

# UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA INFORMATICA ED ELETTRICA E MATEMATICA APPLICATA



Laurea Magistrale in Ingegneria Informatica

## **Relazione Project Work - Software Architecture & Design - ShapeTastic**

**Gruppo N°16 - AH**

### **Second Sprint Retrospective**

#### **Studenti:**

Cutolo Ciro	0622702532	<a href="mailto:c.cutolo7@studenti.unisa.it">c.cutolo7@studenti.unisa.it</a>
Frasca Gaetano	0622702610	<a href="mailto:g.frasca10@studenti.unisa.it">g.frasca10@studenti.unisa.it</a>
D'Ambrosio Gennaro	0622702464	<a href="mailto:g.dambrosio65@studenti.unisa.it">g.dambrosio65@studenti.unisa.it</a>
Bottiglieri Alessio	0622702583	<a href="mailto:a.bottiglieri16@studenti.unisa.it">a.bottiglieri16@studenti.unisa.it</a>

## Sommario

1	Pre-game .....	4
1.1	Initial product backlog .....	4
1.1.1	Basic operation of the program .....	4
1.1.2	Shape editing .....	7
1.1.3	Display of the drawing .....	10
1.1.4	More shapes and editing operations .....	13
1.1.5	Groups and shape libraries .....	15
1.2	Definition of Done .....	17
1.3	Description of software architecture .....	18
1.3.1	Architettura .....	18
1.4	Architecture and technologies chosen .....	19
1.5	Mock-up interface .....	19
2	First Sprint .....	20
2.1	First sprint Planning .....	20
2.1.1	Stima iniziale della Project Velocity .....	20
2.1.2	First Sprint Backlog .....	20
2.1.3	Suddivisione in task e assegnazione .....	21
2.2	First Sprint - Applicazione Pattern .....	26
2.2.1	Simple Factory Pattern .....	26
2.2.2	Prototype Pattern .....	27
2.3	First Sprint - Testing .....	28
2.3.1	US1: Finestra iniziale .....	28
2.3.2	US2: Forme supportate .....	29
2.3.3	US3: Aggiunta forme geometriche .....	30
2.3.4	US4: Scelta colore del bordo delle forme .....	31
2.3.5	US5: Scelta colore di riempimento delle forme .....	33
2.3.6	US6: Salvataggio di un disegno .....	35
2.3.7	US7: Caricamento di un disegno .....	37
2.3.8	US8: Selezione di una Forma .....	39
2.3.9	US9 : Eliminazione di una Forma .....	40
2.3.10	US10: Copia e Incolla .....	42
2.3.11	US11: Taglia .....	43
2.4	First Sprint Backlog .....	45
2.5	Updated Product Backlog .....	46

2.6	Project Burndown chart: .....	47
2.7	First Sprint - Review .....	48
2.7.1	Project velocity measured: .....	48
2.7.2	Problemi riscontrati durante l'implementazione .....	48
2.7.3	Modifiche da apportare al Product Backlog .....	48
2.8	First Sprint - Retrospective.....	49
2.8.1	Starfish Diagram .....	49
3	Second sprint .....	50
3.1	Second Sprint Planning .....	50
3.1.1	Modifiche Product Backlog.....	50
3.1.2	Stima aggiornata della Project Velocity.....	51
3.1.3	Selezione delle User Stories per la seconda sprint .....	51
3.1.4	Suddivisione in task e assegnazione .....	52
3.2	Second Sprint - Applicazione Pattern .....	56
3.2.1	Command Pattern .....	56
3.3	Second Sprint - Testing .....	58
3.3.1	US12: Undo .....	58
3.3.2	US13: Modifica della posizione di una forma .....	59
3.3.3	US14: Modifica del colore di una forma .....	60
3.3.4	US15: Modifica delle dimensioni di una forma.....	61
3.3.5	US16: Modifica del livello di una forma .....	61
3.3.6	US17: Modifica del livello di zoom.....	62
3.3.7	US18: Area di disegno espandibile .....	63
3.3.8	US19: Scroll area di disegno.....	64
3.3.9	US20: Visualizzazione della griglia .....	65
3.3.10	US21: Cambiare dimensioni griglia .....	66
3.4	Second Sprint Backlog.....	67
3.5	Updated Product Backlog .....	68
3.6	Project Burndown chart: .....	69
3.7	Second Sprint - Review .....	70
3.7.1	Project velocity measured: .....	70
3.7.2	Problemi riscontrati durante l'implementazione .....	70
3.7.3	Modifiche da apportare al Product Backlog .....	70
3.8	Second Sprint - Retrospective.....	71
3.8.1	Starfish Diagram .....	71

# 1 Pre-game

Link utili:

- Trello Board: <https://trello.com/b/1OiONvWL/pw16-sad>
- Repository GitHub: [https://github.com/CiroCutolo/Group16\\_SoftwareArchitecture-Design\\_ProjectWork](https://github.com/CiroCutolo/Group16_SoftwareArchitecture-Design_ProjectWork)

## 1.1 Initial product backlog

### 1.1.1 Basic operation of the program

#### 1.1.1.1 Finestra iniziale

**Story:**

Come utente,  
Vorrei vedere una finestra inizialmente vuota,  
Così che possa iniziare a disegnare su di essa.

**Acceptance Criteria:**

- Dato che l'applicazione è stata appena avviata,  
Quando l'interfaccia utente viene caricata,  
Allora l'utente vede una finestra inizialmente vuota.

**Priority:** Alta

**Story Points:** 1

#### 1.1.1.2 Forme supportate

**Story:**

Come utente,  
Vorrei poter scegliere tra segmento di linea, rettangolo ed ellisse,  
Così che possa creare disegni utilizzando queste forme.

**Acceptance Criteria:**

- Dato che l'interfaccia è visibile,  
Quando l'utente guarda la sezione con l'elenco delle forme,  
Allora vede tre opzioni: linea, rettangolo ed ellisse,

**Priority:** Alta

**Story Points:** 3

#### 1.1.1.3 Aggiunta forme geometriche

**Story:**

Come utente,  
Vorrei poter selezionare una forma geometrica tra le disponibili e posizionarla con il mouse,  
Così che possa aggiungere quella forma in un punto specifico della finestra.

**Acceptance Criteria:.**

- Dato che l'utente non ha selezionato alcuna forma,  
Quando effettua un clic sull'area di disegno,  
Allora non deve essere eseguita alcuna azione e non deve comparire nessuna anteprima.
- Dato che l'utente ha selezionato una forma tra quelle disponibili,  
Quando preme il tasto sinistro del mouse sull'area di disegno,  
Allora deve iniziare la fase di disegno, memorizzando le coordinate iniziali del puntatore.
- Dato che l'utente sta trascinando il mouse dopo aver cliccato sull'area di disegno,  
Quando sposta il cursore tenendo premuto il tasto,  
Allora deve essere mostrata in tempo reale un'anteprima visiva della forma, aggiornata dinamicamente in base alla posizione del cursore.
- Dato che l'utente ha rilasciato il mouse e la forma è stata inserita,  
Quando termina l'interazione di disegno,  
Allora l'anteprima temporanea deve scomparire, lasciando visibile solo la forma definitiva.
- Dato che l'utente ha completato il trascinamento,  
Quando rilascia il tasto sinistro del mouse,  
Allora la forma selezionata deve essere inserita in modo definitivo nell'area di disegno, con le dimensioni e la posizione corrispondenti all'azione eseguita.

**Priority:** Alta

**Story Points:** 3

#### *1.1.1.4 Scelta colore del bordo delle forme*

##### **Story:**

Come utente,  
Vorrei poter scegliere il colore del bordo della forma da aggiungere,  
Così che possa personalizzarne l'aspetto.

##### **Acceptance Criteria:**

- Dato che l'interfaccia è visibile, quando l'utente guarda la sezione "Colore bordo", allora vede un selettore contenente almeno 8 colori
- Dato che una forma non è ancora stata disegnata, quando l'utente seleziona un colore per il bordo, allora il colore selezionato viene associato alla forma da disegnare
- Dato che l'utente ha selezionato un colore per il bordo, quando disegna una nuova forma, allora la forma presenta un bordo del colore scelto

**Priority:** Alta

**Story Points:** 3

#### *1.1.1.5 Scelta colore di riempimento delle forme*

##### **Story:**

Come utente,  
Vorrei scegliere il colore di riempimento delle forme chiuse (come l'ellissi),  
Così che possa personalizzarne l'aspetto.

**Acceptance Criteria:**

- Dato che l'interfaccia è visibile, quando l'utente guarda la sezione "Colore Riempimento", allora vede un selettore contenente almeno 8 colori
- Dato che l'interfaccia è visibile, quando seleziona un colore di riempimento nell'apposita sezione, allora il colore viene associato alla prossima forma chiusa da disegnare
- Dato che un colore di riempimento è stato selezionato, quando l'utente disegna una forma chiusa, allora l'interno della forma viene riempito con il colore scelto

**Priority:** Alta

**Story Points:** 3

#### *1.1.1.6 Salvataggio dei disegni*

**Story:**

Come utente,  
Vorrei poter salvare il disegno in un file,  
Così che possa conservarlo sul mio dispositivo.

**Acceptance Criteria:**

- Dato che l'interfaccia è visibile, quando l'utente guarda l'interfaccia, allora vede un bottone "Salva"
- Dato che il bottone "Salva" è visibile, quando l'utente lo clicca, allora si apre una finestra di dialogo per scegliere dove salvare il file
- Dato che l'utente ha scelto un percorso di salvataggio, quando conferma l'azione, allora il disegno viene salvato correttamente in un formato immagine

**Priority:** Alta

**Story Points:** 5

#### *1.1.1.7 Caricamento dei disegni*

**Story:**

Come utente,  
Vorrei poter caricare un disegno precedentemente salvato,  
Così che possa continuare a lavorarci.

**Acceptance Criteria:**

- Dato che l'interfaccia è visibile, quando l'utente guarda l'interfaccia, allora è presente un bottone "Carica"
- Dato che il bottone "Carica" è visibile, quando l'utente lo clicca, allora si apre una finestra di selezione file per scegliere un disegno nel formato immagine supportato
- Dato che l'utente ha selezionato un file, quando conferma la scelta, allora il disegno viene caricato correttamente nell'area di disegno dell'applicazione

**Priority:** Alta

**Story Points:** 5

## 1.1.2 Shape editing

### 1.1.2.1 Selezione forma

#### **Story:**

Come utente,  
Voglio poter selezionare una forma già disegnata e applicarvi un'operazione tramite mouse,  
in modo da poterne variare le proprietà.

#### **Acceptance criteria:**

- Data una forma presente nel riquadro di disegno,  
Quando clicco la forma con il mouse,  
Allora questa deve risultare selezionata e pronta per essere manipolata.

**Priority:** Medio - Alta

**Story points:** 3

### 1.1.2.2 Eliminazione di una forma

#### **Story:**

Come utente,  
Voglio poter eliminare la forma selezionata,  
in modo da poterla cancellare liberamente dal riquadro di disegno.

#### **Acceptance criteria:**

- Data una forma selezionata,  
Quando clicco l'opzione "elimina",  
Allora la forma deve essere rimossa dal riquadro di disegno.

**Priority:** Medio - Alta

**Story points:** 1

### 1.1.2.3 Copia e Incolla

#### **Story:**

Come utente,  
Voglio poter copiare e incollare le forme selezionate,  
in modo da poterle salvare negli appunti e riutilizzare nel riquadro di disegno.

#### **Acceptance criteria:**

- Date delle forme selezionate,  
Quando clicco sull'opzione "copia",  
Allora la selezione deve essere salvata negli appunti.
- Date delle forme salvate negli appunti,  
Quando clicco l'opzione "incolla",  
Allora le forme devono comparire nel riquadro nella posizione del cursore al momento del click.

**Priority:** Medio - Alta

**Story points:** 3

#### 1.1.2.4 Taglia

##### **Story:**

Come utente,  
voglio poter tagliare le forme selezionate,  
in modo da rimuoverle dal riquadro di disegno e copiarle negli appunti.

##### **Acceptance criteria:**

- Date una o più forme selezionate,  
Quando clicco sull'opzione "taglia",  
Allora le forme devono essere rimosse dal riquadro di disegno e copiate negli appunti.

**Priority:** Medio - Alta

**Story points:** 2

#### 1.1.2.5 Undo

##### **Story:**

Come utente,  
voglio poter annullare illimitatamente le operazioni precedentemente effettuate,  
in modo da poter tornare a versioni precedenti del riquadro di disegno.

##### **Acceptance criteria:**

- Date delle operazioni effettuate,  
Quando clicco il tasto apposito per effettuare l'annullamento (undo),  
Allora deve essere annullata l'ultima azione performata nel riquadro di disegno.

**Priority:** Medio - Alta

**Story points:** 5

#### 1.1.2.6 Modifica della posizione di una forma

##### **Story:**

Come utente,  
voglio poter modificare la posizione della forma selezionata,  
in modo da poterla spostare liberamente all'interno del riquadro di disegno.

##### **Acceptance criteria:**

- Data una forma selezionata,  
Quando la trascino con il mouse,  
Allora la forma deve seguire il movimento del cursore e aggiornare la sua posizione nel riquadro in tempo reale.

**Priority:** Medio - Alta

**Story points:** 2

#### 1.1.2.7 Modifica del colore di una forma

##### **Story:**



Come utente,  
voglio poter modificare il colore, sia del contorno che dell'interno, della forma selezionata,  
in modo da personalizzarne liberamente l'aspetto.

**Acceptance criteria:**

- Data una forma selezionata,  
Quando modifico il colore tramite la sezione apposita,  
Allora la forma deve aggiornare il suo aspetto in tempo reale, riflettendo i nuovi colori scelti.

**Priority:** Medio - Alta

**Story points:** 2

*1.1.2.8 Modifica delle dimensioni di una forma*

**Story:**

Come utente,  
voglio poter modificare le dimensioni della forma selezionata,  
in modo da ingrandirla o ridurla liberamente all'interno del riquadro di disegno.

**Acceptance criteria:**

- Data una forma selezionata,  
Quando modifico le dimensioni tramite menu contestuale,  
Allora la forma deve cambiare la propria dimensione.

**Priority:** Medio - Alta

**Story points:** 5

*1.1.2.9 Modifica del livello di una forma*

**Story:**

Come utente,  
voglio poter modificare il livello di una forma o di un gruppo di forme,  
in modo da controllare la sovrapposizione e portarle in piani differenti.

**Acceptance criteria:**

- Dato una o più forme presenti nel riquadro di disegno,  
Quando seleziono "Porta in primo piano",  
Allora la forma, o il gruppo di forme, deve/devono comparire sopra tutte le altre.
- Dato una o più forme presenti nel riquadro di disegno,  
Quando seleziono il comando apposito (ad es. "Porta avanti di un livello"),  
Allora la forma, o il gruppo di forme, deve/devono spostarsi di un livello sopra rispetto all'attuale posizione.
- Dato una o più forme presenti nel riquadro di disegno,  
Quando seleziono il comando apposito (ad es. "Porta indietro di un livello"),  
Allora la forma, o il gruppo di forme, deve/devono spostarsi di un livello sotto rispetto all'attuale posizione.

- Dato una o più forme presenti nel riquadro di disegno,  
Quando seleziono "Manda in ultimo piano",  
Allora la forma, o il gruppo di forme, deve/devono comparire sotto tutte le altre.

**Priority:** Medio - Alta

**Story points:** 3

### 1.1.3 Display of the drawing

#### 1.1.3.1 *Modifica del livello di zoom*

**Story:**

Come utente,  
Voglio poter modificare il livello di zoom dell'area di disegno,  
Così da poter osservare l'intero disegno o i suoi dettagli.

**Acceptance criteria:**

- Dato un disegno visualizzato nella canvas,  
Quando l'utente seleziona un nuovo livello di zoom tramite bottoni, slider o menu,  
Allora il livello di zoom viene aggiornato correttamente tra almeno quattro valori predefiniti (es. 25%, 50%, 100%, 200%).
- Dato un qualsiasi livello di zoom attivo,  
Quando l'utente modifica lo zoom,  
Allora l'interfaccia mostra chiaramente il nuovo livello di zoom selezionato.
- Dato un disegno composto da una o più forme,  
Quando l'utente effettua uno zoom in o out,  
Allora le forme mantengono proporzioni corrette e posizione coerente all'interno dell'area di disegno.

**Priority:** Media

**Story points:** 5

#### 1.1.3.2 *Area di disegno espandibile*

**Story:**

Come utente,  
voglio che l'area di disegno possa essere più grande della finestra,  
così da poter creare disegni senza essere limitato dallo spazio visibile.

**Acceptance criteria:**

- Data un'area di disegno attiva,  
Quando l'utente crea un disegno più grande della finestra,  
Allora l'area si espande dinamicamente per contenere tutto il contenuto, senza perdere o tagliare alcuna parte.
- Dato un disegno che si estende oltre i limiti visibili della finestra,  
Quando l'utente inserisce o modifica forme in una zona non visibile,

Allora tali forme vengono correttamente gestite, visualizzate e conservate anche se si trovano al di fuori dell'area attualmente visibile.

**Priority:** Media

**Story points:** 3

#### *1.1.3.3 Scroll area di disegno*

**Story:**

Come utente,  
voglio poter scorrere l'area di disegno,  
così da visualizzare e modificare qualsiasi parte del mio disegno in caso l'area di disegno sia più grande dell'interfaccia.

**Acceptance Criteria:**

- Data un'area di disegno con dimensioni maggiori rispetto alla finestra,  
Quando l'utente visualizza l'area,  
Allora vengono mostrate automaticamente una scrollbar orizzontale e una verticale per permettere lo scorrimento.
- Data un'area di disegno con scrollbar visibili,  
Quando l'utente le utilizza per navigare,  
Allora può visualizzare correttamente ogni parte del disegno, senza perdita di contenuto né distorsioni.

**Priority:** Media

**Story points:** 2

#### *1.1.3.4 Visualizzazione della griglia*

**Story:**

Come utente,  
voglio poter visualizzare una griglia sull'area di disegno,  
così da posizionare più facilmente le forme.

**Acceptance criteria:**

- Data un'area di disegno con forme già inserite,  
Quando l'utente attiva la visualizzazione della griglia,  
Allora vengono mostrate linee orizzontali e verticali equidistanti, posizionate sopra lo sfondo ma sotto le forme.
- Data una griglia visibile,  
Quando l'utente interagisce con le forme (es. selezione o trascinamento),  
Allora le linee della griglia non interferiscono con tali operazioni.
- Data la visualizzazione della griglia attivata,  
Quando l'area di disegno viene aggiornata (es. tramite scroll o zoom),  
Allora la griglia si muove coerentemente con la canvas e adatta la distanza tra le linee proporzionalmente al livello di zoom.

**Priority:** Media

**Story points:** 3

#### *1.1.3.5 Attivazione e disattivazione della griglia*

**Story:**

Come utente,  
voglio poter attivare o disattivare la griglia,  
così da poterla usare solo se necessario.

**Acceptance criteria:**

- Data una sezione dell'interfaccia con un controllo per la griglia (es. checkbox o voce di menu),  
Quando l'utente attiva o disattiva l'opzione,  
Allora la griglia viene mostrata o nascosta in tempo reale nell'area di disegno.
- Dato lo stato corrente della griglia (attiva o disattiva),  
Quando l'utente utilizza altre funzionalità (es. inserimento di forme, zoom),  
Allora lo stato della griglia rimane invariato e il suo stato è chiaramente visibile nell'interfaccia.

**Priority:** Media

**Story points:** 1

#### *1.1.3.6 Cambiare dimensioni griglia*

**Story:**

Come utente,  
voglio poter scegliere la dimensione della griglia,  
così da poterla adattare meglio al tipo di disegno che sto facendo.

**Acceptance Criteria:**

- Data una griglia visibile sull'area di disegno,  
Quando l'utente seleziona una nuova dimensione tra i valori predefiniti (es. 10, 20, 40 unità logiche),  
Allora la griglia si aggiorna in tempo reale, mostrando linee equidistanti e proporzionate anche in presenza di zoom.
- Data una griglia disattivata,  
Quando l'utente modifica la dimensione della griglia,  
Allora non viene mostrata alcuna modifica visiva, ma il nuovo valore viene memorizzato e applicato alla successiva attivazione.

