of the program • Create a space for the tool bar	1

US2

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to be able to use the mouse on the window and choose the geometrical shape to add to my drawing **So that** I can position the geometrical form in the chosen position	**Given** a drawing screen **When** I click on the window **Then** the shape is drawn at the selected position	MUST	1	• Create an event handler for the mouse	1

US3

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to add line segments **So that** I can place line segments into my drawings	**Given** a list of shapes **When** I click 'Line segment' AND I click two points in the window	MUST	1	• Create a segment line button in the toolbar • Implement the logic for segment line creation • Implementation and	1

	Then a line segment, linking the points, is added to the drawing			execution of tests	

US4

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to add rectangles **So that** I can place rectangles into my drawings	**Given** a list of shapes **When** I click 'Rectangle' **Then** a rectangle is added to the drawing with a defined width and height	MUST	1	• Create rectangle shape button in the toolbar • Implement the logic for rectangle creation • Implementation and execution of tests	1

US5

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to add ellipses **So that** I can place ellipses into my drawings	**Given** a list of shapes **When** I click 'Ellipse' **Then** an ellipse is added to the drawing with a defined width and height	MUST	1	• Create ellipse shape button in the toolbar • Implement the logic for ellipse creation • Implementation and execution of tests	1

US6

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to choose the color of the geometrical shape **So that** I can color the border of the added shape	**Given** a selected shape **When** I open the color palette with exactly 8 predefined color options AND I choose one of this color **Then** the selected shape changes its border color	MUST	2	• Create a color icon for the color of the border of a shape • Add eight different colors in the color menu • Implement the logic that after selecting the color and placing a shape, the border of the shape is coloured by the selected one • Implementation and execution of tests	1

US7

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to choose the color of a closed shape **So that** I can color the interior of the shape	**Given** a selected closed shape **When** I open the color palette with exactly 8 predefined color options AND I choose	MUST	2	• Create a button for internal color • Add eight different color in the color menu • Implement the logic that after selecting the color and placing a closed shape, the interior of the shape is coloured by the selected one	1

	one of this color **Then** the interior selected closed shape color changes			• Implementation and execution of tests	

US8

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I wan**t to save the drawing on a file **So that** I don't lose the draw	**Given** a drawing **When** I click 'Save' **Then** the drawing is saved to a file	MUST	2	• Create the "Save" button in the toolbar • Implement save logic for the save button • Implementation and execution of tests	1

US9

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to load a drawing previously saved on a file **So that** I can continue modifying the drawing whenever I want	**Given** a drawing screen **When** I click 'Load' AND I select a file **Then** the selected drawing appears on the screen	MUST	2	• Create the "Load" button in the toolbar • Implement load logic for the load button • Implementation and execution of tests	1

US10

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to be able to select one shape	**Given** a drawing with shapes	MUST	1	• Implement the logic to manage the click on a shape	1

with the mouse **So that** I can perform any operation on the selected shape	**When** I click shape **Then** it becomes highlighted			• Implementation and execution of tests	

US11

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to delete a selected shape **So that** I can remove the shape from the drawing	**Given** a selected shape **When** I click 'Delete' **Then** the shape is removed	SHOULD	2	• Create a menu for right click on the window • Create "Delete" button in the right click menu • Implement the logic to delete the shape • Implementation and execution of tests	1

US12

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to move a selected shape in a different position **So that** I can modify the position of a shape	**Given** a selected shape **When** I drag it to a new position **Then** the shape is moved into the new location	SHOULD	2	• Add handler for mouse drag functions • Implementation and execution of tests	1

US13

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to change the color of a	**Given** a selected shape	SHOULD	2	• Implement the logic for changing the color of a	1

selected shape **So that** I can choose an appropriate color for the drawing	**When** I choose a new color from the palette **Then** the shape color is updated			placed shape • Implementation and execution of tests	

US14

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to change the size of a selected shape **So that** I can resize the shape	**Given** a selected shape **When** I click on a placed shape AND modify its dimensions in the specific boxes **Then** size shape is updated	SHOULD	3	• Create size field in the toolbar • Implement the logic to manage the resize of a shape • Implementation and execution of tests	1

