

Università degli Studi di Salerno

---

CORSO DI LAUREA MAGISTRALE: INGEGNERIA  
INFORMATICA



Software Architecture And Design  
Project:  
DrawSnap, Java Drawing Tool

Group Number 7:  
Anzivino Giuseppe Fabrizio  
Calabrese Raffaele  
Cirillo Francesco Pio  
Fasolino Alessandra

---

Anno Accademico 2024/2025

# Indice

<b>1. Pre-Game.....</b>	<b>2</b>
1.1. Environment Configuration.....	2
Gestione del Tempo.....	2
Linguaggio di Programmazione.....	2
Repository Github/Google Docs.....	2
Ambiente di sviluppo.....	2
Trello.....	2
1.2. Initial Product Backlog.....	3
User Stories/Acceptance Criteria.....	3
Definition of Done (DoD).....	9
1.3. Software Architecture.....	9
Design Architetturale.....	10
Considerazioni sui Design Pattern.....	10
Mock-up.....	11
1.4. 1st Sprint Planning.....	12
Planning Poker.....	12
Estimated Velocity.....	12
Burndown Chart.....	13
Sprint Goal.....	13

# 1. Pre-Game

## 1.1. Environment Configuration

Durante la fase di Pre-Game è stato svolto un primo dibattito sul come gestire in maniera efficiente e organizzata la realizzazione di un applicativo riguardante il disegno geometrico (chiamato dal gruppo DrawSnap, Java Drawing Tool). L'oggetto del dibattito ha riguardato principalmente l'organizzazione del tempo e degli incontri, la configurazione dei diversi ambienti e la scelta di quali strumenti e quale linguaggio utilizzare.

### Gestione del Tempo

Per ottimizzare la gestione del tempo è stato deciso di pianificare incontri periodici e sono state previste come modalità sia la partecipazione in presenza che online. In modo da conciliare le diverse esigenze dei componenti del gruppo è stato deciso di alternare le due modalità, scegliendo di volta in volta quella più adatta.

### Linguaggio di Programmazione

L'applicativo verrà realizzato in linguaggio Java e per impedire incompatibilità e/o conflitti durante la condivisione del codice (es. operazione di push), tutti i componenti del gruppo si sono occupati di installare la stessa versione, ossia JSE 23.0. Insieme a quest'ultimo, sono stati scaricati anche l'ultima versione del JDK riguardante JavaFX e il relativo Scene Builder. Sono state anche svolte delle prove per controllare che il framework di JUnit funzioni correttamente.

### Repository Github/Google Docs

In seguito è stata creata la repository sulla piattaforma Github e sono stati invitati come collaboratori i diversi componenti del gruppo e il Professore in modo da garantire una buona condivisione del codice e del materiale riguardante la progettazione dell'applicativo. Per gestire in maniera efficiente la documentazione il gruppo ha adottato Google Docs in modo da permettere ai diversi componenti di lavorarci contemporaneamente. Di seguito viene riportato il link riguardante il repository Github:

<https://github.com/francescopiocirillo/draw-snap-drawing-tool>

### Ambiente di sviluppo

La scelta dell'ambiente di sviluppo per l'implementazione dell'applicativo è ricaduta su IntelliJ IDEA, particolarmente adatto allo sviluppo in Java. Inoltre, tale IDE risulta ben integrato con sistemi come Git e Github, fondamentali per il lavoro in team, e gestisce in maniera autonoma le dipendenze con i progetti facilitandone la configurazione.

### Trello

Infine, per garantire una gestione strutturata delle diverse attività tramite metodologia Scrum è stata realizzata una Bacheca su Trello, ossia una piattaforma di gestione progetti con la quale il gruppo può monitorare lo stato di avanzamento in modo visuale e intuitivo. Dopo aver creato la bacheca, sono state anche allestite delle liste utili per il corretto utilizzo della metodologia Scrum e per un buon avanzamento del progetto. Tra queste liste distinguiamo:

- Product Backlog
- Sprint Backlog
- Task-to-do

- In Progress
- To verify
- Done

Di seguito viene riportato il link per accedere alla Board di Trello usata per la progettazione dell'applicativo:

<https://trello.com/b/dB0QntH8/draw-snap>

## 1.2. Initial Product Backlog

Dopo essersi preparati sufficientemente per la progettazione dell'applicativo, la prima attività svolta durante la fase di Pre-Game ha previsto la realizzazione di una prima versione del Product Backlog, ossia l'elenco strutturato e ordinato delle diverse funzionalità che il sistema deve fornire.