**Priority:** Media

**Story points:** 3

## 1.1.4 More shapes and editing operations

### 1.1.4.1 Poligoni irregolari

#### **Story:**

Come utente,  
voglio poter creare poligoni irregolari come nuove forme,  
in modo da poter ideare figure personalizzate nel riquadro di disegno.

#### **Acceptance criteria:**

- Dato un poligono irregolare presente nel riquadro di disegno,  
Quando lo seleziono per cliccare sul comando apposito (ad es. "rendi una forma - poligono irregolare"),  
Allora questo poligono irregolare deve diventare una forma riutilizzabile.
- Data una nuova forma "poligono irregolare" inserita,  
Quando la seleziono come una qualsiasi altra forma,  
Allora devo poterlo manipolare come le altre forme.

**Priority:** Medio - Bassa

**Story points:** 8

### 1.1.4.2 Stringhe

#### **Story:**

Come utente,  
voglio poter creare stringhe di testo riconosciute come forme,  
in modo da manipolare le stringhe come le forme solitamente utilizzabili.

#### **Acceptance criteria:**

- Data una stringa presente nel riquadro di disegno,  
Quando clicco sul comando apposito (ad es. "rendi una forma - stringa"),  
Allora questa stringa deve diventare una forma riutilizzabile.
- Data una nuova forma "stringa" inserita,  
Quando la seleziono,  
Allora devo poterla manipolare come le altre forme.
- Data una forma di tipo "stringa" selezionata,  
Quando utilizzo i controlli di modifica del testo,  
Allora devo poter modificare la dimensione dei caratteri della stringa stessa.

**Priority:** Medio - Bassa

**Story points:** 5

### 1.1.4.3 Rotazione di una forma

#### **Story:**

Come utente,  
voglio poter ruotare arbitrariamente una forma (o gruppo di forme),  
in modo da poterne regolare liberamente l'angolazione, nel riquadro di disegno.

**Acceptance criteria:**

- Data una forma, presente nel riquadro di disegno,  
Quando la seleziono,  
Allora devo essere disponibile un comando, o un punto di manipolazione, apposito per ruotarla (ad es. "ruota").
- Data una forma per cui è stato selezionato il comando apposito per ruotarla, oppure per cui è sfruttato il punto di manipolazione dedito alla rotazione,  
Quando muovo il mouse lungo la circonferenza (non visibile) che circonda la forma,  
Allora la forma deve ruotare, seguendo in tempo reale i movimenti del cursore.

**Priority:** Medio - Bassa

**Story points:** 5

#### *1.1.4.4 Specchiatura*

**Story:**

Come utente,  
voglio poter specchiare, orizzontalmente e verticalmente, una forma (o un gruppo di forme) selezionata,  
in modo da poterne variare l'orientamento visivo.

**Acceptance criteria:**

- Data una forma presente nel riquadro di disegno,  
Quando la seleziono,  
Allora deve essere disponibile un comando apposito per specchiarla (ad es. "specchia la selezione").
- Data una forma selezionata,  
Quando attivo il comando dedito alla specchiatura,  
Allora la forma deve essere specchiata rispetto all'asse orizzontale, o verticale, in base all'opzione selezionata.

**Priority:** Medio - Bassa

**Story points:** 5

#### *1.1.4.5 Stretch di una forma*

**Story:**

Come utente,  
voglio poter allungare o comprimere una forma (o un gruppo di forme) selezionata,  
verticalmente e orizzontalmente,  
in modo da modificarne le proporzioni.

**Acceptance criteria:**

- Data una forma presente nel riquadro di disegno,  
Quando la seleziono,  
Allora deve essere possibile attivare un comando, o sfruttare un punto di manipolazione, per deformarla.

- Data una forma selezionata,  
Quando trascino uno degli appositi punti di controllo lungo l'asse orizzontale o verticale,  
Allora la forma deve modificare la propria dimensione solo lungo quell'asse.

**Priority:** Medio - Bassa

**Story points:** 5

## 1.1.5 Groups and shape libraries

### 1.1.5.1 Selezione Multipla

**Story:**

Come utente,  
Voglio selezionare più forme contemporaneamente,  
Così da poter applicare le stesse operazioni a tutte le forme selezionate.

**Acceptance Criteria:**

- Dato che ho selezionato due o più forme,  
Quando applico una modifica ad una di esse,  
Allora le modifiche vengono applicate a tutte le forme selezionate.

**Priority:** Bassa

**Story Points:** 3

### 1.1.5.2 Grouping

**Story:**

Come utente,  
Voglio raggruppare le forme selezionate,  
Così da poterle trattare come un'unica forma durante le interazioni future.

**Acceptance Criteria:**

- Dato che ho selezionato due o più forme tramite selezione multipla,  
Quando clicco il pulsante "Raggruppa",  
Allora posso interagire con il gruppo di oggetti come fosse un unico.

**Priority:** Bassa

**Story Points:** 5

### 1.1.5.3 Ungrouping

**Story:**

Come utente,  
Voglio disaggregare le forme precedentemente raggruppate,  
Così da poter trattare nuovamente ogni forma in modo indipendente.

**Acceptance Criteria:**

- Dato che ho selezionato una forma ottenuta raggruppando altre forme,  
Quando clicco sul pulsante "Separa",  
Allora ottengo di nuovo le singole forme e posso interagirvi singolarmente.

**Priority:** Bassa

**Story Points:** 3

#### *1.1.5.4 Shape creation commands*

**Story:**

Come utente,  
Voglio definire i "Shape creation commands" salvando una forma selezionata con un nome,  
Così che possa riprodurre la stessa forma in seguito, indipendentemente dalle modifiche apportate all'originale.

**Acceptance Criteria:**

- Dato che ho selezionato una forma, quando clicco il pulsante "Salva Shape Creation Command" e fornisco un nome, posso visualizzarlo tra le forme supportate.
- Dato che ho salvato almeno uno "Shape creation command", quando scelgo quello che mi interessa, posso inserirlo nello spazio di lavoro.

**Priority:** Bassa

**Story Points:** 13

#### *1.1.5.5 Memorizzazione dei "Shape creation commands"*

**Story:**

Come utente,  
Voglio che i "Shape creation commands" siano salvati all'interno del drawing file,  
Così che persistano quando chiudo e riapro il programma o condivido il disegno.

**Acceptance Criteria:**

- Dato che ho salvato almeno uno "Shape creation commands", quando clicco sul pulsante per salvare il progetto, nel drawing file vengono salvate anche i "Shape creation commands" creati.
- Dato che ho salvato un drawing file (contenente "Shape creation commands"), quando lo carico nel mio spazio di lavoro, i comandi sono disponibili per essere utilizzati.

**Priority:** Bassa

**Story Points:** 8

#### *1.1.5.6 Esportazione di uno "Shapes library file"*

**Story:**

Come utente,  
Voglio esportare i miei "Shape creation commands" in un "Shapes library file",  
Così che possa eseguirne il backup, condividerli o utilizzarli in altri disegni.



**Acceptance Criteria:**

- Dato che ho salvato almeno uno "Shape creation commands", quando clicco sul pulsante "Esporta Shapes Library" e scelgo un path, i comandi salvati vengono salvati in uno "Shapes library file".

**Priority:** Bassa

**Story Points:** 8

### 1.1.5.7 Importazione di uno "Shapes library file"

**Story:**

Come utente,

Voglio importare i "Shape creation commands" da un "Shapes library file" nel "drawing file" corrente,

Così che possa riutilizzare forme predefinite o librerie create da altri.

**Acceptance Criteria:**

- Dato che ho uno "Shape library file", quando clicco il pulsante "Import Shape Library", posso utilizzare tutte le forme presenti nel file.

**Priority:** Bassa

**Story Points:** 8

## 1.2 Definition of Done

1. Il codice implementa tutti i requisiti, soddisfacendo il comportamento atteso.
2. Il codice è stato testato, utilizzando casi base e avanzati, producendo output corretti.
3. Il codice è stato sottoposto a revisione incrociata da parte dei membri del team, per garantire la conformità agli standard.
4. Il modulo è stato integrato e testato insieme agli altri componenti del sistema, producendo gli output attesi senza conflitti o errori.
5. Il codice è stato integrato nel branch principale del repository, senza conflitti o errori.
6. Il codice è ben commentato, ove necessario, spiegando in modo conciso la logica e le funzionalità implementate.
7. È stata redatta una documentazione tecnica che descrive le funzionalità e include esempi utili ed espliciti sui casi d'uso delle funzionalità stesse.
8. Le funzionalità implementate rispettano i requisiti riguardanti prestazioni minime e i criteri di efficienza attesi.

## 1.3 Description of software architecture

In questo paragrafo è descritta l'architettura adottata per lo sviluppo del software. Questa rappresenta la struttura fondamentale, il telaio, del sistema. Inoltre, definire un'architettura permette di stabilire i componenti principali, nonché le loro responsabilità.

### 1.3.1 Architettura

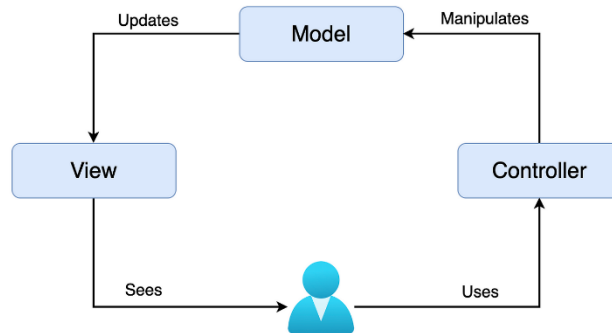


Figura 1.3.1 - Dimostrazione grafica del pattern architetturale MVC

L'applicazione verrà costruita seguendo il paradigma architetturale **Model-View-Controller** (MVC), una scelta comune per lo sviluppo di applicazioni interattive, dotate di interfaccia grafica, come nel caso del software corrente.

Questa architettura permette una netta separazione della logica di dominio, contenuta nel Model, della visualizzazione grafica, contenuta nella View, e della gestione dell'interazione utente, contenuta nel Controller.

Formalmente:

1. **Model:** è responsabile della logica applicativa, gestisce i dati e garantisce la coerenza e l'integrità dei dati (non è a conoscenza né della View né del Controller).
2. **View:** è responsabile della rappresentazione grafica delle informazioni contenute nel Model e si occupa di fornire all'utente un'interfaccia intuitiva e reattiva, che si aggiorni in risposta alle modifiche del Model.
3. **Controller:** media tra Model e View, intercettando gli input dell'utente e interpretandoli, va ad invocare le appropriate operazioni sul Model. Conseguentemente, aggiorna la View per riflettere le modifiche avvenute.

Tra i vantaggi principali di questo tipo di architettura abbiamo:

- **Manutenibilità e testabilità:** vi è una netta separazione tra logica applicativa e interfaccia, consentendo di testare e mantenere le singole componenti indipendentemente le une dalle altre.
- **Estendibilità e modularità:** l'architettura facilita l'aggiunta di nuove funzionalità, permettendo l'applicazione di modifiche localizzate alle componenti interessate senza impattare l'intero sistema, riducendo il rischio di introdurre regressioni nel sistema complessivo.
- **Collaborazione semplificata:** permette una suddivisione del lavoro più efficiente tra i membri del gruppo di lavoro, in modo chiaro. Ogni membro del team può concentrarsi su una componente specifica, in modo da evitare interferenze con il lavoro altrui.

## 1.4 Architecture and technologies chosen

Data la scelta di MVC come architettura abbiamo effettuato le seguenti scelte a livello di tecnologie:

- Linguaggio: Java
- GUI toolkit: JavaFX
- Testing: JUnit
- Versioning: Git + GitHub
- Gestione attività: Trello
- Development environment: NetBeans

## 1.5 Mock-up interface

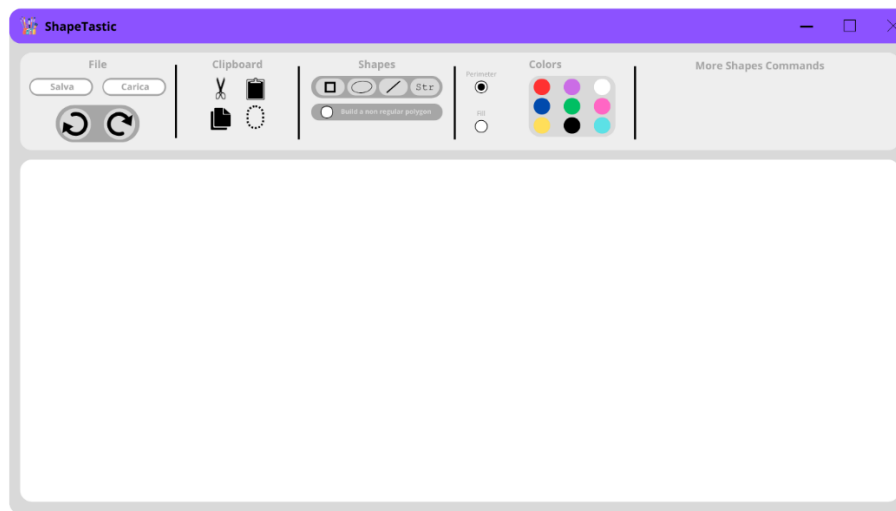


Figura 1.5.1 - Interfaccia prevista per il sistema

## 2 First Sprint

### 2.1 First sprint Planning

#### 2.1.1 Stima iniziale della Project Velocity

In questa fase è importante definire quanti Story Points pensiamo di poter completare in una sprint. Tenendo conto che ci si aspetta che ogni membro del team lavori 8-9 ore per sprint e considerando 1 ora di lavoro per Story Point una prima stima potrebbe essere la seguente:

Velocity = 4 membri \* 8-9 ore = 32-36 Story Points

Quindi la stima iniziale è di circa 32-36 Story Points per sprint.

#### 2.1.2 First Sprint Backlog

User Story	Story Points	
Finestra iniziale	1	
Aggiunta forme geometriche	3	
Forme supportate	3	
Scelta colore del bordo delle forme	3	
Scelta colore di riempimento delle forme	3	
Salvataggio dei disegni	5	
Caricamento dei disegni	5	
Selezione forma	3	
Eliminazione di una forma	1	
Copia e incolla	3	
Taglia	2	
Totale:		32

Tabella 1: First Sprint Backlog definita nella fase Pre-Game

Abbiamo stimato che durante la prima sprint verranno implementate tutte le User Stories ad alta priorità, contenenti le basic operations, e solo una parte di quelle con priorità medio-alta, contenenti le prime operazioni di shape editing. In totale, gli story points previsti per la prima sprint sono 32.

Queste stime si tengono in linea con la stima iniziale riguardante la Project Velocity.

## 2.1.3 Suddivisione in task e assegnazione

### 2.1.3.1 Finestra iniziale (1 SP)

Obiettivo: aprire una finestra vuota all'avvio dell'app.

Tasks	Responsabile
Impostare la finestra principale	Gennaro D'Ambrosio
Inserire di un riquadro di disegno all'interno della finestra principale	Gennaro D'Ambrosio
Scrivere e/o eseguire test per verificare la corretta visualizzazione della finestra e del riquadro di disegno	Ciro Cutolo

Tabella 2: Tasks US1

### 2.1.3.2 Aggiunta forme geometriche (3 SP)

Obiettivo: selezionare una forma e disegnarla con il mouse.

Tasks	Responsabile
Gestire l'occorrenza del click, sul riquadro di disegno, dopo la selezione di una forma, utile per inserire la forma selezionata	Alessio Bottiglieri
Aggiungere delle componenti logiche per creare e visualizzare una linea, una per visualizzare un rettangolo e una per visualizzare un'ellisse, nel riquadro di disegno	Ciro Cutolo
Scrivere e/o eseguire test per verificare l'effettiva presenza di una componente utile alla selezione della forma, nonché il corretto inserimento della forma nel riquadro	Gaetano Frasca

Tabella 3: Tasks US2

*Nota: Questa storia dev'essere implementata dopo la 1.1.1.3 "Forme supportate".*

### 2.1.3.3 Forme supportate (3 SP)

Obiettivo: aggiungere le forme supportate dall'applicazione.

Tasks	Responsabile
Progettare la struttura dati per rappresentare le diverse forme	Ciro Cutolo
Implementare un componente nell'interfaccia che permetta di scegliere la forma da inserire	Gaetano Frasca
Implementare la logica che colleghi il selettore di forme con la selezione effettiva della forma scelta.	Gaetano Frasca

Scrivere e/o eseguire test per verificare l'effettiva presenza di una componente utile alla visualizzazione delle tre forme supportate all'interno dell'interfaccia.	Alessio Bottiglieri
--	---------------------

Tabella 4: Tasks US3

#### 2.1.3.4 Scelta colore bordo forma (3 SP)

Obiettivo: scegliere colore bordo prima di disegnare la forma.

Tasks	Responsabile
Aggiungere un selettore di colore per il bordo delle forme come elemento UI	Alessio Bottiglieri
Implementare un meccanismo che colleghi la scelta del colore alla forma selezionata e lo applichi al bordo della stessa al momento dell'inserimento nel riquadro di disegno	Alessio Bottiglieri
Scrivere e/o eseguire test per verificare la corretta visualizzazione di una componente UI per la selezione del colore bordo nonché il corretto funzionamento della logica implementata per modificare il colore del bordo della forma che stiamo per inserire	Gennaro D'Ambrosio

Tabella 5: Tasks US4

#### 2.1.3.5 Scelta colore riempimento (3 SP)

Obiettivo: selezionare colore di riempimento per rettangolo o ellisse.

Tasks	Responsabile
Aggiungere un selettore di colore per la sezione interna delle forme come elemento UI	Ciro Cutolo
Implementare un meccanismo che colleghi la scelta del colore alla forma selezionata e lo applichi all'interno della stessa al momento dell'inserimento nel riquadro di disegno	Ciro Cutolo
Scrivere e/o eseguire test per verificare la corretta visualizzazione di una componente UI per la selezione del colore di riempimento nonché il corretto funzionamento della logica implementata per modificare il colore di riempimento della forma che stiamo per inserire	Gaetano Frasca

Tabella 6: Tasks US5

### 2.1.3.6 Salvataggio del disegno (5 SP)

Obiettivo: Salvare il disegno in un file.

Tasks	Responsabile
Implementare la logica per convertire e salvare i disegni	Gennaro D'Ambrosio
Aggiungere un componente, dedicato al salvataggio, all'interfaccia ed implementare una finestra di dialogo utile all'utente per scegliere dove salvare il file	Gennaro D'Ambrosio
Collegare il componente dedicato al salvataggio all'effettiva funzione di salvataggio	Gennaro D'Ambrosio
Scrivere e/o eseguire test per verificare la corretta visualizzazione del pulsante di salvataggio e della finestra di dialogo per il salvataggio del file nonché il corretto funzionamento della logica di salvataggio	Ciro Cutolo

Tabella 7: Tasks US6

### 2.1.3.7 Caricamento del disegno (5 SP)

Obiettivo: Caricare un disegno precedentemente salvato.

Tasks	Responsabile
Scrivere la logica utile a caricare un disegno da file	Gaetano Frasca
Aggiungere un pulsante "carica" all'interfaccia ed implementare una finestra di dialogo per permettere all'utente di selezionare il file da caricare	Gaetano Frasca
Collegare il pulsante "carica" all'effettiva funzione di caricamento	Gaetano Frasca
Scrivere e/o eseguire test per verificare la corretta visualizzazione del pulsante di caricamento e della finestra di dialogo per il caricamento del file nonché il corretto funzionamento della logica di caricamento	Alessio Bottiglieri

Tabella 8: Tasks US7

### 2.1.3.8 Selezione forma (3 SP)

Obiettivo: selezionare una forma già presente nel disegno con il mouse.