US15

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to use a cut operation on a shape **So that** I can remove and possibly paste the shape	**Given** a selected shape **When** I click ‘Cut’ **Then** the selected shape is removed and copied to the clipboard AND possibly paste it in a	SHOULD	2	• Create the “Cut” button in the right click menu • Implement the logic linked to the cut button to cut a shape • Implementation and execution of tests	1

	different place				

US16

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to use a copy operation on a shape **So that** I can copy an already placed shape	**Given** a selected shape **When** I click 'Copy' **Then** the selected shape is copied to the clipboard AND possibly paste it in a different position	SHOULD	2	• Create the "Copy" button in the right click menu • Implement the logic linked to the copy button to copy a shape • Implementation and execution of tests	1

US17

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to use a paste operation **So that** I can place again a copied shape	**Given** a drawing screen **When** I click 'Paste' **Then** the shape on the clipboard is added to the drawing	SHOULD	2	• Create the "Paste" button in the right click menu • Implement the logic linked to the paste button to paste a shape • Implementation and execution of tests	1

US18

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** that my operations are	**Given** an action has occurred **When** I click	SHOULD	3	• Create the "undo" button in the interface	1

defeasible **So that** I can undo my operations	the undo arrows **Then** I can undo the last operation AND previous state is restored			• Implement the logic of the button, that delete the last operation done • Implementation and execution of tests	

US19

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to send a shape to the front **So that** I can order the overlapping shapes	**Given** the overlapping shapes **When** I click 'To the front' **Then** the selected shape moves to the front	SHOULD	3	• Create the "to the front" option when a shape is selected • Implement the logic that places the selected shape to the front • Implementation and execution of tests	2

US20

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to send a shape to the back **So that** I can order the overlapping shapes	**Given** overlapping shapes **When** I click 'To the back' **Then** the selected shape moves to the back	SHOULD	3	• Create the "to the back" option when a shape is selected • Implement the logic that places the selected shape to the back • Implementation and execution of tests	2

US21

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to zoom the interface **So that** I can see better my drawing	**Given** a drawing **When** I click on the lens icon AND I can zoom up to 4 levels in or out **Then** the view zooms in or out accordingly	COULD	3	• Create the “zoom in” and “zoom out” buttons in the interface • Implement “zoom in” button that, for every click, zoom in the interface • Implement the “zoom out” button that, for every click zooms out on the interface • Implementation and execution of tests	2

US22

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to use the scroll bar **So that** I can see the other parts of the drawing	**Given** a drawing screen larger than window **When** I click and drag the scroll bar **Then** the other parts of the drawing become visible	COULD	2	• Create a horizontal scrollbar at the bottom of the window • Create a vertical scrollbar at the right of the window • Implementation and execution of tests	2

US23

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to activate and deactivate the grid **So that** I can place the shapes more easily	**Given** a drawing screen **When** I click on the grid icon **Then** the grid appears if deactivated or disappears if activated	COULD	3	• Create a “toggle grid” button in the toolbar • Implement the logic of the button that shows or hides a grid • Implementation and execution of tests	2

US24

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to choose the grid size **So that** I can draw more precisely	**Given** an active grid **When** I select a new grid size **Then** the grid spacing updates accordingly	COULD	3	• Create a “grid size” option in the toolbar • Implement the logic that change the size of the grid • Implementation and execution of tests	2

US25

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to enter an arbitrary polygon **So that** I can make more complex drawing	**Given** a drawing screen **When** I click ‘Polygon’ AND I click on different points on the drawing **Then** these will be connected	COULD	5	• Create a “Polygon” option in the toolbar • Create the logic that connects the selected points creating a closed shape • Implementation	2

	by lines that will be visible on the drawing sheet			and execution of tests	

US26

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to add a text string as a shape **So that** I can include titles or notes where needed	**Given** a drawing screen **When** I select the text string shape AND I write it AND I choose the size **Then** it will be added as a shape	COULD	3	• Create a “text” option in the toolbar • Create a field to write a text string • Create a field to choose the size of a text string • Implement the logic that places a text in the chosen point • Implementation and execution of tests	2