### User Stories/Acceptance Criteria

Per rendere facilmente comprensibili i bisogni e le richieste degli utenti, le diverse funzionalità sono state formulate attraverso una raccolta di User Stories. Ognuna di esse è stata poi arricchita da un Acceptance Criteria che specifica le condizioni minime che devono essere soddisfatte per considerare la funzionalità completa. La formulazione delle User Stories e degli Acceptance Criteria è stata resa più omogenea e ben strutturata facendo uso di formati standard del seguente tipo:

#### User Stories

Come <>  
voglio <>  
in modo <>

#### Acceptance Criteria

Dato che <>  
quando <>  
allora <>

Nella seguente tabella vengono riportate le User Stories prodotte dal team insieme ai relativi Acceptance Criteria. Ogni User Stories presenta un identificativo e fa riferimento ad una specifica User Epic, ossia un insieme di obiettivi o attività che risultano difficili da realizzare in una singola Sprint e che vanno suddivisi in User Stories più piccole e più gestibili. Inoltre le User Epics sono state definite in ordine di priorità, partendo da quella più alta a quella più bassa.

UE1 - Base Operation		
ID	User Story	Acceptance Criteria
US1.1	<b>Come</b> utente <b>voglio</b> essere accolto da una finestra vuota quando apro il programma <b>in modo</b> da iniziare a disegnare.	<b>Dato che</b> dispongo dell'applicazione <b>quando</b> la apro <b>allora</b> sono accolto da un foglio di disegno vuoto e una barra degli strumenti.
US1.2	<b>Come</b> utente <b>voglio</b> poter creare un disegno composto da una o più forme geometriche <b>in modo</b> da poter disegnare.	<b>Dato che</b> ho l'applicazione aperta <b>quando</b> voglio creare un disegno <b>allora</b> dispongo nella barra degli strumenti di diverse forme geometriche che posso usare per comporlo.

US1.3	<b>Come</b> utente <b>voglio</b> poter scegliere una forma geometrica da aggiungere <b>in modo</b> da poter disegnare come voglio.	<b>Dato che</b> ho l'applicazione aperta <b>quando</b> clicco sull'icona di una figura nella barra degli strumenti <b>allora</b> seleziono la figura per inserirla.
US1.4	<b>Come</b> utente <b>voglio</b> poter selezionare una posizione usando il mouse <b>in modo</b> da aggiungere la figura nel foglio di disegno.	<b>Dato che</b> ho selezionato una figura per mezzo della barra degli strumenti <b>quando</b> clicco con il mouse su un punto del Canvas <b>allora</b> la figura viene posizionata in quel punto.
US1.5	<b>Come</b> utente <b>voglio</b> poter aggiungere segmenti di linea al disegno <b>in modo</b> da poterle disegnare.	<b>Dato che</b> ho aperto l'applicazione <b>quando</b> clicco sull'icona dei segmenti di linea nella barra degli strumenti <b>allora</b> posso inserire un segmento di linea cliccando su un punto del foglio di disegno.
US1.6	<b>Come</b> utente <b>voglio</b> poter aggiungere rettangoli <b>in modo</b> da poterli disegnare.	<b>Dato che</b> ho aperto l'applicazione <b>quando</b> clicco sull'icona del rettangolo nella barra degli strumenti <b>allora</b> posso inserire un rettangolo cliccando su un punto del foglio di disegno.
US1.7	<b>Come</b> utente <b>voglio</b> poter aggiungere ellissi <b>in modo</b> da poterle disegnare.	<b>Dato che</b> ho aperto l'applicazione <b>quando</b> clicco sull'icona dell'ellissi nella barra degli strumenti <b>allora</b> posso inserire un'ellisse cliccando su un punto del foglio di disegno.
US1.8	<b>Come</b> utente <b>voglio</b> poter scegliere il colore della forma da aggiungere tra almeno 8 possibili colori <b>in modo</b> da poter personalizzare il disegno.	<b>Dato che</b> ho aperto l'applicazione <b>quando</b> clicco la forma nella barra degli strumenti <b>allora</b> posso scegliere il colore della stessa tra almeno 8 colori differenti prima di inserirla.
US1.9	<b>Come</b> utente se aggiungo una forma che definisce un'area chiusa <b>voglio</b> poter scegliere il colore al suo interno <b>in modo</b> da rendere il disegno più bello.	<b>Dato che</b> ho aperto l'applicazione <b>quando</b> clicco nella barra degli strumenti una figura che definisce un'area chiusa <b>allora</b> posso scegliere il colore interno alla stessa.