Tasks	Responsabile
Implementare la logica utile a determinare quando il click del mouse ricade su di una forma	Alessio Bottiglieri
Aggiungere la logica utile a evidenziare graficamente la forma selezionata	Alessio Bottiglieri
Implementare la logica che permetta di gestire la forma selezionata, per effettuare azioni rispetto a quest'ultima (Gestione dello stato interno "forma selezionata" per abilitare operazioni successive")	Ciro Cutolo
Scrivere e/o eseguire test per verificare la corretta visualizzazione grafica di una forma selezionata nonché della logica per gestire lo stato interno di quest'ultima	Gennaro D'ambrosio

Tabella 9: Tasks US8

### 2.1.3.9 Eliminazione di una forma (1 SP)

Obiettivo: eliminare una forma selezionata dal disegno

Tasks	Responsabile
Aggiungere una voce al menu, "Elimina", accanto ad una forma selezionata	Ciro Cutolo
Implementare la logica che colleghi la pressione della voce di menu apposita con l'effettiva eliminazione della forma selezionata	Ciro Cutolo
Implementare la logica che permetta di aggiornare correttamente il riquadro di disegno, dopo l'eliminazione	Ciro Cutolo
Scrivere e/o eseguire test per verificare la corretta visualizzazione della voce "Elimina" accanto alla forma selezionata nonché della logica per aggiornare il riquadro dopo l'eliminazione	Alessio Bottiglieri

Tabella 10: Tasks US9



### 2.1.3.10 Copia e incolla (3 SP)

Obiettivo: copiare la forma selezionata ed incollarla in un altro punto

Tasks	Responsabile
Implementare struttura dati per rappresentare gli "appunti" (clipboard)	Alessio Bottiglieri
Implementare graficamente i pulsanti di "Copia" ed "Incolla"	Alessio Bottiglieri
Implementare la logica utile alla gestione delle funzioni "copia" e "incolla"	Alessio Bottiglieri
Creare un collegamento tra la pressione delle voci "copia" e "incolla" e la logica che permetta di azionare effettivamente i comandi	Gennaro D'Ambrosio
Scrivere e/o eseguire test per verificare la corretta visualizzazione dei pulsanti di "copia" e "incolla" accanto alla forma selezionata nonché la logica per implementare questi comandi	Ciro Cutolo

Tabella 11: Tasks US10

### 2.1.3.11 Taglia (2 SP)

Obiettivo: Rimuovere e salvare la forma negli appunti

Tasks	Responsabile
Aggiungere una voce di menu "taglia"	Ciro Cutolo
Definire la logica utile a sfruttare contemporaneamente le funzioni "copia" ed "elimina"	Ciro Cutolo
Implementare la logica che colleghi la pressione della voce di menu con l'effettiva azione del comando "taglia"	Ciro Cutolo
Scrivere e/o eseguire test per verificare la corretta visualizzazione del pulsante "taglia" accanto alla forma selezionata nonché la logica per implementare effettivamente il comando	Gaetano Frasca

Tabella 12: Tasks US11

## 2.2 First Sprint – Applicazione Pattern

### 2.2.1 Simple Factory Pattern

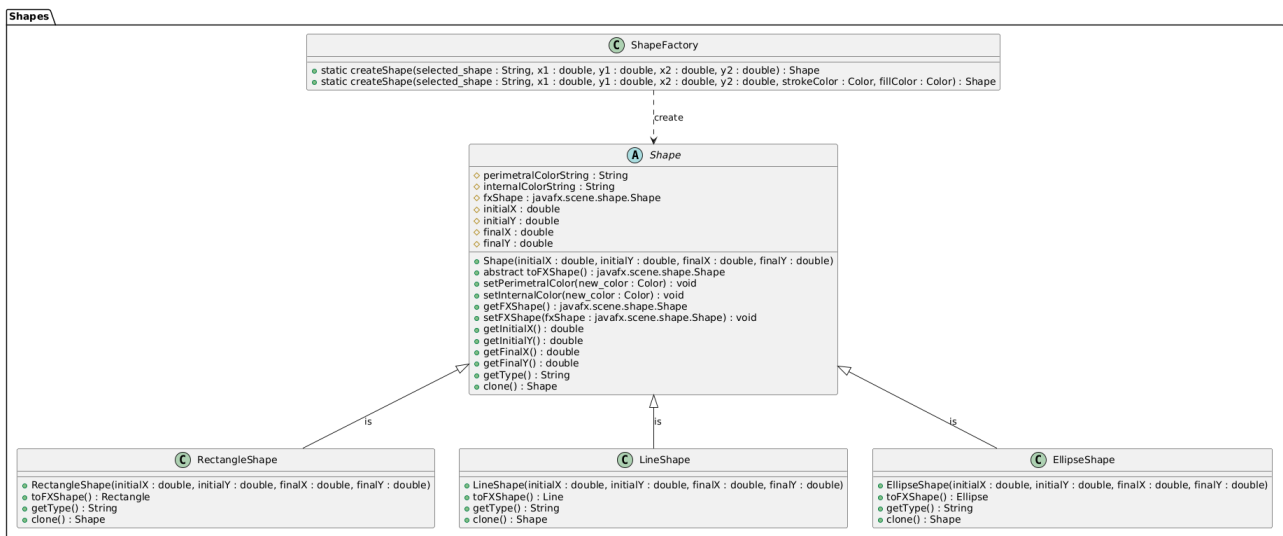


Figura 2: Simple Factory Pattern - Class Diagram

Creational Pattern utilizzato per rendere le classi indipendenti dalla logica di creazione degli oggetti che usano. Nel caso particolare viene applicato alle classi dell'architettura che andranno a rappresentare le forme supportate dal programma: linea, rettangolo ed ellisse.

- L'interfaccia *Shape* riveste il ruolo di *Product*, comune a tutti gli oggetti che possono essere creati dal *Creator*. funge da tipo comune per tutte le forme.
- Le classi concrete *LineShape*, *RectangleShape* ed *EllipseShape*, dette in generale *ConcreteProduct*, implementano l'interfaccia *Shape* per definire il comportamento specifico di ciascuna forma.
- La classe *ShapeFactory* agisce come una fabbrica, un *Creator*, incapsulando la logica di istanziazione all'interno di un unico metodo, il *factory method*, che riceve un identificatore del tipo di forma, una stringa, e restituisce il nuovo oggetto forma corrispondente.

Questo approccio semplifica la creazione degli oggetti, garantisce coerenza e permette all'applicazione di generare nuove forme senza esporre la logica di istanziazione al resto del codice.

Per questo sistema, si è scelto di applicare il Simple Factory al posto del Factory Method, perché più adatto a un'applicazione di piccole dimensioni con un numero limitato di tipi di oggetti e sufficiente per le esigenze del progetto. L'utilizzo del pattern Factory avrebbe introdotto una complessità eccessiva, richiedendo sottoclassi per ogni tipo di forma, senza apportare vantaggi significativi in questo contesto.

## 2.2.2 Prototype Pattern

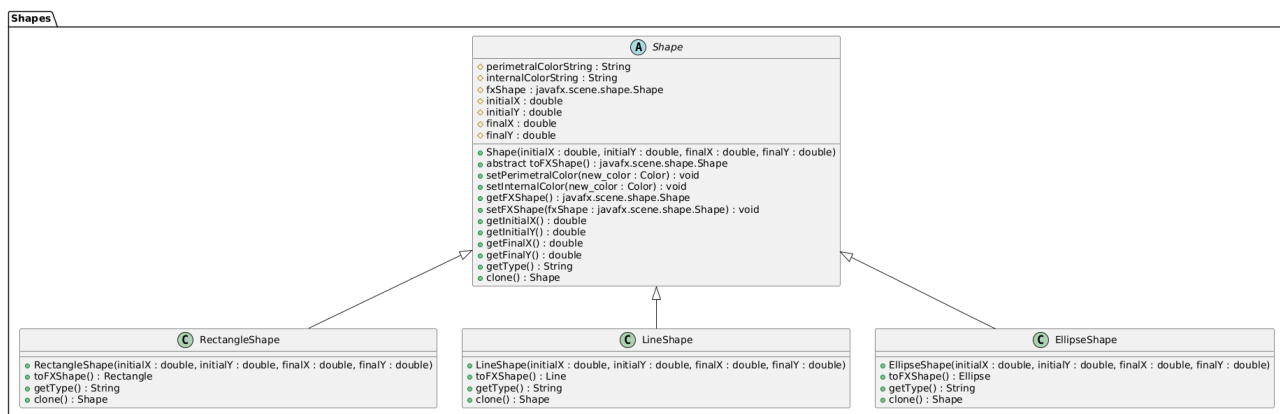


Figura 3: Prototype Pattern - Class Diagram

Il pattern Prototype è stato applicato per consentire la clonazione degli oggetti Shape all'interno dell'applicazione di disegno, in modo efficiente e flessibile. In particolare, questo pattern si rivela utile quando si ha la necessità di creare nuove istanze di oggetti complessi partendo da un oggetto esistente, anziché ricorrere ogni volta alla costruzione manuale.

Nel nostro caso, la classe astratta Shape, che riveste il ruolo di *Prototype*, implementa l'interfaccia Cloneable e fornisce un metodo clone(), rendendolo accessibile alle classi ConcretePrototype. Queste classi concrete, nel nostro caso LineShape, RectangleShape e EllipseShape, ereditano la logica di clonazione di base e di personalizzarla ove necessario. Ad esempio, LineShape ridefinisce clone() per creare una copia logica e grafica dell'oggetto originale, inclusa la duplicazione dell'oggetto JavaFX Line, che rappresenta la parte visuale della forma.

La scelta del Prototype consente di evitare la creazione ripetuta di oggetti tramite costruttori, soprattutto quando molte proprietà devono essere replicate. Inoltre, la logica di clonazione è incapsulata in ciascuna classe rappresentate una forma, rendendo il sistema facilmente estendibile: ogni nuova forma potrà semplicemente implementare la propria versione di clone().

Nel complesso, l'uso del pattern Prototype ha migliorato la modularità e la riusabilità del codice, permettendo la duplicazione rapida delle forme nel canvas senza introdurre dipendenze rigide o codice ridondante.

## 2.3 First Sprint – Testing

Come previsto dalla Definition of Done (DoD), tutte le funzionalità sviluppate durante la sprint sono state sottoposte a test per garantirne la correttezza e la qualità.

In particolare, sono stati effettuati:

- Unit Test sulle classi Java, al fine di verificare il comportamento corretto delle singole unità di codice in isolamento. Tali test sono presenti all'interno della repository di progetto, insieme al relativo codice sorgente.
- Functional Test sull'interfaccia utente, con l'obiettivo di validare il corretto funzionamento delle funzionalità dal punto di vista dell'utente finale. Alcuni di questi test sono documentati nelle sezioni successive.

L'esecuzione di questi test ha permesso di identificare e correggere tempestivamente eventuali anomalie, contribuendo al rispetto dei criteri di accettazione e al rilascio di incrementi di prodotto conformi agli standard qualitativi definiti.

### 2.3.1 US1: Finestra iniziale

#### 2.3.1.1 Test ID : 1.1

**Obiettivo del test:** Verificare che il programma si avvii correttamente e visualizzi una finestra inizialmente vuota.

**Precondizioni:** Nessuna.

**Passi:**

1. Avviare l'applicazione

**Risultato atteso:** L'applicazione si avvia e viene visualizzata una finestra con due sezioni vuote, una toolbar e un foglio di disegno visibile.

**Risultato ottenuto:** L'applicazione mostra correttamente due sezioni vuote, la toolbar e il foglio di disegno all'avvio.

**Risultato:** Successo.



Figura 4: Screen - Finestra iniziale

## 2.3.2 US2: Forme supportate

### 2.3.2.1 Test ID : 2.1

**Obiettivo del test:** Verificare la presenza visiva dei 3 pulsanti nella GUI che indicano le 3 forme supportate.

**Precondizioni:** L'applicazione è avviata.

**Passi:**

1. Avviare l'applicazione
2. Verificare la presenza della sezione di selezione delle forme

**Risultato atteso:** Sono presenti tre pulsanti, uno per la selezione del rettangolo, uno per la selezione dell'ellisse ed uno per la selezione della linea.

**Risultato ottenuto:** Nella sezione relativa alla selezione delle forme sono presenti i pulsanti per rettangolo, ellisse e linea.

**Risultato:** Successo.

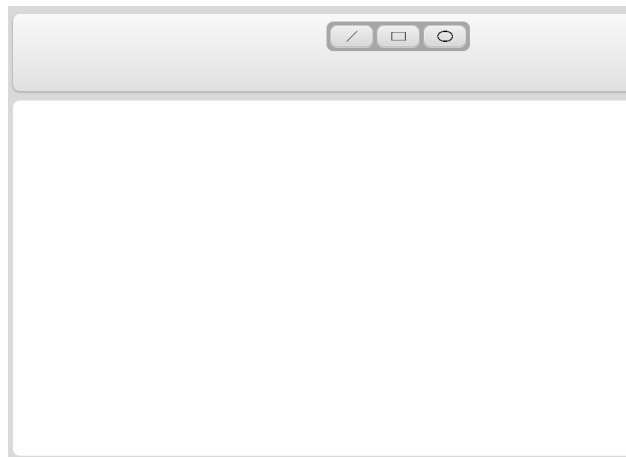


Figura 5: Screen - Forme supportate

### 2.3.2.2 Test ID : 2.2

**Obiettivo del test:** Verificare che la pressione del pulsante relativo alla linea comporti effettivamente la scelta della forma selezionata.

**Precondizioni:**

- L'applicazione è avviata
- Sono presenti i 3 pulsanti per la selezione della forma

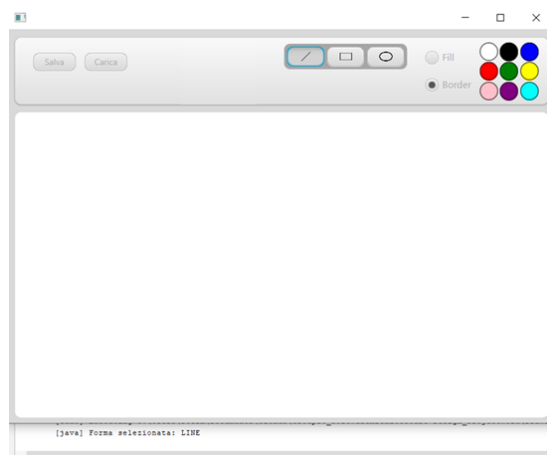
**Passi:**

1. Cliccare sul pulsante di selezione delle forme relativo alla linea

**Risultato atteso:** Dopo la pressione del pulsante viene stampata nel terminale una riga con scritto "Forma selezionata: LINE", mentre graficamente è possibile osservare che il pulsante selezionato appare annerito rispetto agli altri.

**Risultato ottenuto:** In seguito alla pressione del pulsante è stata stampata la riga "Forma selezionata "LINE" ed il bottone appare "cliccato".

**Risultato:** Successo.



**Nota:** La stampa è temporanea ed usata solo per i test. In particolare è stata stampata la variabile *selectedShape*, così da mostrare il cambiamento logico che avviene dopo la pressione del pulsante.

**Nota:** Per la US2 sono stati effettuati i test per verificare il corretto funzionamento dei tre pulsanti di selezione della forma, ma per brevità è stato mostrato solo quello del pulsante "linea".

### 2.3.3 US3: Aggiunta forme geometriche

#### 2.3.3.1 Test ID : 3.1

**Obiettivo del test:** Verificare il corretto inserimento delle forme selezionate nel riquadro di disegno.

**Precondizioni:**

- L'applicazione è avviata
- È possibile selezionare una delle forme supportate

**Passi:**

1. Selezionare la forma "linea" cliccando sul bottone (/).
2. Cliccare in un punto arbitrario del canvas, tenere premuto e trascinare il cursore per definire la forma.
3. Rilasciare quando la forma è della dimensione desiderata.

**Risultato atteso:** Cliccando, trascinando e rilasciando nel canvas, è possibile inserire correttamente una forma. Durante l'azione compare una guida temporanea che mostra un'anteprima della forma. Al rilascio del tasto, la guida viene sostituita dalla forma definitiva..

**Risultato ottenuto:** L'inserimento della forma avviene correttamente, con anteprima dinamica (guida temporanea) e conferma visiva al rilascio del mouse.

**Risultato:** Successo



Figura 6: Screen - Aggiunta forme geometriche

## 2.3.4 US4: Scelta colore del bordo delle forme

### 2.3.4.1 Test ID : 4.1

**Obiettivo del test:** Verificare la presenza di una sezione chiamata "Colore bordo" nell'interfaccia con un selettore contenente almeno 8 colori.

**Precondizioni:**

- L'applicazione è avviata

**Passi:**

1. Verificare che nell'interfaccia sia presente la palette di colori ed il pulsante di opzione per scegliere il colore di bordo

**Risultato atteso:** Nella sezione relativa alla selezione del colore bordo è presente un selettore di colori con almeno 8 colori disponibili.

**Risultato ottenuto:** Nella sezione relativa alla selezione del colore bordo è presente un selettore di colori con almeno 8 colori disponibili.

**Risultato:** Successo.

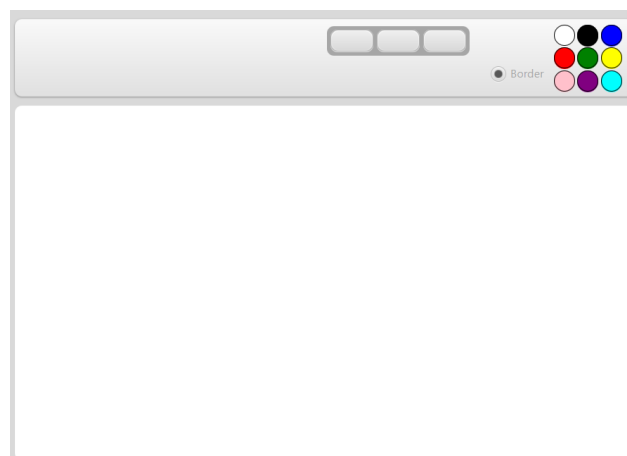


Figura 7: Screen 1 - Scelta colore del bordo delle forme

#### 2.3.4.2 Test ID : 4.2

**Obiettivo del test:** Verificare graficamente la selezione di uno dei colori disponibili nell'interfaccia relativi al colore del bordo di una forma.

**Precondizioni:**

- L'applicazione è avviata

**Passi:**

1. Osservare l'interfaccia e la sezione relativa alla scelta del colore bordo
2. Selezionare uno dei colori disponibili

**Risultato atteso:** Nella sezione relativa alla selezione del colore bordo una volta selezionato uno dei colori si colorerà del medesimo colore il contorno della scritta "Border" come conferma che il colore è stato selezionato con successo.

**Risultato ottenuto:** Nella sezione relativa alla selezione del colore bordo una volta selezionato uno dei colori si è colorato del medesimo colore il contorno della scritta "Border" come conferma che il colore è stato selezionato con successo.

**Risultato:** Successo.

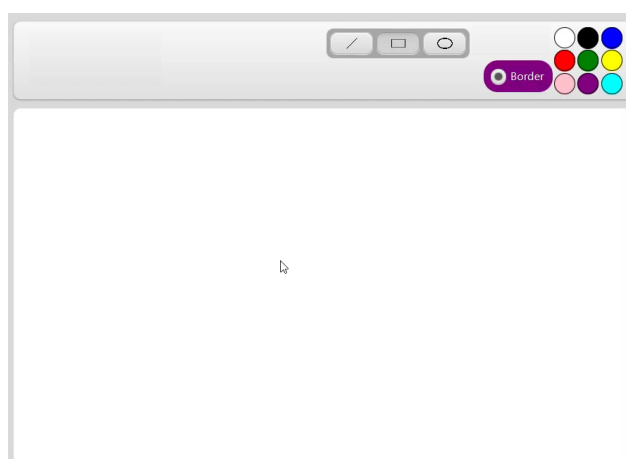


Figura 8: Screen 2 - Scelta colore del bordo delle forme

#### 2.3.4.3 Test ID : 4.3

**Obiettivo del test:** Verificare che una volta che ho selezionato un colore di bordo quando si andrà ad inserire una delle forme disponibili quest'ultima presenti il bordo del colore selezionato.

**Precondizioni:**

- L'applicazione è avviata

**Passi:**

1. Osservare l'interfaccia e la sezione relativa alla scelta del colore bordo
2. Selezionare uno dei colori disponibili
3. Inserire una forma



**Risultato atteso:** Dopo aver selezionato un colore bordo forma ed inserito una delle forme disponibili quest'ultima presenta come colore di bordo lo stesso selezionato dall'utente.

**Risultato ottenuto:** Dopo aver selezionato un colore bordo forma ed inserito una delle forme disponibili quest'ultima presenta come colore di bordo lo stesso selezionato dall'utente. Il test ha avuto successo.