US27

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to rotate a shape by an arbitrary angle **So that** I can change the shape direction	**Given** a selected shape **When** I click ‘Rotation’ AND I set a rotation angle **Then** the shape rotates accordingly	COULD	8	• Create a “Rotate” option in the right click menu of a shape • Create a field in the “Rotate” option where an angle can be specified • Implement the logic that rotates the selected shape at the chosen angle • Implementation and execution of tests	2 partially + 3 partially

US28

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to mirror a shape horizontally or vertically **So that** I can mirror the shape more easily	**Given** a selected shape **When** I click 'Mirror horizontal' **Then** the shape mirrors with a vertical axis of symmetry. **Given** a selected shape **When** click "Mirror vertical" **Then** the shape mirrors with a horizontal axis of symmetry.	COULD	8	• Create the "Mirror horizontal" and "Mirror vertical" options in the right click menu of a shape • Implement the logic that mirrors the shape horizontally or vertically • Implementation and execution of tests	3

US29

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to stretch a shape horizontally or vertically **So that** I can modify the shape appearance	**Given** a selected shape **When** I stretched it in a direction (horizontally or vertically) **Then** it changes accordingly	COULD	8	• Create the "Stretch" options in the right click menu of a shape • Implement the logic that stretches the shape • Implementation and execution of tests	3

US30

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to be able to select more shapes with the mouse **So that** I can perform any operations on the selected shapes	**Given** a set of shapes **When** I select multiple shapes **Then** all selected shapes can be moved or modified together	COULD	3	• Implement the logic that can select more than one shape • Implementation and execution of tests	3

US31

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to group more than one selected shape **So that** I can do the same operations on all the selected shapes	**Given** a set of shapes **When** I select more than one and click 'Group' **Then** I group the selected shapes AND perform operations on them as if they were a single shape	WON'T	5	• Create a "Group" option in the right click menu of a shape • Implement the logic that performs the same operations on the selected shape • Implementation and execution of tests	3

US32

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to ungroup the selected	**Given** a selected grouped shape **When** I click	WON'T	5	• Create an "Ungroup" option in the	3

shapes **So that** I can operate on a single shape	'Ungroup' **Then** the shapes become separate elements again			right click menu of a shape ● Implementation and execution of tests	

US33

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to add new shape creation commands **So that** I can create my own shapes	**Given** a drawing **When** I click 'New Shape Command' AND assign it a name **Then** the shape is saved in the command library with all its properties **Given** a created shape command **When** I click its associated button **Then** a copy of the original shape is inserted into the drawing exactly as it was when the command was created (not the current state if the shape has since changed or been deleted)	WON'T	8	● Create a "New shape commands" option in the toolbar ● Create a field to write the name of the shape ● Implement the logic that saves a shape with the same properties as when it was created ● Implementation and execution of tests	3

US34

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to export a library of shapes created **So that** I can save my personal shapes	**Given** the saved shapes **When** I click 'Export library' **Then** a 'shapes library file' is created	WON'T	8	• Create a "Export library" button in the toolbar • Implement the logic that saves the personal shapes in a file • Implementation and execution of tests	3 partially

US35

Description	Acceptance Criteria	Priority	Story Points	Tasks	Sprint n.
As a user **I want** to import a library of shapes created **So that** I can reuse my personal shapes	**Given** an existing 'shapes library file' **When** I click 'Import library' **Then** the saved shapes appear	WON'T	8	• Create a "Import library" button in the toolbar • Implement the logic that import the personal shapes from a saved file • Implementation and execution of tests	3 partially

N.B. For the breakdown of the 35 User Stories into three Sprints, the team's capacity was estimated based on an average availability of about 32 hours per Sprint (4 team members × approx. 8 hours). For simplicity and planning purposes, we approximated 1 Story Point to roughly 1 hour of work, even though Story Points primarily represent relative effort and complexity, not time. Based on this, we distributed around 32 Story Points per Sprint to match the team's estimated capacity.