US1.10	Come utente <b>voglio</b> poter scegliere la dimensione della forma da aggiungere <b>in modo</b> da poter personalizzare il disegno.	<b>Dato che</b> ho aperto l'applicazione <b>quando</b> clicco il segmento nella barra degli strumenti <b>allora</b> posso scegliere la lunghezza e l'angolo di inclinazione del segmento.
		<b>Dato che</b> ho aperto l'applicazione <b>quando</b> clicco la forma rettangolo o ellisse nella barra degli strumenti <b>allora</b> posso scegliere la dimensione della stessa prima di inserirla.
US1.11	Come utente se aggiungo una forma che definisce un'area chiusa <b>voglio</b> poter scegliere il colore al suo interno <b>in modo</b> da rendere il disegno più bello.	<b>Dato che</b> voglio salvare il lavoro svolto <b>quando</b> clicco sull'icona di salvataggio sulla barra degli strumenti <b>allora</b> si apre la finestra di dialogo per il salvataggio del file.
		<b>Dato che</b> voglio annullare l'operazione di salvataggio <b>quando</b> è aperta la finestra di dialogo per il salvataggio del file <b>allora</b> premo il pulsante di "Annulla" e il File non viene salvato.
		<b>Dato che</b> voglio salvare il disegno in una specifica cartella <b>quando</b> è aperta la finestra di dialogo per il salvataggio del file <b>allora</b> scelgo il percorso dove salvarlo.
US1.12	Come utente <b>voglio</b> poter caricare un disegno da un file <b>in modo</b> da non perdere il lavoro svolto.	<b>Dato che</b> voglio riaprire un file salvato in locale <b>quando</b> clicco sull'icona di caricamento sulla barra degli strumenti <b>allora</b> si apre la finestra di dialogo per la scelta del file.
		<b>Dato che</b> voglio annullare l'operazione di caricamento <b>quando</b> è aperta la finestra di dialogo per il caricamento del file <b>allora</b> premo il pulsante di "Annulla" e il File non viene caricato.
		<b>Dato che</b> voglio caricare il disegno da una specifica cartella <b>quando</b> è aperta la finestra di dialogo per il caricamento del file <b>allora</b> scelgo il percorso da dove scaricarlo.