**Risultato:** Successo.

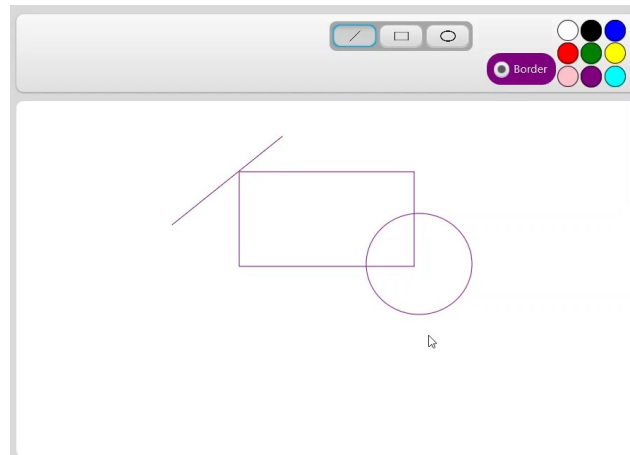


Figura 9: Screen 3 - Scelta colore del bordo delle forme

## 2.3.5 US5: Scelta colore di riempimento delle forme

### 2.3.5.1 Test ID : 5.1

**Obiettivo del test:** Verificare la presenza nell'interfaccia di una sezione per la selezione del colore di riempimento, accessibile tramite un pulsante di opzione ("radio button") che consente di scegliere "Fill" e una palette contenente almeno 8 colori.

**Precondizioni:**

- L'applicazione è avviata

**Passi:**

1. Verificare la presenza della palette dei colori per il riempimento.
2. Verificare che il selettore contenga 8 opzioni di colore.

**Risultato atteso:** È presente una sezione dedicata alla scelta del colore di riempimento ed il selettore visualizza gli 8 colori tra cui scegliere.

**Risultato ottenuto:** Il selettore è visibile e mostra gli 8 colori per la scelta del riempimento.

**Risultato:** Successo.

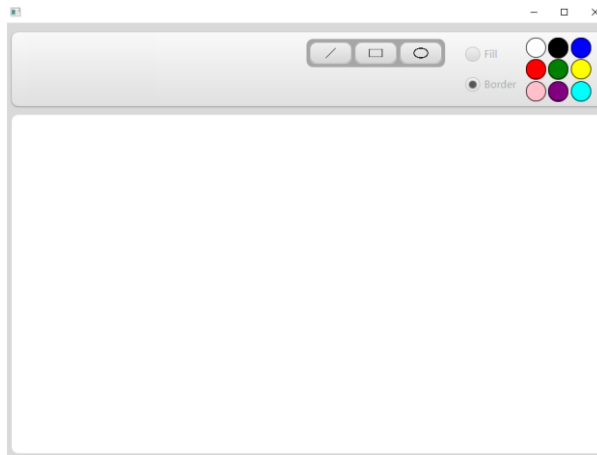


Figura 10: Screen 1 - Scelta colore di riempimento delle forme

#### 2.3.5.2 Test ID : 5.2

**Obiettivo del test:** Verificare graficamente che la selezione di un colore di riempimento mostri un feedback visivo nell'interfaccia.

**Precondizioni:**

- L'applicazione è avviata

**Passi:**

1. Navigare alla sezione di selezione del colore di riempimento.
2. Selezionare uno dei colori disponibili nella palette.

**Risultato atteso:** Una volta selezionato un colore, un elemento visivo (ad esempio, il contorno o sfondo del pulsante "Fill") cambia colore per confermare la selezione.

**Risultato ottenuto:** Il colore selezionato viene evidenziato chiaramente nella UI come feedback visivo.

**Risultato:** Successo

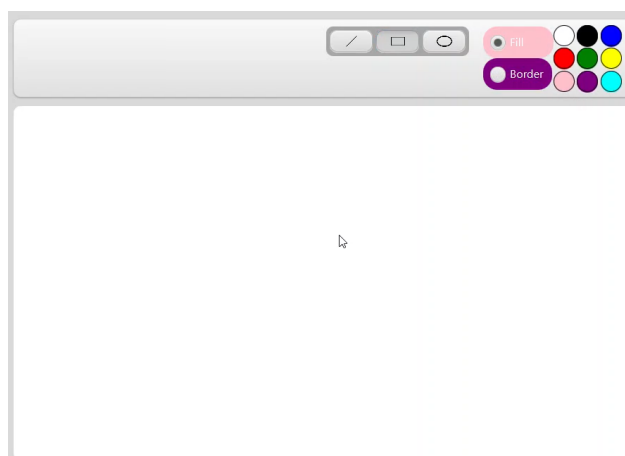


Figura 11: Screen 2 - Scelta colore di riempimento delle forme

### 2.3.5.3 Test ID : 5.3

**Obiettivo del test:** Verificare che il colore di riempimento selezionato venga applicato correttamente a una forma inserita nel canvas.

**Precondizioni:**

- L'applicazione è avviata
- È stato selezionato un colore di riempimento

**Passi:**

1. Inserire una forma nel riquadro di disegno (es. rettangolo o cerchio).

**Risultato atteso:** La forma inserita nel canvas presenta un riempimento del colore selezionato.

**Risultato ottenuto:** Il colore di riempimento visualizzato sulla forma corrisponde a quello scelto dall'utente nella palette.

**Risultato:** Successo.

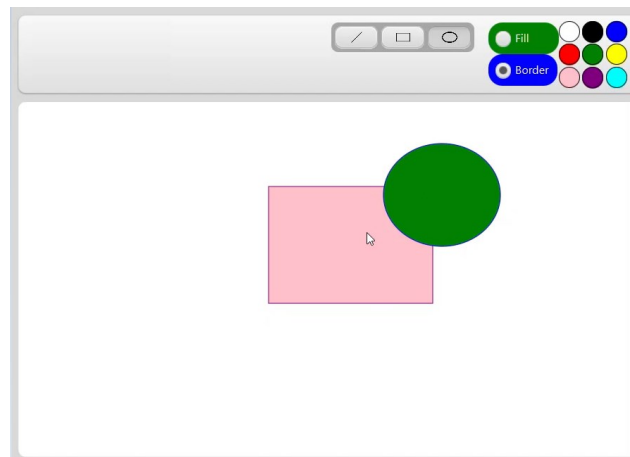


Figura 12: Screen 3 - Scelta colore di riempimento delle forme

### 2.3.6 US6: Salvataggio di un disegno

#### 2.3.6.1 Test ID : 6.1

**Obiettivo del test:** Verificare la presenza di un pulsante "Salva" nella GUI nella sezione relativa al caricamento e salvataggio.

**Precondizioni:**

- L'applicazione è avviata

**Passi:**

1. Osservare l'interfaccia e la sezione relativa al caricamento e salvataggio.

**Risultato atteso:** Nella sezione relativa al caricamento e salvataggio è presente un pulsante "Salva".

**Risultato ottenuto:** Il pulsante "Salva" è visibile nella GUI nella sezione corretta.

**Risultato:** Successo

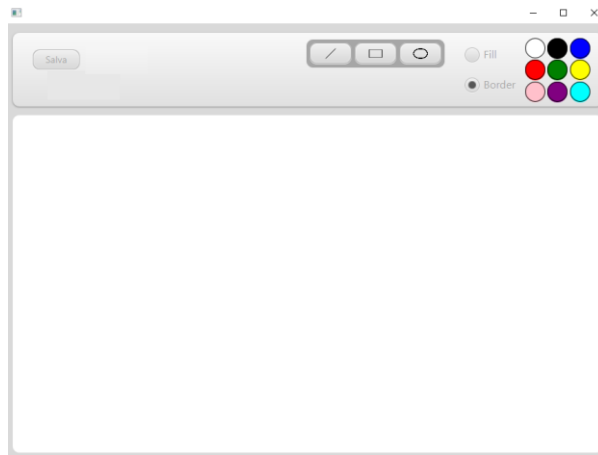


Figura 13: Screen 1 - Salvataggio di un disegno

### 2.3.6.2 Test ID : 6.2

**Obiettivo del test:** Verificare che cliccando sul pulsante "Salva" si apra una finestra di dialogo per la scelta del nome e del percorso del file da salvare.

**Precondizioni:**

- L'applicazione è avviata
- Il pulsante "Salva" è visibile nella GUI

**Passi:**

1. Osservare l'interfaccia e la sezione relativa al caricamento e salvataggio.
2. Cliccare sul pulsante "Salva".

**Risultato atteso:** Cliccando su "Salva", si apre una finestra di dialogo per specificare il nome e la posizione del file da salvare.

**Risultato ottenuto:** La finestra di dialogo per il salvataggio compare correttamente.

**Risultato:** Successo

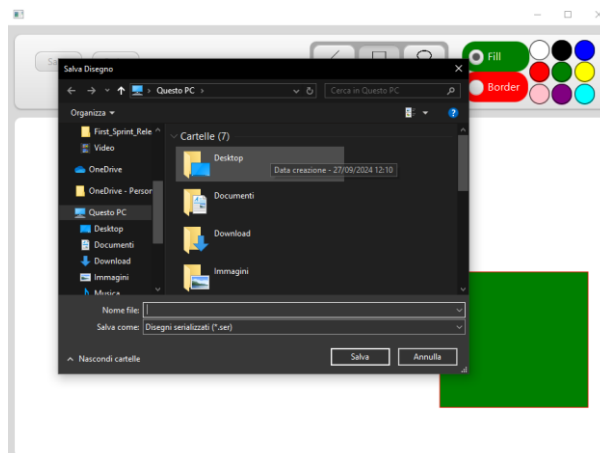


Figura 14 Screen 2 - Salvataggio di un disegno

### 2.3.6.3 Test ID : 6.3

**Obiettivo del test:** Verificare che, dopo aver selezionato nome e posizione, il file venga salvato correttamente nel formato previsto (.ser).

**Precondizioni:**

- L'applicazione è avviata
- Il pulsante "Salva" è visibile nella GUI
- È presente un disegno nell'area di lavoro

**Passi:**

1. Cliccare sul pulsante "Salva".
2. Nella finestra di dialogo, scegliere una posizione, digitare un nome valido e salvare.
3. Verificare che il file venga creato nel percorso indicato.

**Risultato atteso:** Il disegno viene salvato correttamente come file con estensione .ser nella posizione scelta.

**Risultato ottenuto:** Il file .ser è generato nella posizione selezionata e contiene i dati corretti del disegno.

**Risultato:** Successo

### 2.3.7 US7: Caricamento di un disegno

#### 2.3.7.1 Test ID : 7.1

**Obiettivo del test:** Verificare la presenza di un pulsante "carica" nella GUI.

**Precondizioni:**

- L'applicazione è avviata

**Passi:**

1. Osservare l'interfaccia e la sezione relativa al caricamento e salvataggio.

**Risultato atteso:** Nella sezione relativa al caricamento e salvataggio è presente un pulsante "carica"

**Risultato ottenuto:** Nella sezione relativa al caricamento e salvataggio è presente il pulsante "carica". Il test ha avuto successo

**Risultato:** Successo

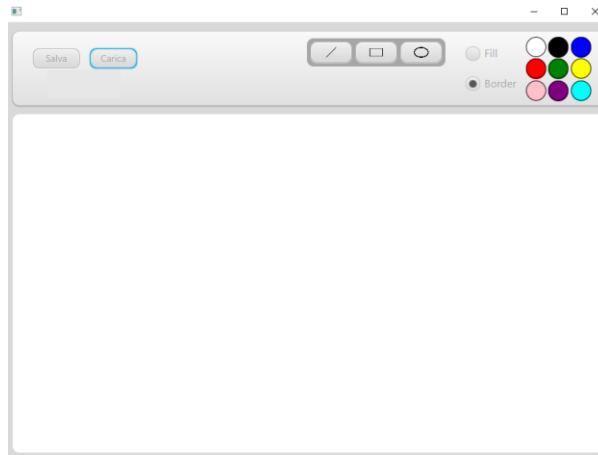


Figura 15 Screen 1 - Salvataggio di un disegno

### 2.3.7.2 Test ID : 7.2

**Obiettivo del test:** Verificare che cliccando sul pulsante "Carica" si apra una finestra di dialogo per la selezione del file.

**Precondizioni:**

- L'applicazione è avviata
- Il pulsante "carica" è visibile nella GUI.

**Passi:**

1. Osservare l'interfaccia e la sezione relativa al caricamento e salvataggio.
2. Cliccare sul pulsante "carica".

**Risultato atteso:** Cliccando su "Carica", si apre correttamente la finestra per selezionare un file.

**Risultato ottenuto:** Dopo aver premuto sul pulsante "carica" viene visualizzata la finestra di selezione del file. Il test ha avuto successo.

**Risultato:** Successo

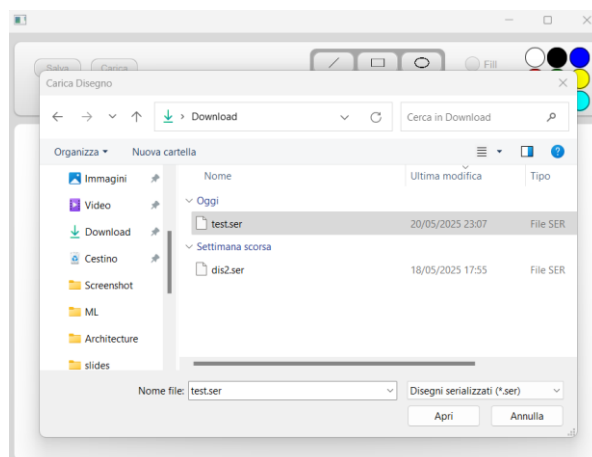


Figura 16: Screen2 - Caricamento di un disegno

### 2.3.7.3 Test ID : 7.3

**Obiettivo del test:** Verificare che, dopo aver selezionato un file immagine e confermato, il disegno venga visualizzato nell'area di disegno.

**Precondizioni:**

- L'applicazione è avviata
- Il pulsante "carica" è visibile nella GUI
- E' disponibile un file .ser precedentemente salvato

**Passi:**

1. Avviare l'applicazione
2. Osservare l'interfaccia e la sezione relativa al caricamento e salvataggio.
3. Cliccare sul pulsante "carica".
4. Selezionare un file immagine valido (.ser)

**Risultato atteso:** Il disegno selezionato viene caricato e visualizzato correttamente nell'area di disegno.

**Risultato ottenuto:** L'area di disegno mostra correttamente il disegno salvato in precedenza.

**Risultato:** Successo.

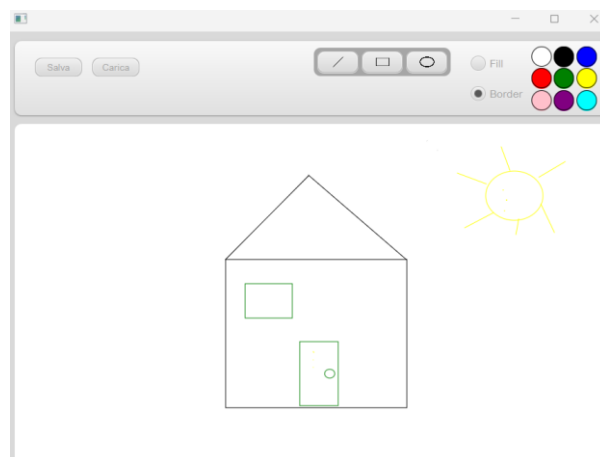


Figura 17: Screen 3 - Caricamento di un disegno

### 2.3.8 US8: Selezione di una Forma

#### 2.3.8.1 Test ID: 8.1

**Obiettivo del test:** Verificare che quando l'utente clicca con il tasto sinistro una forma presente nel pannello di disegno essa venga evidenziata in modo che sappiamo che è pronta per essere manipolata

**Precondizioni:**

- L'applicazione è avviata
- È stata inserita almeno una forma nell'area di disegno

**Passi:**

1. Avviare l'applicazione

2. Inserire almeno una forma nell'area di disegno
3. Fare clic con il tasto sinistro sulla forma inserita

**Risultato atteso:** la forma sulla quale l'utente clicca con il tasto sinistro viene evidenziata graficamente.

**Risultato ottenuto:** la forma sulla quale l'utente ha cliccato con il tasto sinistro è stata evidenziata graficamente.

**Risultato:** Successo

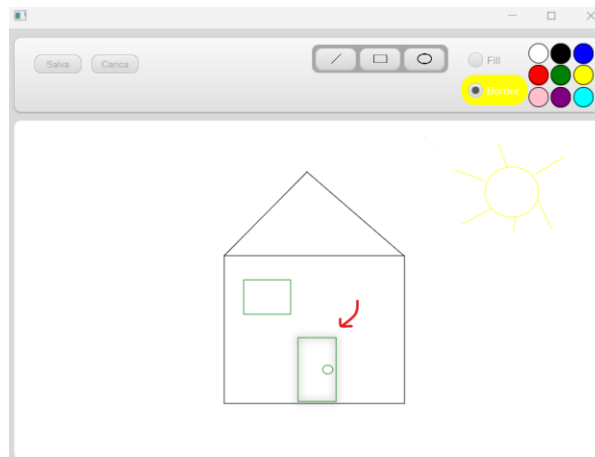


Figura 18: Screen - Selezione di una forma

## 2.3.9 US9 : Eliminazione di una Forma

### 2.3.9.1 Test ID: 9.1

**Obiettivo del test:** Verificare che la voce "Elimina" compaia nel menu contestuale quando si seleziona una forma.

**Precondizioni:**

- L'applicazione è avviata
- È presente almeno una forma disegnata sulla canvas

**Passi:**

1. Avviare l'applicazione
2. Selezionare una forma presente nell'area di disegno
3. Fare clic con il tasto destro all'interno del riquadro dell'applicazione

**Risultato atteso:** Il menu contestuale mostra la voce "Elimina" ogni volta che una forma è selezionata.

**Risultato ottenuto:** Il menu contestuale ha al suo interno la voce "Elimina".

**Risultato:** Successo



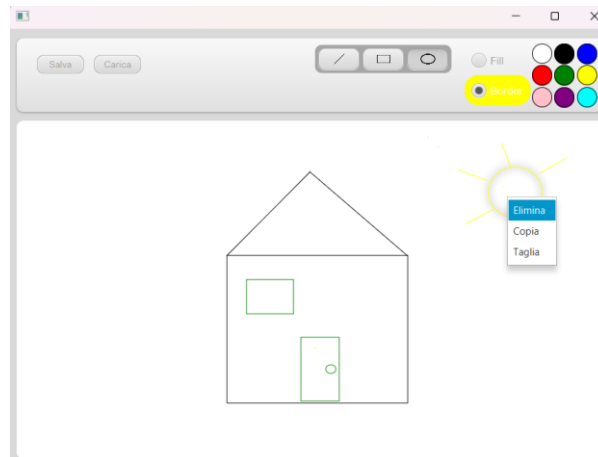


Figura 19: Screen 1 - Eliminazione di una forma

### 2.3.9.2 Test ID: 9.2

**Obiettivo del test:** Verificare che, cliccando “Elimina” dal menu contestuale, la forma venga effettivamente rimossa dall’area di disegno.

**Precondizioni:**

- L’applicazione è avviata
- È presente almeno una forma disegnata sulla canvas
- La forma è selezionata ed il menu contestuale è aperto

**Passi:**

1. Avviare l’applicazione
2. Selezionare una forma presente nell’area di disegno
3. Fare clic con il tasto destro all’interno del riquadro dell’applicazione
4. Cliccare sulla voce elimina

**Risultato atteso:** La forma selezionata viene correttamente rimossa dalla canvas e il riquadro viene aggiornato.

**Risultato ottenuto:** Dopo aver premuto il tasto “Elimina” la figura è stata effettivamente eliminata dall’area di disegno.