Since the total estimated points for all User Stories is about 118, it was necessary to select the main features to be implemented within the three Sprints, distributing about 32 Story Points for each.

In the first Sprint, the basic User Stories were included, which are necessary to allow the user to create, place and color simple shapes, as well as to save and load the drawing, editing operations (cut, copy, paste). Then US n. 1 through n. 18 will be implemented.

In the second Sprint, the User Stories inherent functionalities related to the management of the arrangement of overlapping shapes, supporting functionalities such as zoom and grid were provided. In addition, features such as text input, polygon creation and some of the figure distortion features will be handled.
Then US n. 19 through n. 26 will be implemented and n. 27 partially will be implemented.

Finally, the third Sprint was devoted to more advanced and complex features, such as figure grouping and custom shape export/import options.
Thus, will be completed US n. 27 and US n. 28 through n. 32 will be implemented.
US n. 33, n. 34 and n. 35 will, in addition, be initiated and remain in the backlog for possible later implementation.
This ensured a balanced distribution of the workload among the three Sprints, respecting the team's capabilities and the goal of implementing the main functionalities within the available time frame.

Sprint n.	**Total Story Points for Sprint**
1	32
2	32
3	30 + 2
	96

1.2. Definition of Done

- The user story implementation meets ALL the acceptance criteria.
- The unit tests were written, executed and passed.
- All acceptance criteria have at least a test case.
- End-user documentation is available.
- All the code has been properly documented.
- All bugs fixed.
- The interface is functional and consistent with the design.

1.3. 1st sprint planning and backlog

#US	Task	Assigned To	Status
1	● Create an empty page of the program ● Create a space for the tool bar	Iannone Davide	In Progress
2	● Create an event handler for the mouse	Petrone Gianluca	In Progress
3	● Create a segment line button in the toolbar ● Implement the logic for segment line creation ● Implementation and execution of tests	Interface: Iannone Davide Logic: Petrone Gianluca	In Progress
4	● Create rectangle shape button in the toolbar ● Implement the logic for rectangle creation ● Implementation and execution of tests	Interface: Iannone Davide Logic: Petrone Gianluca	In Progress
5	● Create ellipse shape button in the toolbar ● Implement the logic for ellipse creation ● Implementation and execution of tests	Interface: Iannone Davide Logic: Petrone Gianluca	In Progress
6	● Create a color icon for the color of the border of a shape ● Add eight different colors in the color menu	Interface: Iannone Davide Logic: Petrone Gianluca	In Progress

	● Implement the logic that after selecting the color and placing a shape, the border of the shape is coloured by the selected one ● Implementation and execution of tests		
7	● Create a button for internal color ● Add eight different color in the color menu ● Implement the logic that after selecting the color and placing a closed shape, the interior of the shape is coloured by the selected one ● Implementation and execution of tests	Interface: Iannone Davide Logic: Petrone Gianluca	In Progress
8	● Create the “Save” button in the toolbar ● Implement save logic for the save button ● Implementation and execution of tests	Interface: Iannone Davide Logic: Petrone Gianluca	In Progress
9	● Create the “Load” button in the toolbar ● Implement load logic for the load button ● Implementation and execution of tests	Interface: Iannone Davide Logic: Petrone Gianluca	In Progress
10	● Implement the logic to manage the click on a shape ● Implementation and execution of tests	Lomazzo Noemi	In Progress
11	● Create a menu for right click on the window ● Create “Delete” button in the right click menu ● Implement the logic to delete the shape ● Implementation and execution of tests	Interface: Sirica Simone Logic: Lomazzo Noemi	In Progress