## UE2 - Shape Editing

ID	User Story	Acceptance Criteria
US2.1	<b>Come</b> utente <b>voglio</b> poter selezionare la forma del disegno con un'operazione del mouse <b>in modo</b> da svolgere altre operazioni su di essa.	<b>Dato che</b> ho aggiunto delle forme sul foglio di disegno <b>quando</b> clicco sulla forma <b>allora</b> la seleziono e risulta possibile fare delle operazioni su di essa.
US2.2	<b>Come</b> utente <b>voglio</b> poter eseguire l'operazione "Elimina" <b>in modo</b> da eliminare la forma selezionata.	<b>Dato che</b> è presente una forma all'interno del foglio di disegno, <b>quando</b> è selezionata e clicco il pulsante "Elimina" che compare nel menù contestuale <b>allora</b> la figura selezionata verrà eliminata.
US2.3	<b>Come</b> utente <b>voglio</b> poter spostare la forma selezionata <b>in modo</b> da metterla in una posizione diversa.	<b>Dato che</b> è presente una forma all'interno del foglio di disegno, <b>quando</b> è selezionata <b>allora</b> posso spostarla in una posizione diversa.
US2.4	<b>Come</b> utente <b>voglio</b> poter cambiare il colore interno della forma <b>in modo</b> da colorarla in modo diverso.	<b>Dato che</b> è presente una forma all'interno del foglio di disegno, <b>quando</b> è selezionata <b>allora</b> posso cambiarne il colore interno attraverso il menù contestuale.
US2.5	<b>Come</b> utente <b>voglio</b> poter cambiare il colore del bordo della forma <b>in modo</b> da colorarlo in modo diverso.	<b>Dato che</b> è presente una forma all'interno del foglio di disegno, <b>quando</b> è selezionata <b>allora</b> posso cambiare il colore del bordo attraverso il menù contestuale.
US2.6	<b>Come</b> utente <b>voglio</b> poter modificare la dimensione della forma <b>in modo</b> da renderla più piccola o più grande.	<b>Dato che</b> è presente una forma all'interno del foglio di disegno, <b>quando</b> è selezionata <b>allora</b> posso cambiarne la dimensione a piacere.
US2.7	<b>Come</b> utente <b>voglio</b> poter eseguire l'operazione di "Taglia" <b>in modo</b> da copiare e poi eliminare la forma selezionata.	<b>Dato che</b> è presente una forma all'interno del foglio di disegno, <b>quando</b> è selezionata e clicco il pulsante "Taglia" che compare nel menù contestuale <b>allora</b> la figura selezionata verrà copiata e poi eliminata.
US2.8	<b>Come</b> utente <b>voglio</b> poter eseguire l'operazione "Copia" <b>in modo</b> da copiare la forma selezionata.	<b>Dato che</b> è presente una forma all'interno del foglio di disegno, <b>quando</b> è selezionata e clicco il pulsante "Copia" che compare nel menù contestuale <b>allora</b> la figura selezionata verrà copiata.

US2.9	<b>Come</b> utente <b>voglio</b> poter eseguire l'operazione di "Incolla" <b>in modo</b> da incollare ciò che è stato copiato precedentemente.	<b>Dato che</b> ho copiato un elemento precedentemente <b>quando</b> seleziono un punto nella finestra col tasto destro e clicco il tasto "Incolla" che compare dal menu contestuale <b>allora</b> verrà incollato sul disegno l'elemento copiato nel punto selezionato.
US2.10	<b>Come</b> utente <b>voglio</b> poter annullare le operazioni un numero illimitato di volte <b>in modo</b> da annullare le modifiche effettuate al disegno.	<b>Dato</b> che ho svolto un numero indefinito di modifiche al disegno, <b>quando</b> clicco il tasto "Annulla modifica" <b>allora</b> verrà cancellata l'operazione precedentemente svolta, fino a quando non sono presenti modifiche effettuate.
		<b>Dato che</b> ho un foglio di disegno <b>quando</b> non è stata effettuata alcuna modifica <b>allora</b> il tasto "Annulla modifica" è disattivato.
US2.11	<b>Come</b> utente <b>voglio</b> poter cambiare i diversi livelli di visibilità delle forme <b>in modo</b> da mettere in primo piano o in secondo piano le diverse figure.	<b>Dato che</b> ho due o più forme nel disegno che si sovrappongono e ne seleziono una, <b>quando</b> seleziono "In primo piano" dal menù contestuale <b>allora</b> vedrò la figura selezionata messa in primo piano rispetto a tutte le altre figure presenti nel disegno.
		<b>Dato che</b> ho due o più forme nel disegno che si sovrappongono e ne seleziono una, <b>quando</b> seleziono "In secondo piano" dal menù contestuale <b>allora</b> vedrò la figura selezionata messa in secondo piano rispetto a tutte le altre figure presenti nel disegno.
UE3 - Drawing Visualization		
ID	User Story	Acceptance Criteria
US3.1	<b>Come</b> utente <b>voglio</b> poter accedere a 4 livelli diversi di zoom dell'interfaccia <b>in modo</b> da poter cambiare lo zoom del disegno.	<b>Dato</b> che ho inserito una figura nel disegno, <b>quando</b> schiaccio il pulsante relativo ad uno dei