**Risultato:** Successo

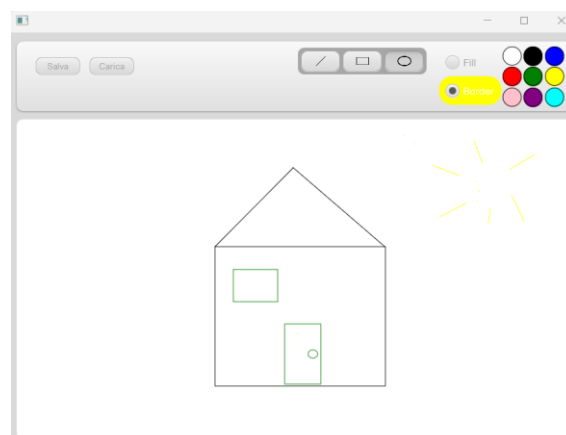


Figura 20: Screen 2 - Eliminazione di una forma

## 2.3.10 US10: Copia e Incolla

### 2.3.10.1 Test ID: 10.1

**Obiettivo del test:** Verificare che le voci "Copia" e "Incolla" compaiano nel menu contestuale quando si seleziona una forma.

**Precondizioni:**

- L'applicazione è avviata
- È presente almeno una forma disegnata nell'area di disegno

**Passi:**

1. Selezionare una forma presente nell'area di disegno
2. Fare clic con il tasto destro sulla forma

**Risultato atteso:** Il menu contestuale mostra le voci "Copia" e "Incolla".

**Risultato ottenuto:** Le voci "Copia" e "Incolla" sono visibili nel menu contestuale solo quando una forma è selezionata.

**Risultato:** Successo

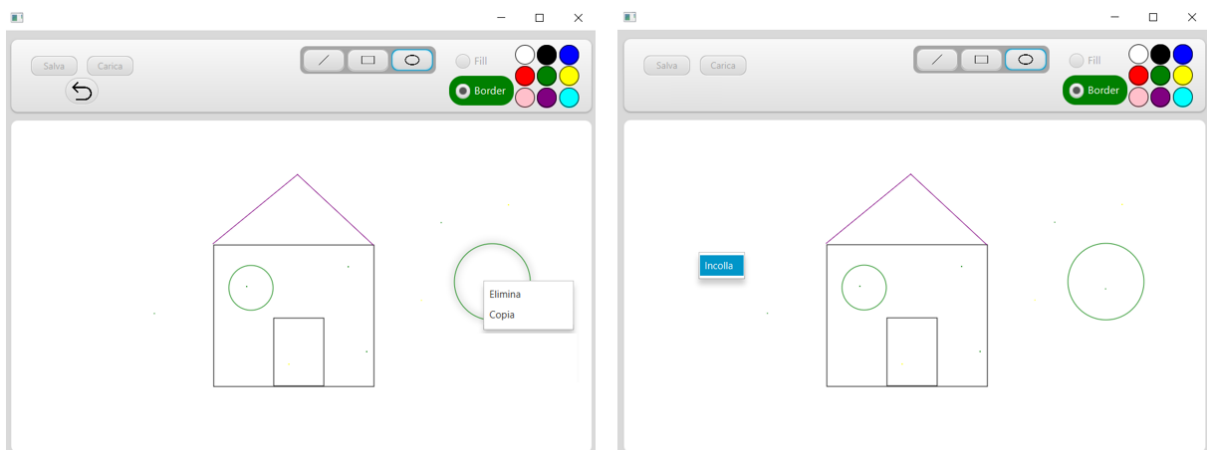


Figura 22: Screen 1 - Copia e Incolla

### 2.3.10.2 Test ID: 10.2

**Obiettivo del test:** Verificare che sia possibile copiare e incollare una forma utilizzando le voci del menu contestuale.

**Precondizioni:**

- L'applicazione è avviata
- È presente almeno una forma disegnata nell'area di disegno

**Passi:**

1. Selezionare una forma nell'area di disegno.
2. Fare clic con il tasto destro sulla forma selezionata.
3. Cliccare su "Copia" nel menu contestuale.
4. Deselezionare la forma.
5. Fare clic con il tasto destro in un punto qualsiasi del canvas e selezionare "Incolla"..

**Risultato atteso:** La forma copiata viene incollata correttamente nella posizione del canvas selezionata.

**Risultato ottenuto:** La nuova forma appare nel punto desiderato e ha le stesse caratteristiche della forma copiata.

**Risultato:** Successo

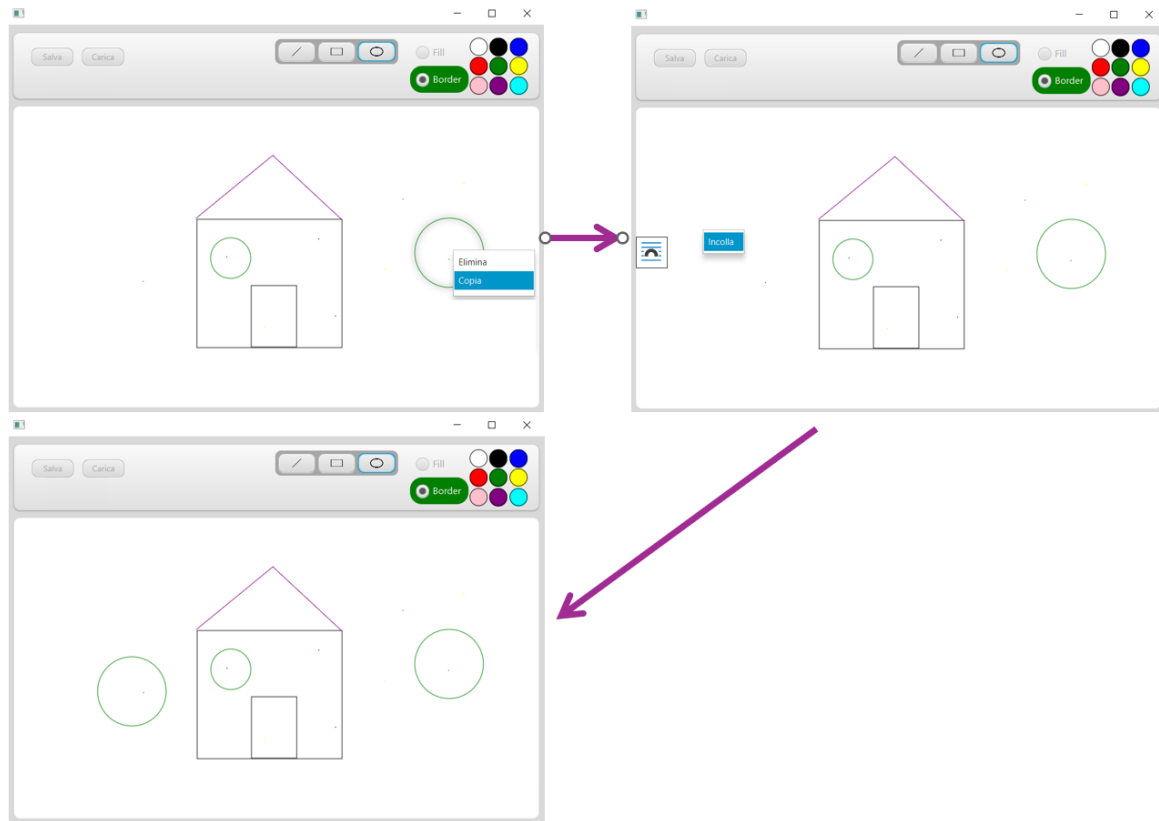


Figura 21: Screen - Copia e Incolla

## 2.3.11 US11: Taglia

### 2.3.11.1 Test ID: 11.1

**Obiettivo del test:** Verificare che la voce "Taglia" compaia nel menu contestuale quando si seleziona una forma.

**Precondizioni:**

- L'applicazione è avviata
- È presente almeno una forma disegnata sulla canvas

**Passi:**

1. Selezionare una forma nell'area di disegno.
2. Fare clic con il tasto destro sulla forma

**Risultato atteso:** Il menu contestuale mostra la voce "Taglia".

**Risultato ottenuto:** La voce "Taglia" è visibile e selezionabile nel menu contestuale solo quando una forma è selezionata.

**Risultato:** Successo

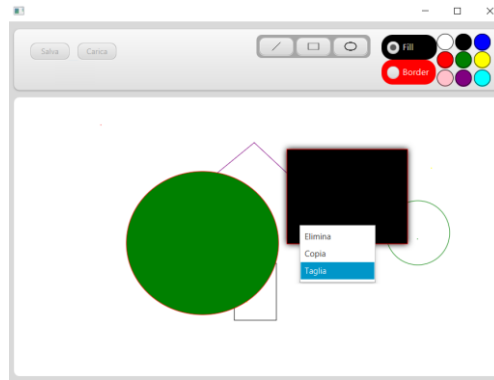


Figura 22: Screen 1 - Taglia

### 2.3.11.2 Test ID: 11.2

**Obiettivo del test:** Verificare che cliccando su "Taglia" una forma venga rimossa, e che sia possibile incollarla in un'altra posizione.

**Precondizioni:**

- L'applicazione è avviata
- È presente almeno una forma disegnata nell'area di disegno

**Passi:**

1. Selezionare una forma nell'area di disegno.
2. Fare clic con il tasto destro sulla forma selezionata.
3. Selezionare "Taglia" dal menu contestuale.
4. Deselezionare la forma.
5. Fare clic con il tasto destro in un punto qualsiasi del canvas.
6. Selezionare "Incolla" dal menu contestuale.

**Risultato atteso:** La forma viene rimossa dall'area di disegno e incollata nel nuovo punto selezionato.

**Risultato ottenuto:** La forma viene tagliata e riposizionata correttamente sul canvas.

**Risultato:** Successo

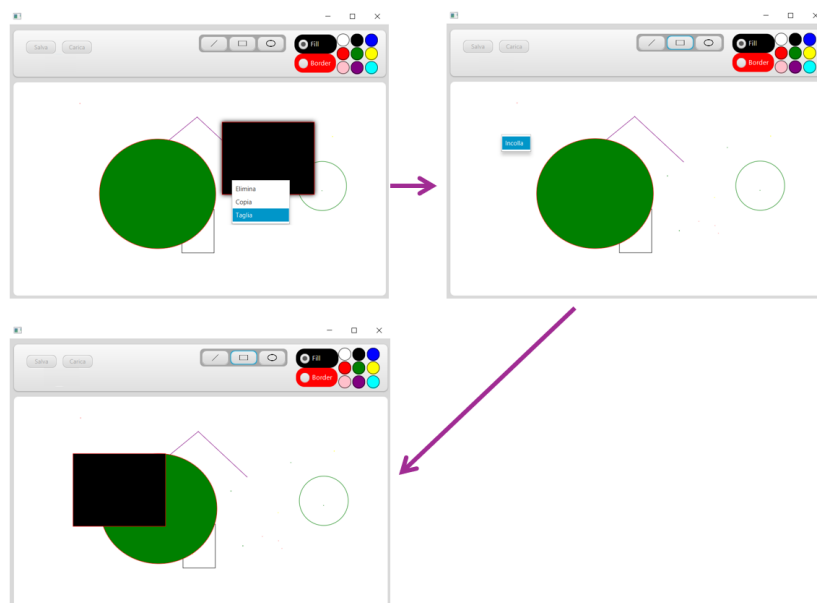


Figura 23: Screen 2 - Taglia

## 2.4 First Sprint Backlog

Durante questa sprint, il team ha lavorato in modo efficace e collaborativo, riuscendo a completare tutti i task pianificati all'interno dello Sprint Backlog. Le user stories previste sono state tutte portate a termine e marcate come Done, in piena conformità con i criteri stabiliti nella Definition of Done. Questo risultato riflette una buona pianificazione iniziale, un'efficace suddivisione dei compiti e un costante monitoraggio dell'avanzamento durante lo sprint. Il rispetto della Definition of Done garantisce inoltre che ogni storia sia stata adeguatamente testata, documentata e integrata nel prodotto, assicurando un incremento di valore concreto e potenzialmente rilasciabile.

User Story	Story Points	Stato
Finestra iniziale	1	DONE
Aggiunta forme geometriche	3	DONE
Forme supportate	3	DONE
Scelta colore del bordo delle forme	3	DONE
Scelta colore di riempimento delle forme	3	DONE
Salvataggio dei disegni	5	DONE
Caricamento dei disegni	5	DONE
Selezione forma	3	DONE
Eliminazione di una forma	1	DONE
Copia e incolla	3	DONE
Taglia	2	DONE

Tabella 13: First Sprint Backlog - Report Fine Sprint

## 2.5 Updated Product Backlog

Backlog ID	User Story	SP	Backlog ID	User Story	SP	Backlog ID	User Story	SP
1	Finestra iniziale	1	13	Modifica della posizione di una forma	2	25	Rotazione di una forma	5
2	Aggiunta forme geometriche	3	14	Modifica del colore di una forma	2	26	Specchiatura	5
3	Forme supportate	3	15	Modifica delle dimensioni di una forma	5	27	Stretch di una forma	5
4	Scelta colore del bordo delle forme	3	16	Modifica del livello di una forma	3	28	Selezione Multipla	3
5	Scelta colore di riempimento delle forme	3	17	Modifica del livello di zoom	5	29	Grouping	5
6	Salvataggio dei disegni	5	18	Area di disegno espandibile	3	30	Ungrouping	3
7	Caricamento dei disegni	5	19	Scroll area di disegno	2	31	Shape creation commands	13
8	Selezione forma	3	20	Visualizzazione della griglia	3	32	Memorizzazione dei "Shape creation commands"	8
9	Eliminazione di una forma	1	21	Attivazione e disattivazione della griglia	1	33	Esportazione di uno "Shapes library file"	8
10	Copia e incolla	3	22	Cambiare dimensioni griglia	3	34	Importazione di uno "Shapes library file"	8
11	Taglia	2	23	Poligoni irregolari	8			

12	Undo	5	24	Stringhe	5			
----	------	---	----	----------	---	--	--	--

Tabella 14: Product backlog aggiornato alla seconda sprint

## 2.6 Project Burndown chart:

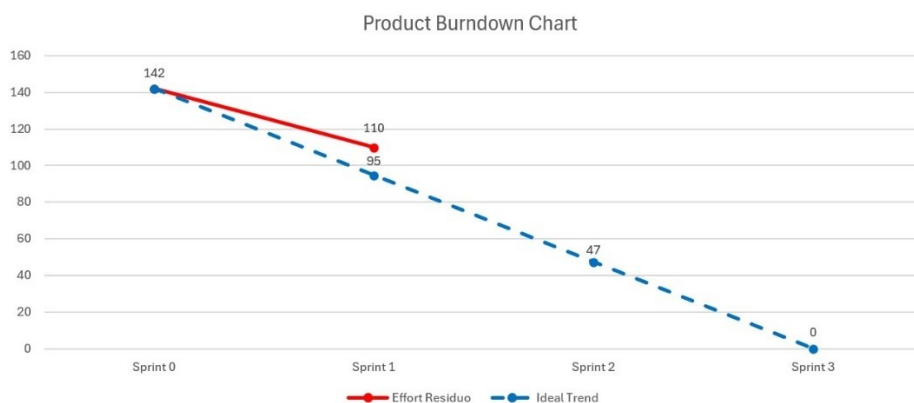


Figura 2.6.1 - Burndown Chart (First Sprint Update)

Il grafico mostra l'andamento del lavoro rimanente (espresso in story points) sprint-by-sprint. La linea tratteggiata blu rappresenta il trend ideale di completamento del progetto, mentre la linea rossa mostra l'andamento effettivo.

Al termine della prima sprint, il lavoro residuo è di 110 SP, il team ha leggermente sottoperformato rispetto alla previsione ideale (riduzione a 95 SP). Tuttavia, la velocity stimata è stata rispettata, mantenendo un ritmo di lavoro sostenibile.

Questo dato è particolarmente significativo: la velocity raggiunta può essere utilizzata come base di riferimento per la pianificazione della prossima sprint.

## 2.7 First Sprint - Review

### 2.7.1 Project velocity measured:

Durante la fase di Planning della prima Sprint era stata stimata una velocity pari a 32 SP. Alla fine della sprint tale stima risulta essere ben bilanciata, in quanto il team è riuscito a “bruciare” tutti gli SP previsti e di conseguenza a completare tutte le storie presenti nello Sprint Backlog.

La suddivisione del lavoro è stata equa, infatti, rispetto ai task previsti, i componenti della squadra hanno adempiuto ad una media di 8 story points, cadauno.

Nome Membro	Story points
Alessio Bottiglieri	8.15
Ciro Cutolo	8.45
Gaetano Frasca	7.70
Gennaro D'Ambrosio	7.70

Tabella 15: Suddivisione SP tra i membri del team - 1st Sprint

*Nota: la suddivisione degli story points è stata effettuata rispetto ai task e alla loro importanza nel contesto della storia da cui sono stati determinati.*

### 2.7.2 Problemi riscontrati durante l'implementazione

Nel corso della sprint non sono emersi technical debt da portare al prossimo sprint, in quanto tutti gli errori e i problemi sono stati individuati e risolti grazie alla cross review applicata all'intero codice. Questo processo di revisione incrociata ha permesso di mantenere alta la qualità del software e di intervenire tempestivamente su eventuali criticità. Se dovessimo individuare una funzionalità particolarmente impegnativa, sarebbe l'implementazione dell'inserimento delle forme sul canvas, che ha richiesto un notevole investimento di tempo soprattutto nella fase iniziale dello sprint

### 2.7.3 Modifiche da apportare al Product Backlog

Durante questo sprint non è stato necessario correggere o modificare in modo significativo le user story, il che dimostra che la fase di planning iniziale e la fase di backlog grooming sono state eseguite correttamente. Le user story erano chiare, ben definite e in linea con la comprensione del team fin dall'inizio. Tuttavia, il lavoro svolto durante questo sprint sarà prezioso per quello successivo, in particolare per aiutarci a definire e raffinare meglio le user story a grana grossa previste per i prossimi sviluppi.



## 2.8 First Sprint - Retrospective

Durante la Sprint Retrospective, il team ha utilizzato lo Starfish Diagram come strumento visivo e collaborativo per riflettere sull'andamento della sprint conclusa. Questo approccio ha permesso di raccogliere in modo strutturato opinioni, esperienze e suggerimenti da parte di tutti i membri del team, facilitando una discussione costruttiva.

### 2.8.1 Starfish Diagram

#### 2.8.1.1 Stop (*things to stop doing*):

- Perdere tempo su dettagli secondari prima di completare le funzionalità principali.
- Procrastinare la scrittura dei test automatici fino a fine implementazione.
- Modificare l'architettura del sistema, senza comunicarlo
- Modificare il codice altrui, senza preavviso

#### 2.8.1.2 Less of:

- Comunicazione frammentata sui canali informali
- Lavoro in parallelo su task fortemente dipendenti

#### 2.8.1.3 Keep doing:

- Suddivisione equa dei task tra i membri del team.
- Rispetto della velocity prevista e delle stime iniziali.
- Uso efficace della GitHub repository e Trello board.

#### 2.8.1.4 More of (*more things to do*):

- Scrivere commenti e documentazione progressiva, durante lo sviluppo.
- Comunicare in maniera esplicita i cambiamenti che si reputano necessari, anche se minimi.

#### 2.8.1.5 Start (*things to start doing*):

- Tenere traccia degli imprevisti per migliorare la stima nei prossimi sprint.

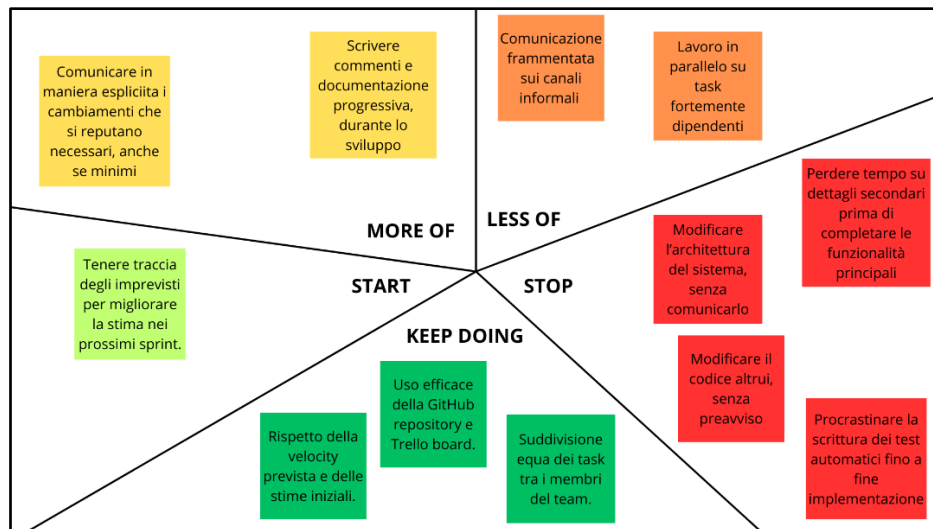


Figura 2: Starfish Diagram - 1st Sprint

## 3 Second sprint

### 3.1 Second Sprint Planning

#### 3.1.1 Modifiche Product Backlog

Durante questo sprint, si è reso necessario rivedere il Product Backlog sulla base delle conoscenze acquisite dal lavoro completato nella prima sprint. In particolare, la User Story #16 ("Modifica del livello di una forma") è stata rivalutata e i suoi SP sono stati aumentati da 3 a 5, poiché l'implementazione ha rivelato un livello di complessità superiore a quello stimato inizialmente.