12	● Add handler for mouse drag functions ● Implementation and execution of tests	Lomazzo Noemi	In Progress
13	● Implement the logic for changing the color of a placed shape ● Implementation and execution of tests	Lomazzo Noemi Sirica Simone	In Progress
14	● Create size field in the toolbar ● Implement the logic to manage the resize of a shape ● Implementation and execution of tests	Interface: Iannone Davide Logic: Petrone Gianluca	In Progress
15	● Create the “Cut” button in the right click menu ● Implement the logic linked to the cut button to cut a shape ● Implementation and execution of tests	Interface: Sirica Simone Logic: Lomazzo Noemi	In Progress
16	● Create the “Copy” button in the right click menu ● Implement the logic linked to the copy button to copy a shape ● Implementation and execution of tests	Interface: Sirica Simone Logic: Lomazzo Noemi	In Progress
17	● Create the “Paste” button in the right click menu ● Implement the logic linked to the paste button to paste a shape ● Implementation and execution of tests	Interface: Sirica Simone Logic: Lomazzo Noemi	In Progress
18	● Create the “undo” button in the interface ● Implement the logic of the button, that delete the last operation done ● Implementation and execution of tests	Sirica Simone	In Progress

19	● Create the “to the front” option when a shape is selected ● Implement the logic that places the selected shape to the front ● Implementation and execution of tests	-	To Do
20	● Create the “to the back” option when a shape is selected ● Implement the logic that places the selected shape to the back ● Implementation and execution of tests	-	To Do
21	● Create the “zoom in” and “zoom out” buttons in the interface ● Implement “zoom in” button that, for every click, zoom in the interface ● Implement the “zoom out” button that, for every click zooms out on the interface ● Implementation and execution of tests	-	To Do
22	● Create a horizontal scrollbar at the bottom of the window ● Create a vertical scrollbar at the right of the window ● Implementation and execution of tests	-	To Do
23	● Create a “toggle grid” button in the toolbar ● Implement the logic of the button that shows or hides a grid ● Implementation and execution of tests	-	To Do
24	● Create a “grid size” option in the toolbar ● Implement the logic that	-	To Do

	change the size of the grid ● Implementation and execution of tests		
25	● Create a “Polygon” option in the toolbar ● Create the logic that connects the selected points creating a closed shape ● Implementation and execution of tests	-	To Do
26	● Create a “text” option in the toolbar ● Create a field to write a text string ● Create a field to choose the size of a text string ● Implement the logic that places a text in the chosen point ● Implementation and execution of tests	-	To Do
27	● Create a “Rotate” option in the right click menu of a shape ● Create a field in the “Rotate” option where an angle can be specified ● Implement the logic that rotates the selected shape at the chosen angle ● Implementation and execution of tests	-	To Do
28	● Create the “Mirror horizontal” and “Mirror vertical” options in the right click menu of a shape ● Implement the logic that mirrors the shape horizontally or vertically ● Implementation and execution of tests	-	To Do
29	● Create the “Stretch” options in the right click menu of a	-	To Do

	shape ● Implement the logic that stretches the shape ● Implementation and execution of tests		
30	● Implement the logic that can select more than one shape ● Implementation and execution of tests	-	To Do
31	● Create a “Group” option in the right click menu of a shape ● Implement the logic that performs the same operations on the selected shape ● Implementation and execution of tests	-	To Do
32	● Create an “Ungroup” option in the right click menu of a shape ● Implementation and execution of tests	-	To Do
33	● Create a “New shape commands” option in the toolbar ● Create a field to write the name of the shape ● Implement the logic that saves a shape with the same properties as when it was created ● Implementation and execution of tests	-	To Do
34	● Create a “Export library” button in the toolbar ● Implement the logic that saves the personal shapes in a file ● Implementation and execution of tests	-	To Do
35	● Create a “Import library”	-	To Do

	button in the toolbar ● Implement the logic that import the personal shapes from a saved file ● Implementation and execution of tests		

1.4. Technologies, Languages, Frameworks & Conventions

Technologies and tools
Programming Language -> Java
Framework -> JavaFX
IDE -> Netbeans
Gui Builder -> Scene Builder
Testing -> JUnit (via Maven)
Board -> Trello