		4 livelli di zoom, <b>allora</b> vedrò la finestra del disegno zoomata secondo il livello richiesto.
US3.2	<b>Come</b> utente <b>voglio</b> poter disegnare su una superficie di disegno più grande della dimensione della finestra <b>in modo</b> tale da poter fare un disegno più grande.	<b>Dato</b> che ho l'applicazione aperta <b>quando</b> voglio creare un disegno ampio <b>allora</b> non sono limitato dalle dimensioni della finestra.
US3.3	<b>Come</b> utente <b>voglio</b> poter scorrere la porzione di disegno visualizzata <b>in modo</b> tale da visualizzare diverse parti del disegno.	<b>Dato</b> che sto disegnando su un foglio più ampio della finestra, <b>quando</b> scorro nel documento <b>allora</b> il disegno scorrerà nella direzione scelta senza limiti.
US3.4	<b>Come</b> utente <b>voglio</b> poter attivare o disattivare una griglia <b>in modo</b> tale da aiutarmi a posizionare le forme.	<b>Dato che</b> talvolta ho bisogno di precisione <b>quando</b> clicco il tasto "griglia" <b>allora</b> posso attivare o disattivare una griglia.
US3.5	<b>Come</b> utente <b>voglio</b> poter scegliere la dimensione della griglia <b>in modo</b> tale da aiutarmi secondo diversi livelli di precisione.	<b>Dato che</b> ho bisogno di precisione e ho attivato una griglia <b>quando</b> clicco l'apposito tasto <b>allora</b> il livello di precisione della stessa cambia.

#### UE4 - Other Functionalities

ID	User Story	Acceptance Criteria
US4.1	<b>Come</b> utente <b>voglio</b> poter inserire un poligono arbitrario <b>in modo</b> da avere più libertà.	<b>Dato che</b> ho bisogno di disegnare una figura particolare <b>quando</b> clicco sull'apposita figura nella barra degli strumenti <b>allora</b> voglio poter scegliere arbitrariamente il numero di lati e le dimensioni della figura.
US4.2	<b>Come</b> utente <b>voglio</b> poter inserire una stringa di testo come forma <b>in modo</b> da poter realizzare composizioni con testo.	<b>Dato che</b> ho bisogno di disegnare del testo <b>quando</b> clicco sull'icona dedicata nella barra degli strumenti <b>allora</b> voglio poter scrivere sul foglio di disegno.
US4.3	<b>Come</b> utente <b>voglio</b> poter ruotare le figure di un angolo a mia scelta <b>in modo</b> da avere più	<b>Dato che</b> ho bisogno di orientare diversamente la figura <b>quando</b> clicco sull'apposito pulsante

	libertà con le composizioni.	dopo aver selezionato la figura <b>allora</b> ho la possibilità di ruotare la figura.
US4.4	<b>Come</b> utente <b>voglio</b> poter specchiare una forma orizzontalmente o verticalmente <b>in modo</b> da poterla modificare ed ottenere effetti visivi particolari.	<b>Dato che</b> ho una figura nel foglio di disegno <b>quando</b> clicco sul pulsante apposito <b>allora</b> posso specchiarla orizzontalmente.
		<b>Dato che</b> ho una figura nel foglio di disegno <b>quando</b> clicco sul pulsante apposito <b>allora</b> posso specchiarla verticalmente.
US4.5	<b>Come</b> utente <b>voglio</b> poter stirare una forma orizzontalmente o verticalmente <b>in modo</b> da poterne alterare l'aspetto.	<b>Dato che</b> ho una figura nel foglio di disegno <b>quando</b> seleziono la figura e clicco il pulsante apposito <b>allora</b> voglio poter stirare in orizzontale o in verticale la figura.

## Definition of Done (DoD)

Per assicurare che ogni User story venga completata in modo esaustivo, il gruppo ha definito una propria Definition of Done (DoD), composta dai seguenti punti:

- Il codice è stato revisionato;
- Il codice è commentato;
- Il codice svolge tutte le funzionalità richieste dalla user story;
- L'implementazione della user story soddisfa tutti i criteri di accettazione;
- Ogni criterio di accettazione ha almeno un caso di test associato;
- Gli unit test sono stati eseguiti e superati;
- I test di integrazione sono stati eseguiti e superati;
- La user story è stata revisionata dal team per assicurarsi che il prodotto realizzato la rispecchia;
- Il deployment della story nell'ambiente di produzione ha release notes.