Backlog ID	User Story	SP		Backlog ID	User Story	SP
16	Modifica del livello di una forma	3	➔	16	Modifica del livello di una forma	5

Inoltre, il team ha individuato l'opportunità di migliorare la chiarezza e la coerenza delle storie utente correlate. In particolare, la User Story #21, che riguardava l'attivazione e la disattivazione della griglia, si sovrapponeva in modo significativo alla #22, incentrata sulla funzionalità di visualizzazione della griglia. Data la loro stretta relazione e per ridurre la ridondanza, abbiamo deciso di consolidare queste due storie. La User Story #21 è stata rimossa e la descrizione e i criteri di accettazione della User Story #22 ("Visualizzazione della griglia") sono stati aggiornati per incorporare le funzionalità di attivazione e disattivazione. Questa modifica garantisce un backlog più snello e riflette una rappresentazione più accurata dell'interazione dell'utente con il componente griglia.

Di seguito è riportata la nuova User Story #21 ("Visualizzazione della griglia"):

#### Story:

Come utente,  
voglio poter visualizzare una griglia sull'area di disegno,  
così da posizionare più facilmente le forme.

#### Acceptance criteria:

- Data una sezione dell'interfaccia per il controllo della griglia,  
Quando l'utente visualizza la sezione interessata,  
Allora è presente un pulsante che permette l'attivazione e disattivazione della griglia
- Data l'area di disegno,  
Quando l'utente attiva la visualizzazione della griglia,  
Allora vengono mostrate linee orizzontali e verticali equidistanti, posizionate sopra lo sfondo ma sotto le forme.
- Data una griglia visibile,  
Quando l'utente interagisce con le forme (es. selezione o trascinamento),  
Allora le linee della griglia non interferiscono con tali operazioni.
- Data la visualizzazione della griglia attivata,  
Quando l'area di disegno viene aggiornata (es. tramite scroll o zoom),  
Allora la griglia si muove coerentemente con la canvas e adatta la distanza tra le linee proporzionalmente al livello di zoom.
- Data l'area di disegno,  
Quando l'utente disattiva la visualizzazione della griglia,  
Allora le linee della griglia scompaiono senza modificare le forme presenti.

**Priority:** Media

**Story points:** 3

### 3.1.2 Stima aggiornata della Project Velocity

Dati i risultati ottenuti nella prima sprint, è stato ritenuto opportuno incrementare la velocity prevista per la sprint corrente, fissandola a 35 story points.

### 3.1.3 Selezione delle User Stories per la seconda sprint

User Story	Story Points
Undo	5
Modifica della posizione di una forma	2
Modifica del colore di una forma	2
Modifica delle dimensioni di una forma	5
Modifica del livello di una forma	5

Modifica del livello di zoom	5	
Area di disegno espandibile	3	
Scroll area di disegno	2	
Visualizzazione della griglia	3	
Cambiare dimensioni griglia	3	
Totale:		35

Tabella 16 - Second sprint backlog

Durante la seconda sprint verranno implementate tutte le User Stories di priorità medio-alta/media, contenenti le rimanenti *shape editing operations*, e tutte le storie legate alla sezione *display of the drawing*. In totale, gli story points previsti per la seconda sprint sono '35'. Queste stime eccedono la velocity stimata per la prima sprint, poiché è stato ritenuto possibile incrementare il lavoro, rispetto al tempo a disposizione.

### 3.1.4 Suddivisione in task e assegnazione

#### 3.1.4.1 Undo (5 SP)

Obiettivo: permettere l'annullamento delle azioni effettuate sull'area di disegno

Tasks	Responsabile
Inserire un componente di interfaccia, utile ad azionare il comando undo	Ciro Cutolo
Implementare la logica dedita all'inversione delle modifiche apportate al riquadro di disegno	Ciro Cutolo
Implementare la logica che colleghi la pressione del pulsante utile per effettuare l'undo, all'effettiva azione del comando	Ciro Cutolo
Scrivere e/o eseguire test che verifichino il corretto funzionamento del componente undo	Gaetano Frasca

Tabella 17: US12

#### 3.1.4.2 Modifica della posizione di una forma (2 SP)

Obiettivo: Consentire all'utente di trascinare una forma selezionata per modificarne la posizione all'interno del canvas

Tasks	Responsabile
Implementare la logica che permette di muovere una forma, dopo averla selezionata	Gennaro D'Ambrosio

Scrivere e/o eseguire i test per accertare la possibilità di muovere le forme	Alessio Bottiglieri
---	---------------------

Tabella 18: Tasks US13

### 3.1.4.3 Modifica del colore di una forma (2 SP)

Obiettivo: Modifica del colore di contorno e riempimento di una forma già presente sulla canvas.

Tasks	Responsabile
Implementare la logica utile a cambiare il colore di una forma selezionata	Alessio Bottiglieri
Scrivere e/o eseguire i test atti a verificare l'effettiva possibilità di cambiare il colore di forme già presenti nel riquadro di disegno	Gennaro D'Ambrosio

Tabella 19: Tasks US14

### 3.1.4.4 Modifica delle dimensioni di una forma (5 SP)

Obiettivo: Ridimensionamento libero delle forme nel canvas.

Tasks	Responsabile
Implementare la logica utile a modificare le dimensioni di una forma, una volta selezionata	Gaetano Frasca
Scrivere e/o eseguire i test per verificare la possibilità di modificare le dimensioni di una forma già presente nel riquadro di disegno	Ciro Cutolo

Tabella 20: Tasks US15

### 3.1.4.5 Modifica del livello di una forma (5 SP)

Obiettivo: Gestione del livello di sovrapposizione delle forme.

Tasks	Responsabile
Implementare una struttura di livelli di disegno	Alessio Bottiglieri
Implementare la logica utile a inserire un componente del disegno in uno dei livelli liberi	Alessio Bottiglieri
Implementare la logica utile a modificare il livello di una forma già presente nel riquadro di disegno	Alessio Bottiglieri
Scrivere e/o eseguire il test per verificare che sia possibile spostare un elemento, presente	Ciro Cutolo

nel riquadro di disegno, in un diverso livello di visualizzazione	
---	--

Tabella 21: Tasks US16

#### 3.1.4.6 Modifica del livello di zoom (5 SP)

Obiettivo: Controllo dello zoom sull'area di disegno.

Tasks	Responsabile
Inserire un componente di interfaccia, utile a modificare il livello di zoom	Gennaro D'Ambrosio
Implementare la logica dedita allo modifica del livello di zoom	Gennaro D'Ambrosio
Implementare la logica che colleghi la pressione del pulsante utile per modificare il livello di zoom all'effettiva azione del comando	Gennaro D'Ambrosio
Scrivere e/o eseguire test che verifichino il corretto funzionamento del componente di zoom	Gaetano Frasca

Tabella 22: Tasks US17

#### 3.1.4.7 Area di disegno espandibile (3 SP)

Obiettivo: Canvas espandibile oltre i limiti della finestra visibile.

Tasks	Responsabile
Implementare un riquadro di disegno espandibile e scrollabile	Ciro Cutolo
Implementare un meccanismo che permetta di adattare dinamicamente le dimensioni del riquadro di disegno, considerando anche forme al di fuori della zona visibile	Ciro Cutolo
Scrivere e/o eseguire test per accertare il corretto funzionamento del riquadro espandibile	Gennaro D'Ambrosio

Tabella 23: Tasks US18

#### 3.1.4.8 Scroll area di disegno (2 SP)

Obiettivo: Scorrimento dell'area di disegno per navigare nei contenuti.

Tasks	Responsabile
Inserire un componente di interfaccia utile a muoverci all'interno dell'area di disegno	Ciro Cutolo

Implementare la logica dedicata al movimento all'interno dell'area di disegno	Ciro Cutolo
Implementare la logica che colleghi l'interazione con il componente dedicato a muoverci all'interno dell'area di disegno con l'effettiva azione del comando	Ciro Cutolo
Scrivere e/o eseguire test che verifichino il corretto funzionamento del componente per muoverci all'interno dell'area di disegno	Alessio Bottiglieri

Tabella 24: Tasks US19

### 3.1.4.9 Visualizzazione della griglia (3 SP)

Obiettivo: Visualizzazione opzionale della griglia sul canvas.

Tasks	Responsabile
Inserire un componente di interfaccia utile ad attivare e disattivare la griglia nell'area di disegno	Gaetano Frasca
Implementare la logica dedicata all'inserimento e alla rimozione della griglia nell'area di disegno	Gaetano Frasca
Implementare la logica che colleghi l'interazione con il componente grafico utile all'attivazione e alla disattivazione della griglia con l'effettiva azione dei comandi	Gaetano Frasca
Scrivere e/o eseguire test che verifichino il corretto funzionamento dei componenti per l'inserimento e l'eliminazione della griglia dall'area di disegno	Gennaro D'Ambrosio

Tabella 25: Tasks US20

### 3.1.4.10 Cambiare dimensioni griglia (3 SP)

Obiettivo: Personalizzazione della dimensione della griglia

Tasks	Responsabile
Inserire un componente di interfaccia, utile a modificare la dimensione della griglia	Alessio Bottiglieri
Implementare la logica dedicata alla modifica della dimensione della griglia	Alessio Bottiglieri
Implementare la logica che colleghi la pressione del pulsante utile per modificare la dimensione della griglia all'effettiva azione del comando	Alessio Bottiglieri

Scrivere e/o eseguire test che verifichino il corretto funzionamento del componente perla modifica della dimensione della griglia

Gaetano Frasca

Tabella 26: Tasks US21

## 3.2 Second Sprint – Applicazione Pattern

### 3.2.1 Command Pattern

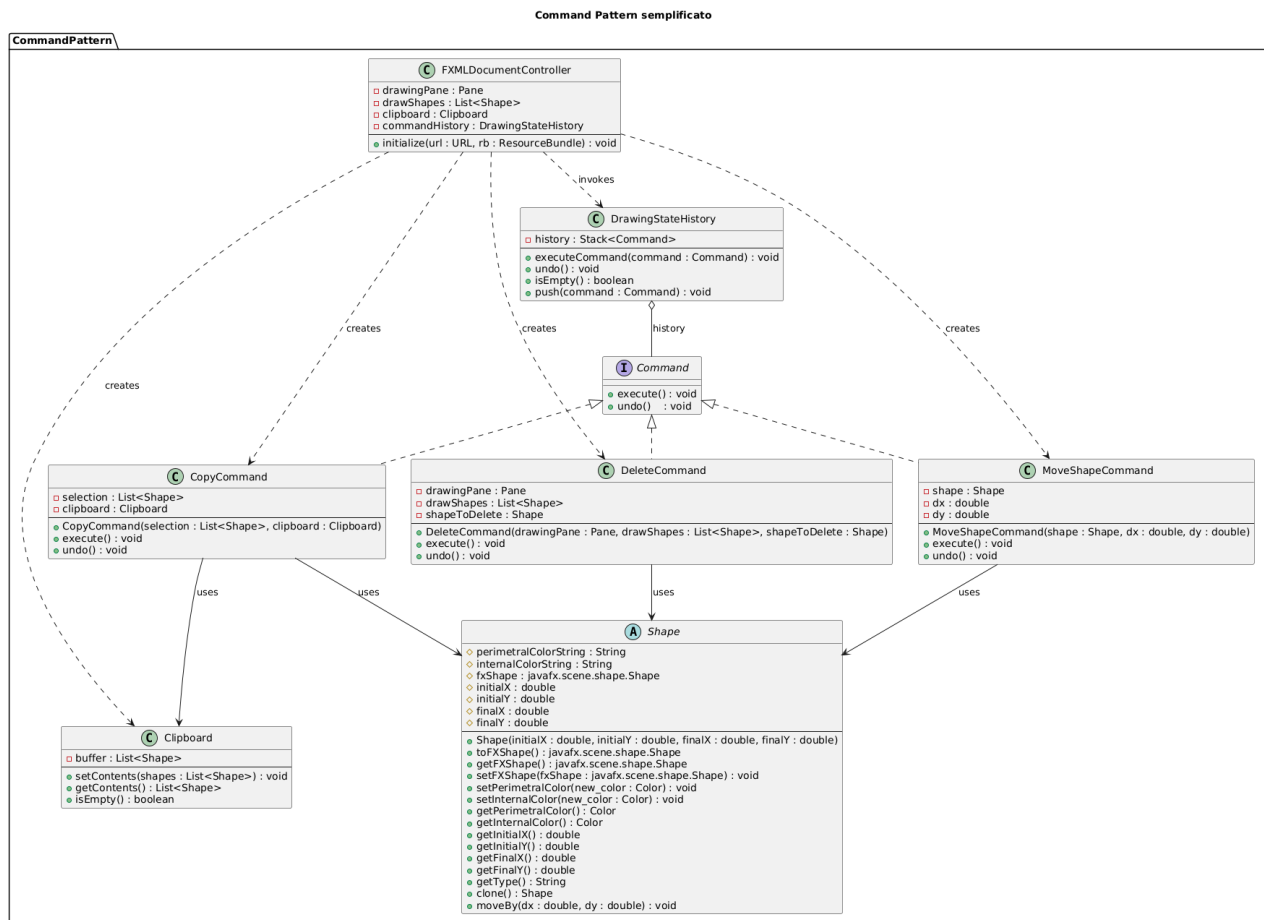


Figura 3: Command Pattern – Class Diagram

*Nota: per semplicità sono state incluse solo tre classi concrete. Inoltre sono stati mostrati solo gli attributi più importanti del Controller*

Il Command è un pattern comportamentale che permette di incapsulare un'azione sotto forma di oggetto, rendendo l'operazione (e il suo annullamento) indipendente dall'interfaccia utente che la invoca. Nella nostra applicazione di disegno è stato adottato questo pattern per realizzare la funzionalità di Undo e, più in generale, per uniformare la gestione di tutte le operazioni che l'utente può compiere sul canvas.

- L'interfaccia Command rappresenta il ruolo di *Command* nel pattern e definisce i metodi *execute()* e *undo()* che tutte le operazioni concrete devono implementare.



- Le classi concrete *InsertShapeCommand*, *DeleteCommand*, *MoveShapeCommand*, *ChangeColorCommand*, *CopyCommand*, *PasteCommand*, *BringForwardCommand*, *BringToFrontCommand*, *SendBackwardCommand*, *SendToBackCommand* (dette *ConcreteCommand*) implementano l'interfaccia *Command* e incapsulano singole operazioni sul canvas, mantenendo al loro interno lo stato necessario per permettere l'operazione di annullamento (*undo()*).
- La classe *DrawingStateHistory* riveste il ruolo di *Invoker*, gestendo l'esecuzione dei comandi tramite il metodo *executeCommand(command)* e mantenendo lo storico delle operazioni per abilitare l'Undo.
- Le classi *Shape* e *Clipboard*, insieme agli oggetti *Pane drawingPane* e *List<Shape> drawShapes* gestiti direttamente dal controller, svolgono il ruolo di *Receiver*. Questi componenti eseguono concretamente le operazioni richieste dai *ConcreteCommand*, come lo spostamento delle forme, la modifica del colore, l'inserimento o la cancellazione nel canvas, le operazioni di copia/incolla ecc.
- Il *FXMLDocumentController* agisce da Client: interpreta le azioni dell'utente, istanzia e configura i comandi concreti e li passa all'Invoker per l'esecuzione.

L'alternativa, cioè gestire ogni azione direttamente nel controller, avrebbe comportato

- Logica di undo replicata per ogni caso d'uso
- Codice del controller monolitico, poco manutenibile e difficilmente testabile

Il *Command* pattern permette invece una struttura modulare, in linea con il *Single Responsibility Principle*: ogni comando si occupa di una sola cosa, e la gestione della memoria della cronologia dei comandi è centralizzata. Questo ha migliorato la coesione del progetto e, insieme ai pattern *Prototype* e *Simple Factory*, ha reso il codice facilmente estendibile e privo di dipendenze rigide.

## 3.3 Second Sprint – Testing

### 3.3.1 US12: Undo

#### 3.3.1.1 Test ID: 12.1

**Obiettivo del test:** Verificare che il pulsante “Undo” compaia nella toolbar.

**Precondizioni:**

- L'applicazione è avviata

**Passi:**

1. Osservare l'interfaccia

**Risultato atteso:** Nella toolbar è presente un pulsante dedicato alla funzione “Undo”.

**Risultato ottenuto:** Nella toolbar è effettivamente presente un pulsante che permette di annullare l'ultima operazione, questo è raffigurato da (↶).

**Risultato:** Successo

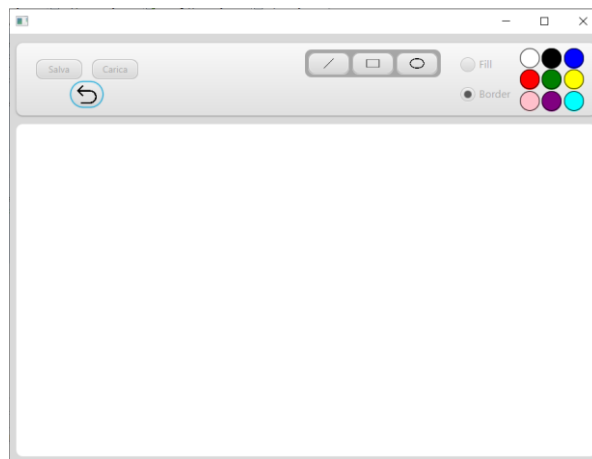


Figura 4: Screen 1: Undo

#### 3.3.1.2 Test ID: 12.2

**Obiettivo del test:** Verificare che tramite il pulsante “Undo”, sia possibile annullare qualsiasi operazione eseguita.

**Precondizioni:**

- L'applicazione è avviata

**Passi:**

1. Inserire una forma

**Risultato atteso:** Cliccando un numero arbitrario di volte il pulsante “undo”, è possibile tornare allo stato precedente del disegno (finestra vuota).

**Risultato ottenuto:** Cliccando il pulsante “undo” è effettivamente possibile annullare l'inserimento della forma.

**Risultato:** Successo

**Nota:** è stata testata la possibile annullare qualsiasi operazione svolta, senza alcun limite, ma per non appesantire inutilmente la documentazione si è mostrato un caso semplice.

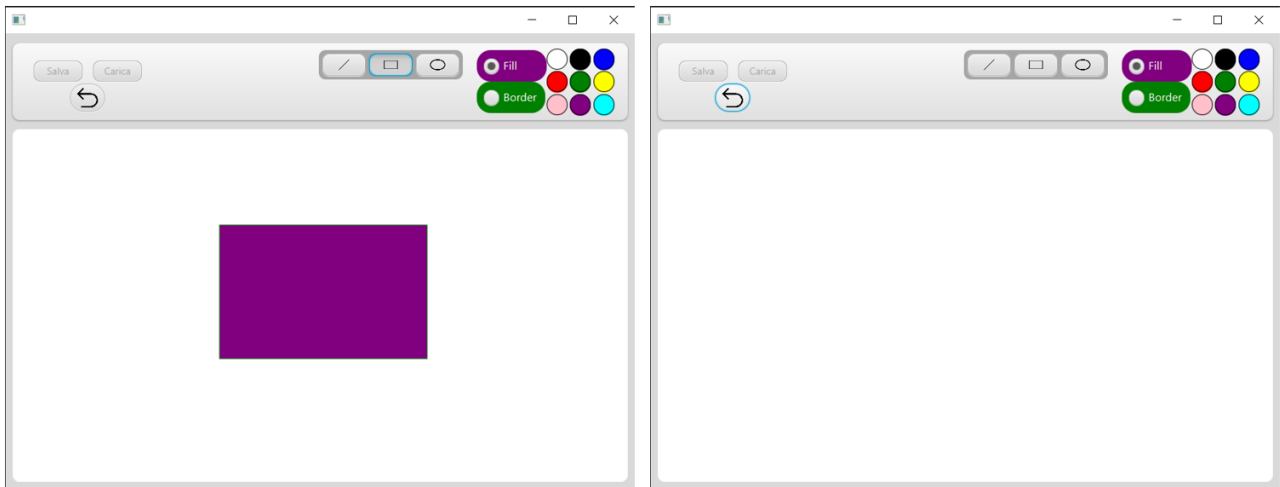


Figura 5: Screen 2 - Undo

### 3.3.2 US13: Modifica della posizione di una forma

#### 3.3.2.1 Test ID: 13.1

**Obiettivo del test:** Verificare che l'utente possa selezionare una forma nel canvas e trascinarla per modificarne la posizione.