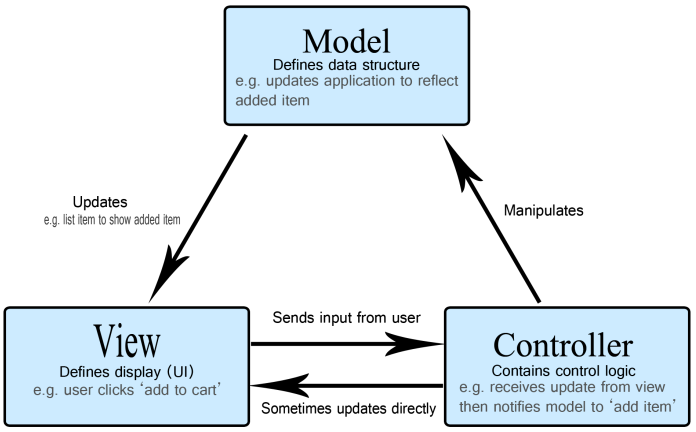
Conventions

- Standard Java style (camelCase for methods and variables, PascalCase for classes)
- Variable names in English, meaningful and consistent

1.5. Description of software architecture and selected design pattern

The architecture we use in this software is the MVC (Model - View - Controller). The MVC consists in three different parts:

- View, that renders the model and corresponds to the User Interface where the user can interact with.
- Model, that represents the data and encapsulates the application state. It notifies the view of changes.
- Controller, that defines the application behaviour and updates the model according to the user actions.



We adopt this pattern to separate model (data) from view to improve the separation of concerns and to achieve loose coupling.

To increase understanding of the basic principles of object-oriented design and to make software systems easier to modify while increasing maintainability, certain design patterns were chosen.
In particular:

- Factory for the creation of the shapes. The **ShapeFactory** class takes care of building the shapes requested by the user, preventing the constructors of specific classes from being used directly in the code. This way, if we want to add new shapes in the future, we can easily do so without modifying other parts of the program;
- Operations such as adding, moving, deleting or coloring a shape are handled by **command** objects. Each operation is saved and can be undone or repeated (undo/redo functionality);
- To allow multiple shapes to be grouped and treated as a single entity (e.g., to move them together), we used the **Composite** pattern. The ShapeGroup class contains multiple Shape objects within it, but it can be used as if it were a normal shape. This allows us to apply the same operations to both individual shapes and groups;
- The **Decorator** pattern allowed us to add extra functionality to shapes, such as border color or fill color;
- For transformations (e.g. rotating or stretching a shape) we used the **Strategy** pattern, so that each transformation is treated as a separate strategy.

1.6. Mock-up Interface

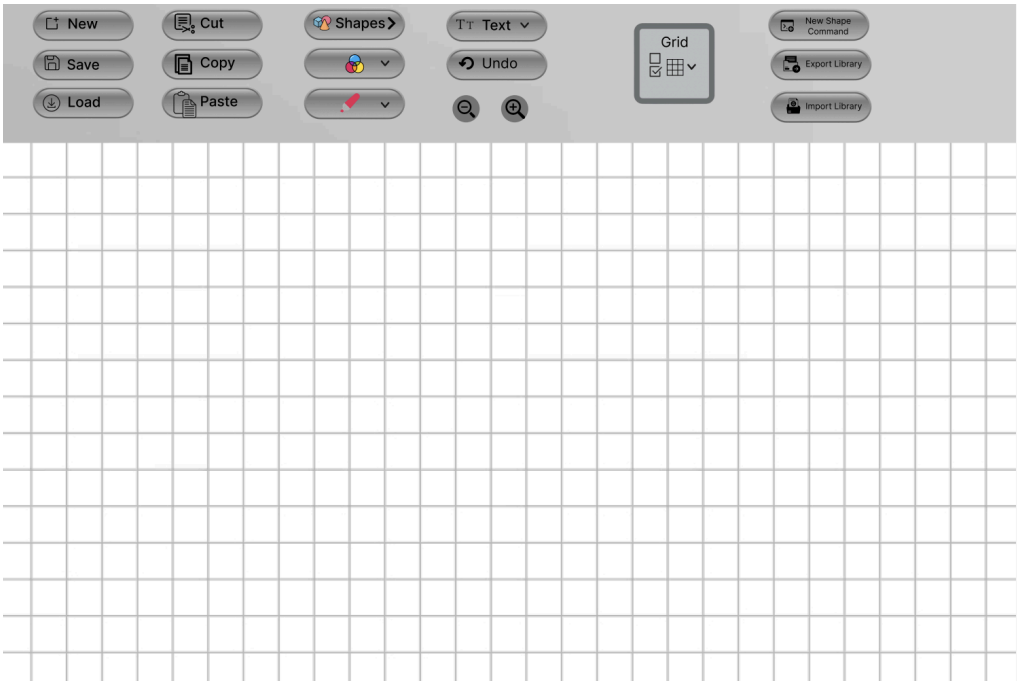


Figura 1: Empty canvas

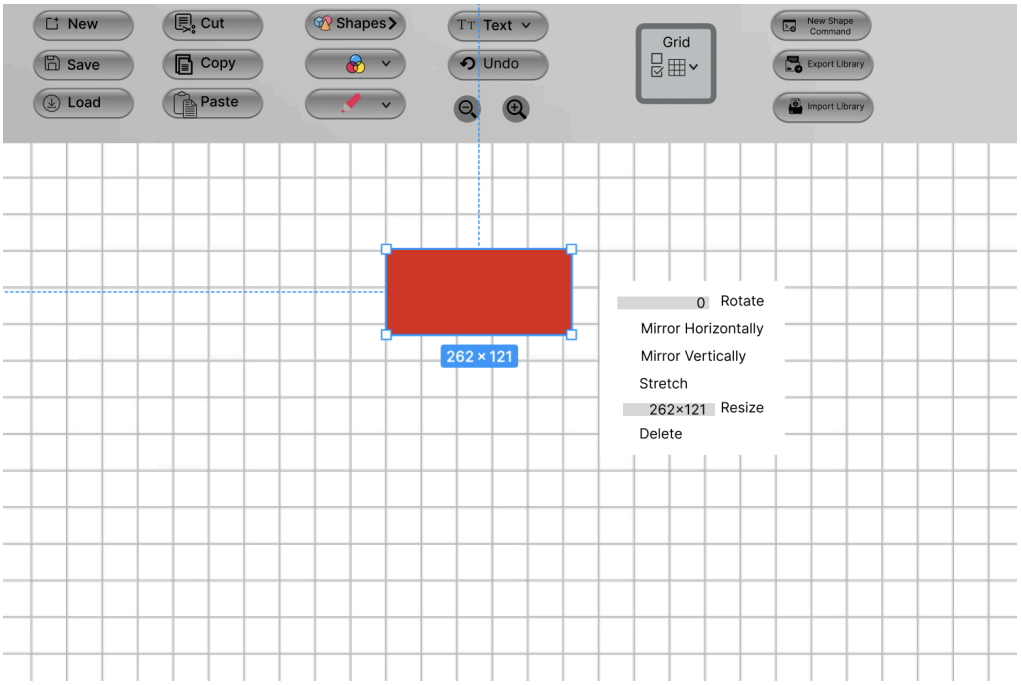


Figura 2: Simple shape options

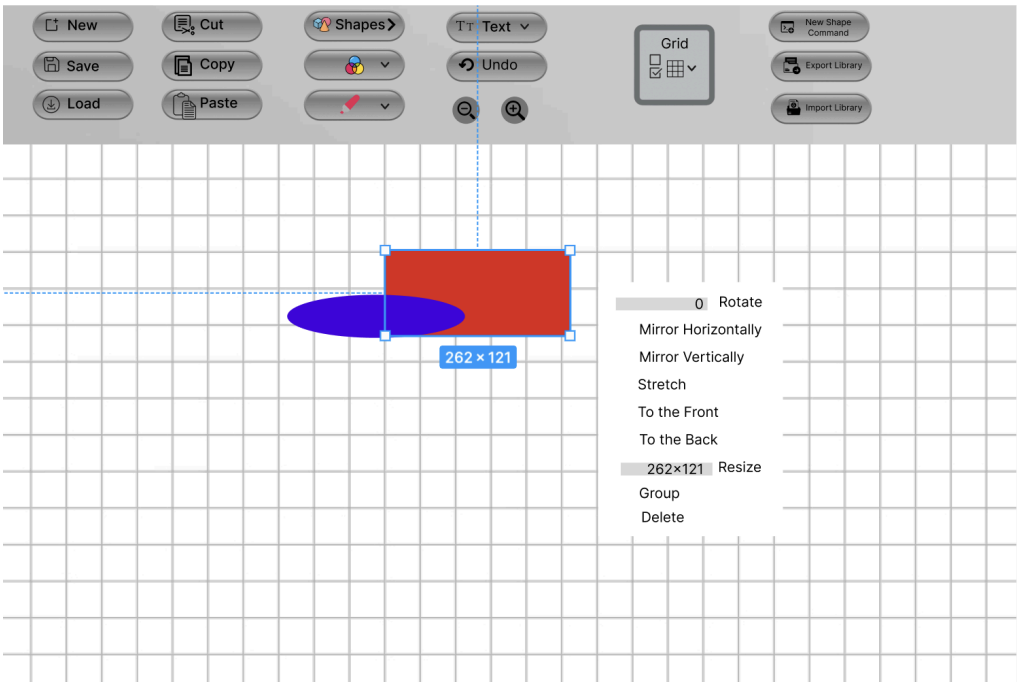