**Precondizioni:**

- L'applicazione è avviata
- E' presente almeno una forma nel riquadro di disegno

**Passi:**

1. Cliccare su una forma già presente nel canvas per selezionarla.
2. Tenere premuto il tasto sinistro del mouse sopra la forma selezionata.
3. Trascinare il cursore in una nuova posizione all'interno del canvas.
4. Rilasciare il tasto del mouse per confermare la nuova posizione.

**Risultato atteso:** La forma segue il cursore durante il trascinamento e, al rilascio del mouse, viene posizionata nella nuova area. Il movimento avviene in tempo reale e con fluidità.

**Risultato ottenuto:** La forma può essere selezionata e spostata liberamente all'interno del canvas. La posizione della forma è aggiornata in tempo reale e correttamente

**Risultato:** Successo

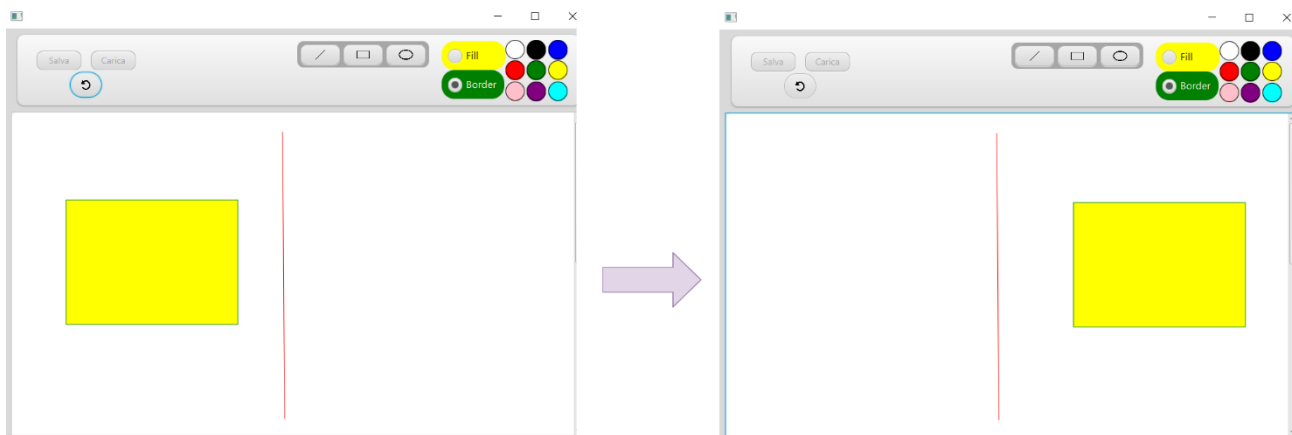


Figura 6: Screen 1 - Modifica posizione di una forma

### 3.3.3 US14: Modifica del colore di una forma

**Obiettivo del test:** Verificare che l'utente possa selezionare una forma nel canvas per modificarne il colore.

**Precondizioni:**

- L'applicazione è avviata
- E' presente almeno una forma nel riquadro di disegno

**Passi:**

1. Cliccare su una forma già presente nel canvas per selezionarla.
2. Scegliere un colore bordo e uno di riempimento dall'apposita sezione nell'interfaccia.
3. Cliccare su "cambia colore"

**Risultato atteso:** La forma selezionata presenterà i colori di bordo e di riempimento scelti.

**Risultato ottenuto:** La forma selezionata presenta i colori di bordo e di riempimento scelti.

**Risultato:** Successo

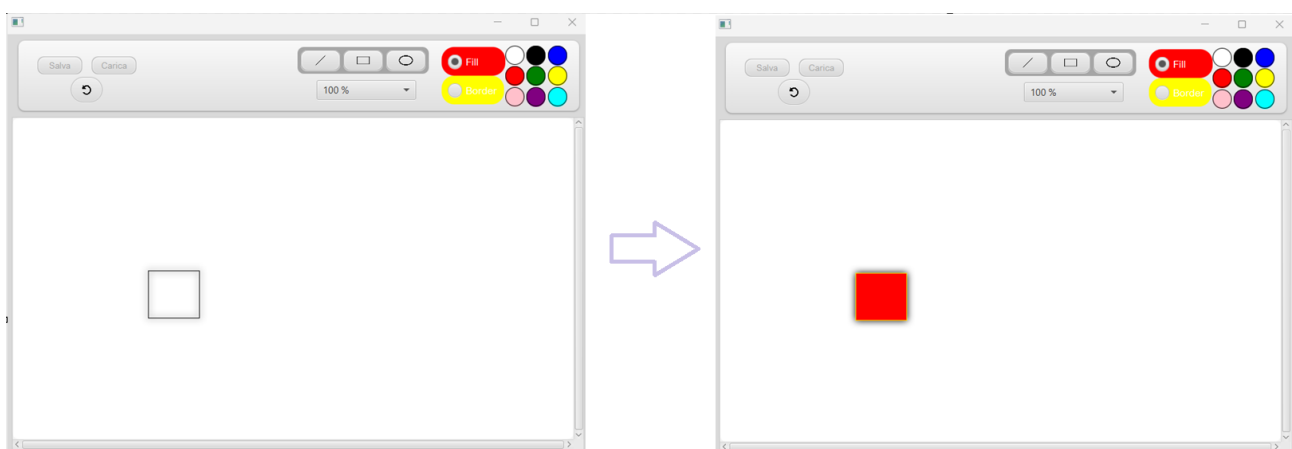


Figura 7: Screen 1 - Modifica del colore di una forma

### 3.3.4 US15: Modifica delle dimensioni di una forma

**Obiettivo del test:** Verificare che l'utente possa modificare la dimensione di una forma selezionata.

**Precondizioni:**

- L'applicazione è avviata
- E' presente almeno una forma nel riquadro di disegno

**Passi:**

4. Cliccare su una forma già presente nel canvas per selezionarla.
5. Cliccare su "Ridimensiona"
6. Scegliere le nuove dimensioni

**Risultato atteso:** La forma selezionata verrà correttamente ridimensionata.

**Risultato ottenuto:** La forma selezionata è stata correttamente ridimensionata.

**Risultato:** Successo

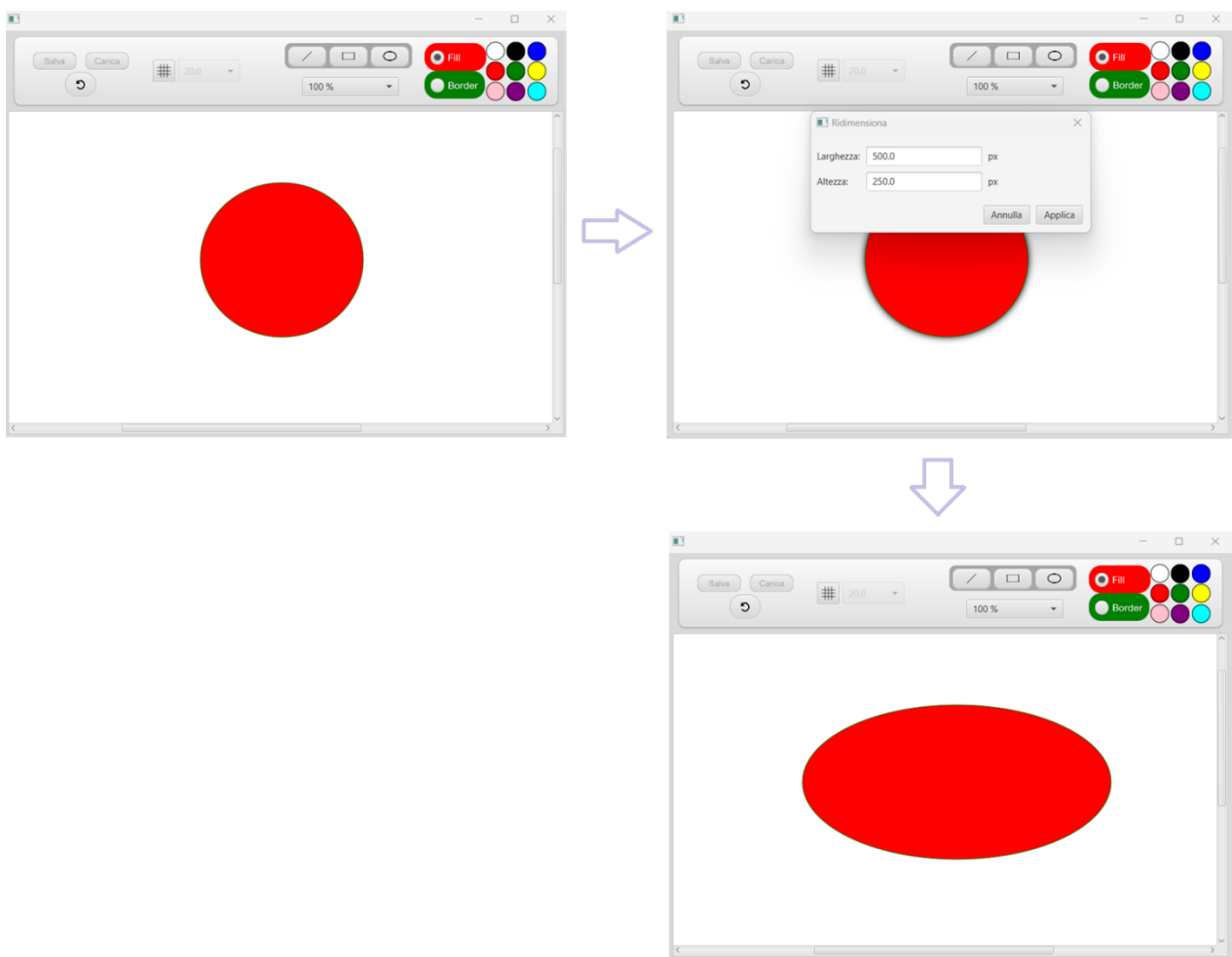


Figura 8: Screen 1 - Ridimensionamento di una forma

### 3.3.5 US16: Modifica del livello di una forma

**Obiettivo del test:** Verificare che l'utente possa modificare il livello di una forma

### Precondizioni:

- L'applicazione è avviata
- Sono presenti almeno due forme nel riquadro di disegno

### Passi:

7. Cliccare su una forma già presente nel canvas per selezionarla.
8. Cliccare su "Cambia livello"
9. Scegliere il nuovo livello

**Risultato atteso:** Il livello della forma selezionata verrà correttamente modificato.

**Risultato ottenuto:** Il livello della forma selezionata è stato correttamente modificato.

**Risultato:** Successo

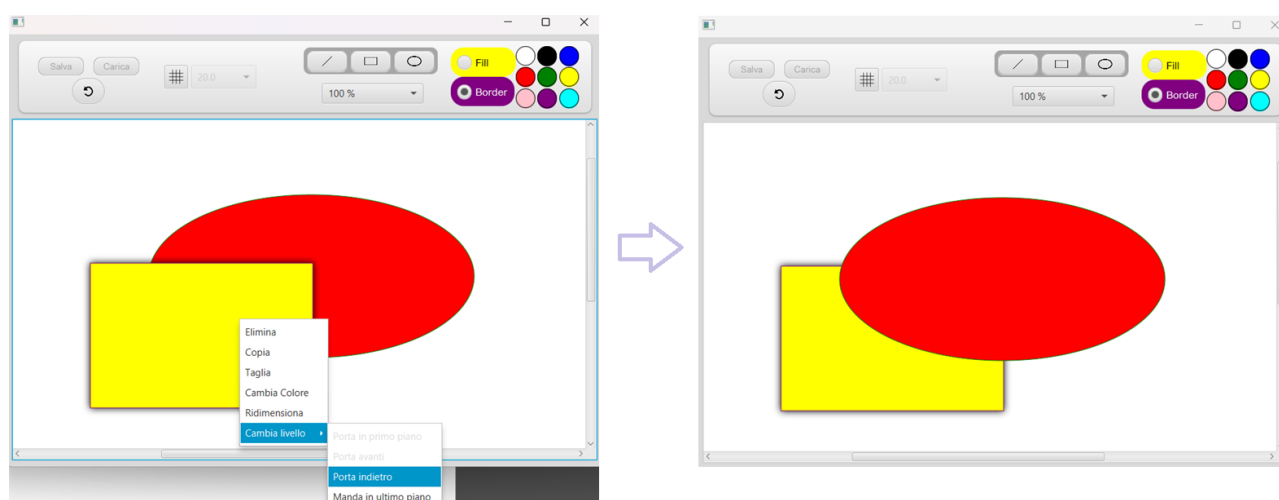


Figura 9: Screen1-Modifica livello di una forma

## 3.3.6 US17: Modifica del livello di zoom

### 3.3.6.1 Test ID 17.1

**Obiettivo del test:** Verificare che tramite un componente dell'interfaccia (context menu) sia possibile zoomare l'area di lavoro.

### Precondizioni:

- L'applicazione è avviata
- Sono presenti delle forme già inserite nell'area di lavoro

### Passi:

1. Cliccare con il tasto destro sull'area di lavoro per aprire il context menu.
2. Selezionare un livello di zoom (es. 150%).

**Risultato atteso:** L'area di lavoro viene ingrandita o ridotta in base al livello di zoom selezionato, e le forme presenti si adattano proporzionalmente alla nuova scala.

**Risultato ottenuto:** L'area di lavoro si adatta correttamente al livello di zoom selezionato, e le forme presenti sono scalate in proporzione.

**Risultato:** Successo

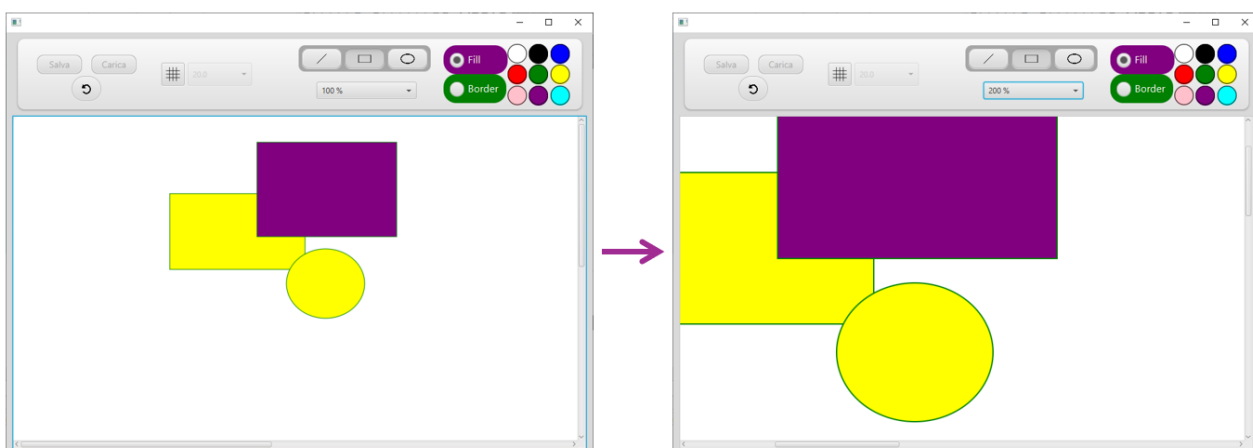


Figura 10: Screen - Modifica del livello di zoom

### 3.3.7 US18: Area di disegno espandibile

#### 3.3.7.1 Test ID 18.1

**Obiettivo del test:** Verificare che l'utente possa espandere l'area di disegno tramite il trascinamento di una forma sul bordo dell'area di disegno durante l'inserimento.

**Precondizioni:**

- L'applicazione è avviata

**Passi:**

1. Scegliere una forma da inserire.
2. Trascinare durante l'inserimento la forma verso il bordo.

**Risultato atteso:** Il bordo si amplierà per contenere la forma che stiamo trascinando.

**Risultato ottenuto:** Il bordo si amplia per contenere la forma che stiamo trascinando.

**Risultato:** Successo

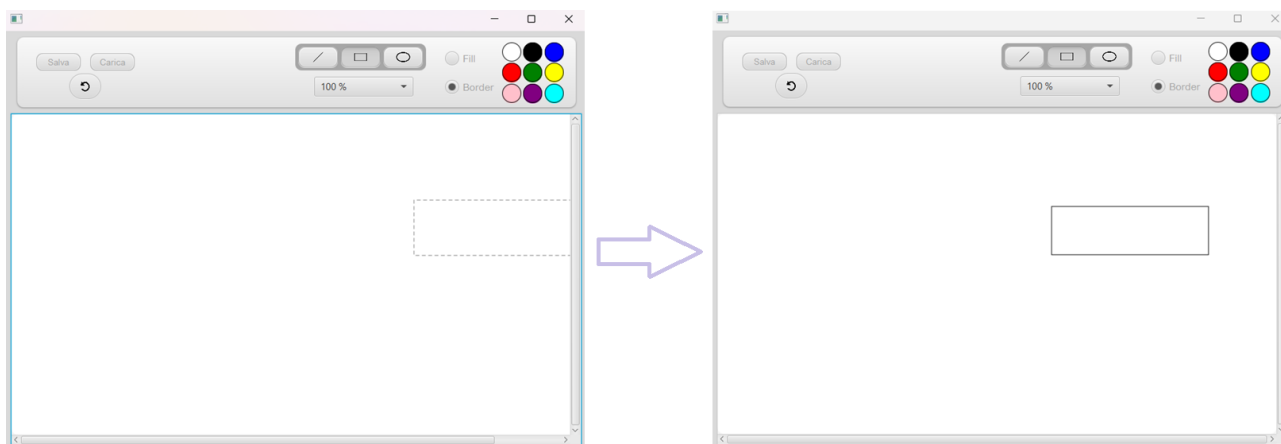


Figura 11: Screen 1 - Area di disegno espandibile

### 3.3.8 US19: Scroll area di disegno

#### 3.3.8.1 Test ID: 19.1

**Obiettivo del test:** Verificare la presenza delle scrollbar (orizzontale e verticale) quando l'area di disegno supera le dimensioni della finestra visibile.

**Precondizioni:**

- L'applicazione è avviata
- L'area di disegno ha dimensioni maggiori della finestra visibile

**Passi:**

1. Avviare l'applicazione
2. Aumentare la dimensione della canvas oltre la finestra visibile (ad esempio con l'inserimento di una forma che eccede i limiti della finestra)
3. Osservare l'interfaccia per verificare la presenza delle scrollbar.

**Risultato atteso:** Quando l'area di disegno eccede i limiti della finestra, sono visibili una scrollbar verticale e una orizzontale che permettono lo scorrimento.