Figura 3: Overlap of simple shapes

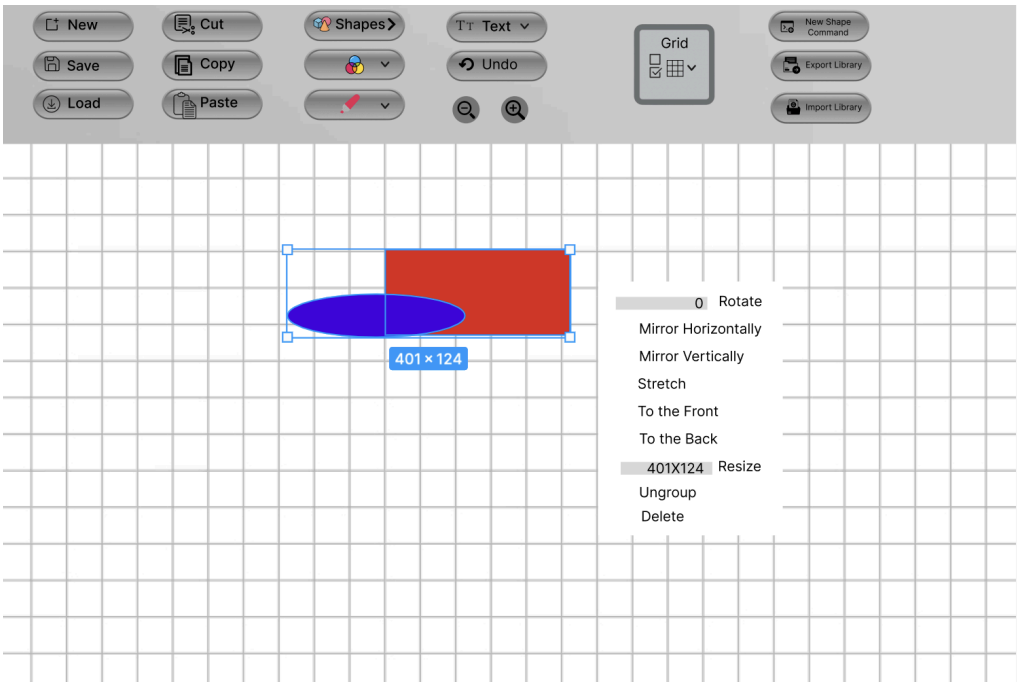


Figura 4: Grouping of simple shapes

2. 1st Sprint Release

3. 2nd Sprint Release

4. 3rd Sprint (Final) Release