**Risultato ottenuto:** Le scrollbar sono presenti quando il contenuto supera le dimensioni dell'interfaccia.

**Risultato:** Successo

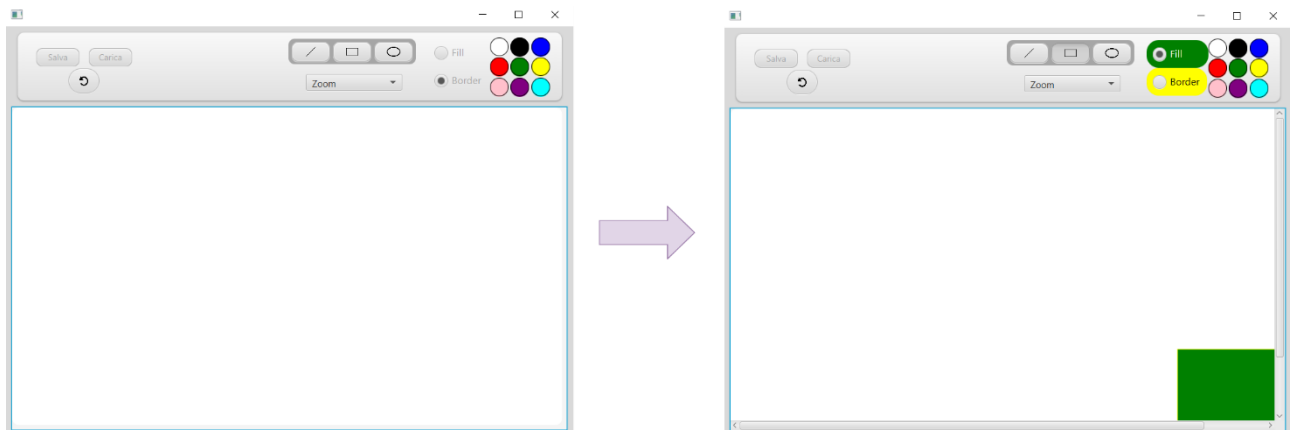


Figura 12: Screen 1 - Scroll area di disegno

#### 3.3.8.2 Test ID: 19.2

**Obiettivo del test:** Verificare il corretto funzionamento dello scorrimento dell'area di disegno tramite le scrollbar.

**Precondizioni:**

- L'applicazione è avviata
- L'area di disegno ha dimensioni maggiori della finestra visibile



- Le scrollbar sono visibili

**Passi:**

1. Avviare l'applicazione
2. Aumentare la dimensione della canvas oltre la finestra visibile (ad esempio con l'inserimento di una forma che eccede i limiti della finestra)
3. Interagire con la scrollbar verticale e quella orizzontale
4. Verificare che l'area di disegno si sia espansa e che il disegno sia completamente visibile

**Risultato atteso:** Lo spostamento tramite scrollbar consente di visualizzare correttamente tutte le parti del disegno, senza distorsioni o perdite di contenuto.

**Risultato ottenuto:** L'interazione con le scrollbar permette di navigare in tutte le direzioni, la canvas si estende effettivamente oltre la parte di schermo visibile e contiene le forme inserite.

**Risultato:** Successo

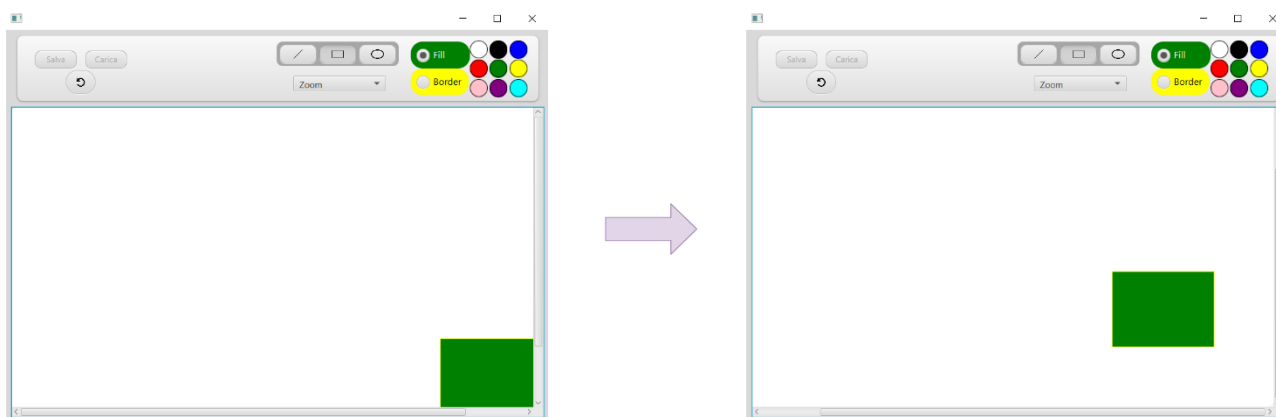


Figura 13: Screen 2 - Scroll area di disegno

### 3.3.9 US20: Visualizzazione della griglia

**Obiettivo del test:** Verificare che l'utente possa inserire e rimuovere una griglia dall'area di disegno.

**Precondizioni:**

- L'applicazione è avviata

**Passi:**

3. Cliccare sull'elemento grafico per l'inserimento della griglia
4. Eventualmente ricliccarlo per rimuovere la griglia dall'area di disegno

**Risultato atteso:** La griglia viene aggiunta e rimossa all'interno dell'area di disegno quando clicco l'apposito pulsante dell'interfaccia utente.

**Risultato ottenuto:** La griglia viene aggiunta e rimossa all'interno dell'area di disegno quando clicco l'apposito pulsante dell'interfaccia utente.

**Risultato:** Successo

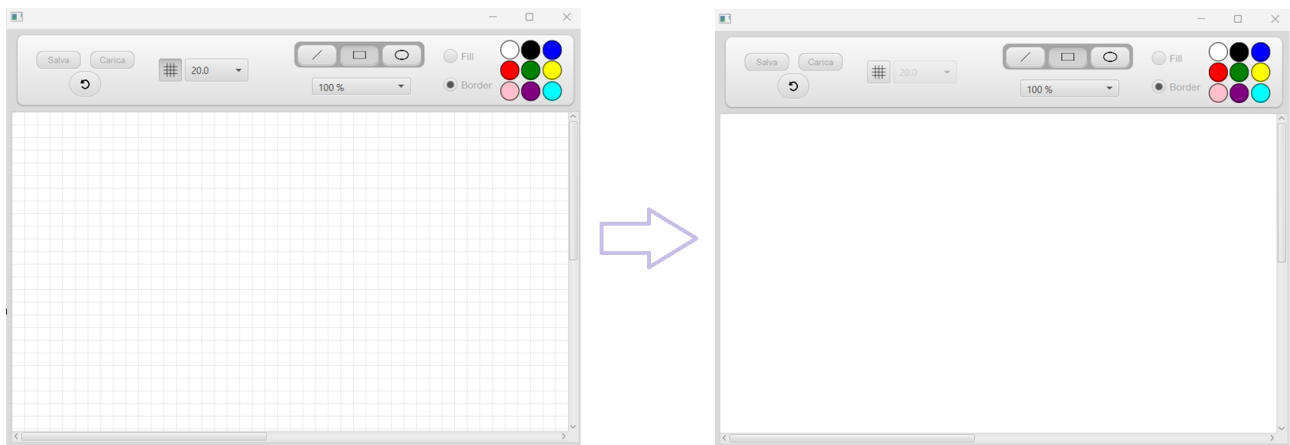


Figura 14: Screen 1 - Visualizzazione griglia

### 3.3.10 US21: Cambiare dimensioni griglia

**Obiettivo del test:** Verificare che l'utente possa modificare la dimensione della griglia

**Precondizioni:**

- L'applicazione è avviata
- La griglia è attiva

**Passi:**

5. Cliccare sull'elemento grafico per l'inserimento della griglia
6. Scegliere la dimensione della griglia dall'apposito elemento UI

**Risultato atteso:** La griglia viene ridimensionata in base alla grandezza selezionata.

**Risultato ottenuto:** La griglia viene ridimensionata in base alla grandezza selezionata.

**Risultato:** Successo

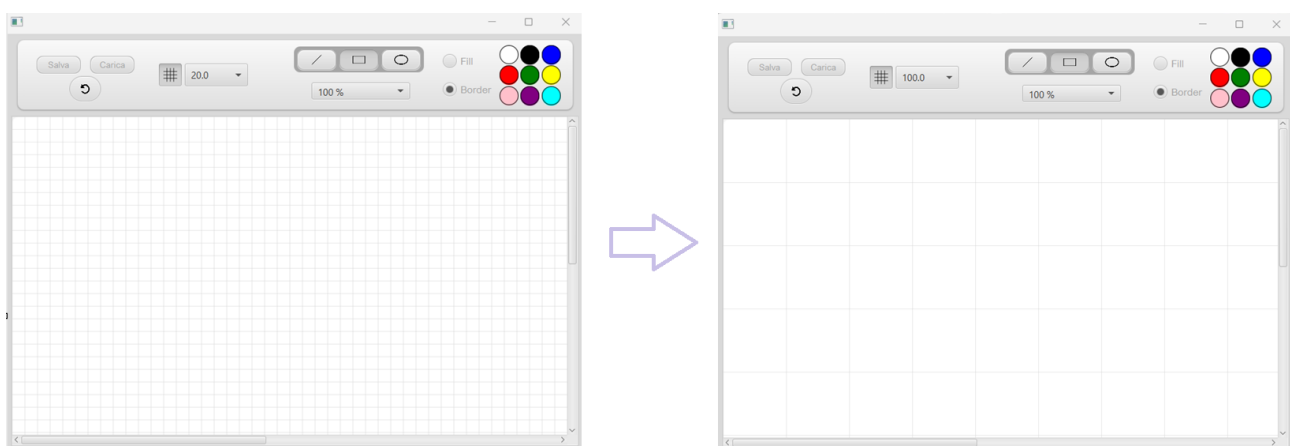


Figura 15: Screen 1 - Cambiare dimensione della griglia

### 3.4 Second Sprint Backlog

Durante questa sprint, il team ha lavorato in modo efficace e collaborativo, riuscendo a completare tutti i task pianificati all'interno dello Sprint Backlog. Le user stories previste sono state tutte portate a termine e marcate come Done, in piena conformità con i criteri stabiliti nella Definition of Done. Questo risultato riflette una buona pianificazione iniziale, un'efficace suddivisione dei compiti e un costante monitoraggio dell'avanzamento durante lo sprint. Il rispetto della Definition of Done garantisce inoltre che ogni storia sia stata adeguatamente testata, documentata e integrata nel prodotto, assicurando un incremento di valore concreto e potenzialmente rilasciabile.

ID	User Story	Story Points	Stato
12	Undo	5	DONE
13	Modifica della posizione di una forma	2	DONE
14	Modifica del colore di una forma	2	DONE
15	Modifica delle dimensioni di una forma	5	DONE
16	Modifica del livello di una forma	5	DONE
17	Modifica del livello di zoom	5	DONE
18	Area di disegno espandibile	3	DONE
19	Scroll area di disegno	2	DONE
20	Visualizzazione della griglia	3	DONE
21	Cambiare dimensioni griglia	3	DONE

Tabella 27: Second Sprint Backlog - Report Fine Sprint

### 3.5 Updated Product Backlog

Backlog ID	User Stories	SP	Backlog ID	User Stories	SP	Backlog ID	User Stories	SP
1	Finestra iniziale	1	12	Undo	5	24	Stringhe	5
2	Aggiunta forme geometriche	3	13	Modifica della posizione di una forma	2	25	Rotazione di una forma	5
3	Forme supportate	3	14	Modifica del colore di una forma	2	26	Specchiatura	5
4	Scelta colore del bordo delle forme	3	15	Modifica delle dimensioni di una forma	5	27	Stretch di una forma	5
5	Scelta colore di riempimento delle forme	3	16	Modifica del livello di una forma	5	28	Selezione Multipla	3
6	Salvataggio dei disegni	5	17	Modifica del livello di zoom	5	29	Grouping	5
7	Caricamento dei disegni	5	18	Area di disegno espandibile	3	30	Ungrouping	3
8	Selezione forma	3	19	Scroll area di disegno	2	31	Shape creation commands	13
9	Eliminazione di una forma	1	20	Visualizzazione della griglia	3	32	Memorizzazione dei "Shape creation"	8

							commands "	
10	Copia e incolla	3	22	Cambiare dimensioni griglia	3	33	Esportazion e di uno "Shapes library file"	8
11	Taglia	2	23	Poligoni irregolari	8	34	Importazio ne di uno "Shapes library file"	8

Tabella 28: Product backlog aggiornato al secondo sprint

### 3.6 Project Burndown chart:

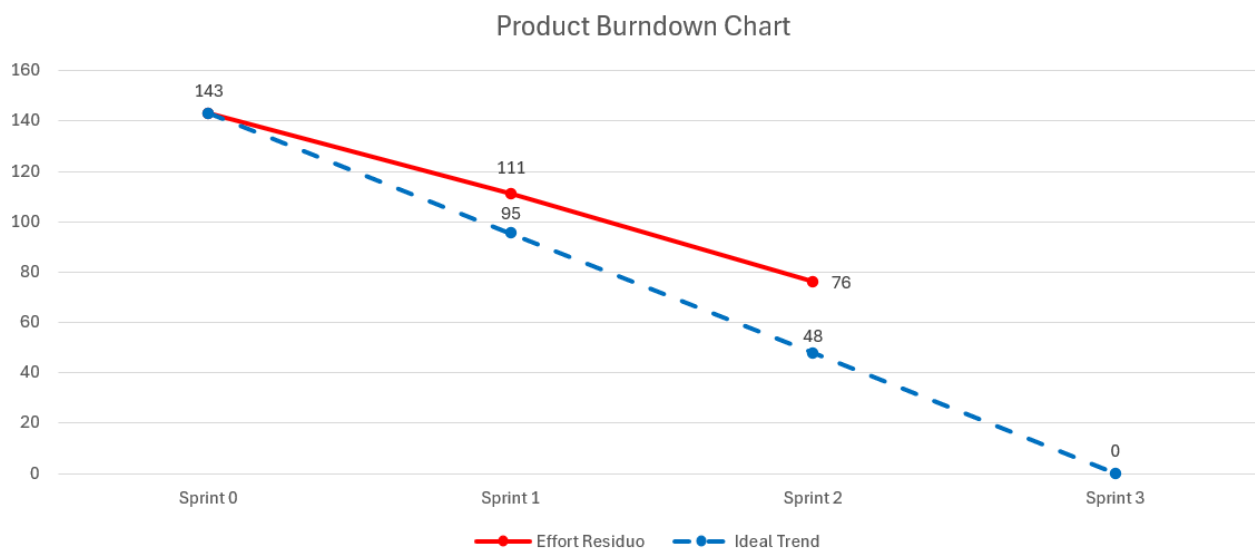


Figura 3.6.1 - Burndown chart (Second Sprint Update)

Il grafico mostra l'andamento del lavoro rimanente (espresso in story points) sprint-by-sprint. La linea tratteggiata blu rappresenta la tendenza ideale di completamento del progetto, mentre la linea rossa mostra l'andamento effettivo.

Al termine della seconda sprint, il lavoro residuo è di 76 SP. Da questo dato, si può evincere che il team ha leggermente sottoperformato rispetto alla previsione ideale (fissa a 48 SP). Tuttavia, la velocity stimata (35 SP) è stata rispettata, mantenendo un ritmo di lavoro sostenibile.

Questo dato è particolarmente significativo: la velocity raggiunta può essere utilizzata come base di riferimento per la pianificazione della prossima sprint.

## 3.7 Second Sprint - Review

### 3.7.1 Project velocity measured:

Durante la fase di Planning della prima Sprint era stata stimata una velocity pari a 35 SP. Alla fine della sprint tale stima risulta essere ben bilanciata, in quanto il team è riuscito a “bruciare” tutti gli SP previsti e di conseguenza a completare tutte le storie presenti nello Sprint Backlog.

La suddivisione del lavoro è stata equa, infatti, rispetto ai task previsti, i componenti della squadra hanno adempiuto ad una media di 9 story points, cadauno.

Nome Membro	Story points
Alessio Bottiglieri	9
Ciro Cutolo	9
Gaetano Frasca	8
Gennaro D'Ambrosio	9

Tabella 29: Suddivisione SP tra i membri del team - 2nd Sprint

*Nota: la suddivisione degli story points è stata effettuata rispetto ai task e alla loro importanza nel contesto della storia da cui sono stati determinati.*

### 3.7.2 Problemi riscontrati durante l'implementazione

Come avvenuto per la Prima Sprint, anche la seconda è stata completata senza accumulare *Technical Debt* o lasciare bug irrisolti per l'iterazione successiva. Questo risultato è stato ottenuto grazie a stime equilibrate, a test approfonditi e a un alto livello di controllo sul codice attraverso una continua revisione incrociata. Tuttavia, una funzionalità particolarmente impegnativa da realizzare durante questa sprint, è stato il ridimensionamento dello spazio di lavoro. Questo compito ha richiesto lo sforzo collettivo di tutto il team per risolvere i vari problemi e bug che si sono presentati, soprattutto a causa dell'ulteriore complessità di garantire che la griglia si adattasse dinamicamente alle nuove dimensioni del canvas.

### 3.7.3 Modifiche da apportare al Product Backlog

Durante questo sprint, sono state apportate alcune modifiche al Product Backlog sulla base delle conoscenze acquisite dallo sviluppo in corso. In particolare, è stata rivalutata la complessità della User Story #3, che ha portato a un aumento dei suoi punti storia da 3 a 5. Inoltre, le User Story #21 (attivazione/disattivazione della griglia) e #22 (visualizzazione della griglia) sono risultate sovrapposte, quindi sono state consolidate eliminando la prima e aggiornando la descrizione e i criteri di accettazione della seconda per riflettere entrambe le funzionalità. Queste modifiche riflettono una comprensione più accurata della portata e delle interdipendenze dei compiti e dimostrano la reattività del team al feedback iterativo e al continuo perfezionamento del backlog.

## 3.8 Second Sprint - Retrospective

Durante la Sprint Retrospective, il gruppo ha utilizzato lo "Starfish Diagram", come struttura collaborativa per valutare le pratiche del team e identificare le aree di miglioramento continuo.

Durante la Seconda Sprint, il team ha fatto consapevolmente riferimento al diagramma realizzato durante l'interazione precedente, che è stato utilizzato come promemoria delle migliori pratiche concordate e delle insidie comuni da evitare. Questo riferimento visivo si è rivelato particolarmente efficace nel promuovere un comportamento coerente ed efficace del team, e nel mantenere un elevato standard di collaborazione, soprattutto nell'area della comunicazione, che era stata precedentemente identificata come un'area chiave di crescita.

Il diagramma seguente riflette i progressi del team in diverse aree e conferma che molte delle lezioni dello sprint precedente sono state interiorizzate con successo.

### 3.8.1 Starfish Diagram

#### 3.8.1.1 Stop (things to stop doing):

- Applicare pattern, senza seguire fedelmente la teoria associata.

#### 3.8.1.2 Less of:

- Redigere troppa documentazione.

#### 3.8.1.3 Keep doing:

- Suddivisione equa dei task tra i membri del team.
- Rispetto della velocity prevista e delle stime iniziali.
- Testing incrociato del sistema e delle sue funzionalità.
- Scrivere commenti e documentazione progressiva, durante lo sviluppo.

#### 3.8.1.4 More of (more things to do):

- Rispettare gli orari e l'organizzazione riguardo i daily meetings.
- Comunicazione efficace e periodica.

#### 3.8.1.5 Start (things to start doing):

- Segnalare bug e difficoltà riscontrate, anche se risolte.

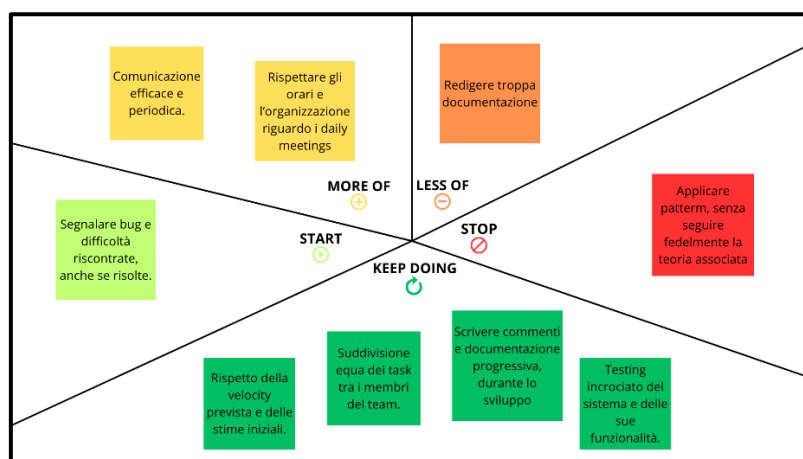


Figura 2: Starfish Diagram - Second Sprint