Questa DoD garantisce che alla fine di ogni sprint possa essere consegnato del software funzionante che risponda alle richieste dettate.

User Stories, Acceptance Criterias e DoD sono state riportate sulla Board Trello per tenere correttamente traccia dello stato di avanzamento del progetto.

## 1.3. Software Architecture

Dopo di che è stato eseguito un brainstorming riguardante il pattern architetturale da utilizzare per garantire una buona progettazione dell'applicativo e buoni parametri di coesione e accoppiamento. Inoltre sono state fatte considerazioni preliminari riguardanti i pattern da mettere in atto durante la

scrittura del codice ed è stato definito un Mockup dell'interfaccia utente tramite un semplice disegno su Draw.io in modo da farsi un'idea su come deve essere l'applicativo.

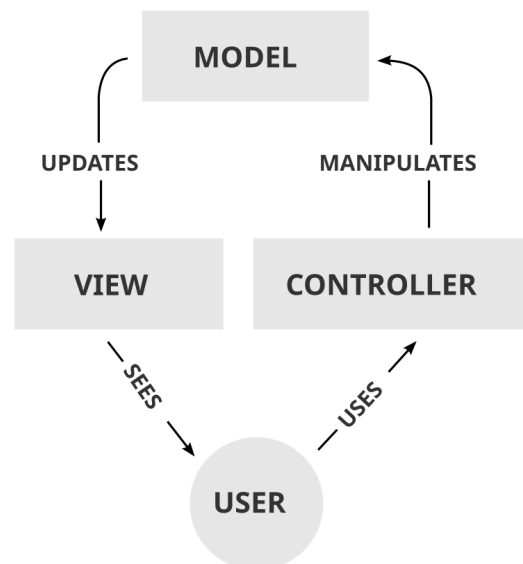
## Design Architetturale

Per lo sviluppo del sistema verrà seguito il pattern architetturale Model-View-Controller (MVC). La scelta di MVC risulta intuitiva ed immediata vista la grande compatibilità con il Workflow relativo all'utilizzo di JavaFXML, che prevede la generazione automatica di una View a partire dalla composizione in Scene Builder e poi in seguito l'integrazione di questa con classi che gestiscono la business logic. Questa architettura permette di suddividere il sistema in elementi separati ed indipendenti, garantendo il principio della separazione delle responsabilità e quindi un basso livello di accoppiamento tra i componenti del sistema. Ciò è fondamentale in quanto garantisce manutenibilità (si può modificare una parte senza intaccare le altre) e di conseguenza una facile testabilità.

Il pattern si compone di 3 principali componenti interconnessi:

- **Model:** rappresenta la logica di business dell'applicazione, contiene infatti i dati e fornisce i metodi per manipolarli. E' indipendente dal Controller e dalla View.
- **View:** si occupa di far visualizzare i dati all'utente attraverso l'interfaccia di cui ne gestisce i componenti. Essa comunica con il Controller per ricevere istruzioni su come presentare i dati e per inviare l'input dell'utente. Nel progetto, sarà costituita dalla schermata utente creata tramite JavaFXML, contenente tasti, menù e il foglio di disegno.
- **Controller:** comunica con il Model per richiedere dati e innescare cambiamenti basati sulle azioni dell'utente, gestendo la logica di input e aggiornando di conseguenza la View.

La vista architetturale segue quindi il seguente schema, utile a comprenderne la struttura ad alto livello:



## Considerazioni sui Design Pattern

In merito all'adozione di Design Pattern al fine di agevolare l'implementazione di determinate funzionalità si procede con le seguenti considerazioni:

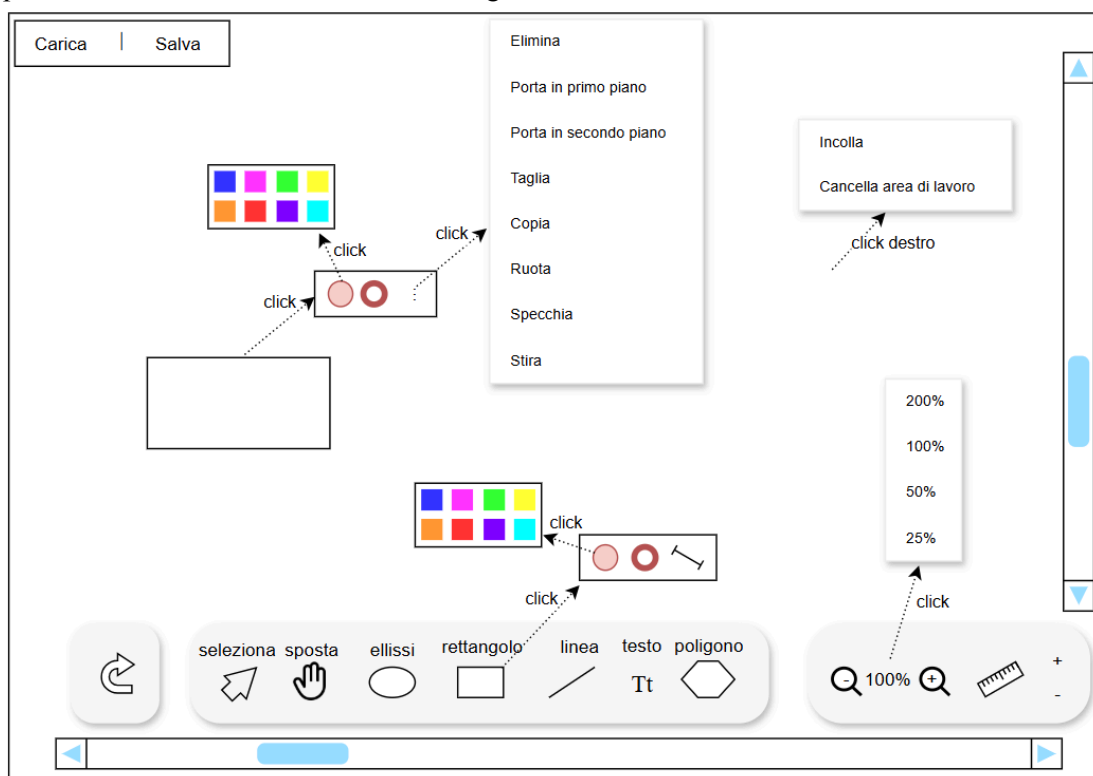
- **Factory Method:** si riconosce il potenziale beneficio dell'utilizzo del Pattern Factory Method al fine di agevolare l'implementazione del processo di creazione dei diversi tipi di forme. In

questo modo la logica di creazione dei diversi tipi di forme può essere indipendente dalla logica di utilizzo delle stesse.

- **Decorator**: il pattern Decorator risulta ottimo per assegnare dinamicamente proprietà ad oggetti esistenti senza dover modificare le loro classi. Si considera quindi il suo utilizzo per trasformare una figura normale in una figura “selezionata” aggiungendo tutte le proprietà grafiche necessarie senza l’aggiunta di logica condizionale.
- **State**: il pattern State può essere usato per gestire le 3 diverse modalità operative riconosciute nell’app: Sposta, Seleziona, Disegna. Utilizzando il pattern risulta possibile compartimentalizzare le 3 modalità ed evitare logica condizionale in quanto lo stesso oggetto (il puntatore) può assumere 3 comportamenti diversi a seconda dello stato interno.
- **Strategy**: al fine di distinguere tra le diverse Forme che si possono disegnare senza dover lavorare sullo stato del canvas queste sono divise con il pattern Strategy seppur facendo parte dello stesso State (Disegna). In questo modo diversi algoritmi e quindi diversi comportamenti possono essere implementati dalla stessa classe.
- **Memento**: il pattern Memento risulta la scelta più adeguata per l’implementazione della funzione Annulla. Infatti il pattern Memento prevede la creazione per ogni stato che si vuole salvare di un Memento che viene gestito, come anche gli altri precedentemente salvati, da un CareTaker.
- **Composite**: per mezzo del pattern Composite si prevede che sarà possibile implementare in modo agevole la funzione per raggruppare diverse figure. Infatti con il pattern Composite è possibile creare composizione di oggetti che a loro volta possono essere oggetti semplici o composti.

## Mock-up

Il team ha provveduto alla realizzazione di un Mock-up volto ad assicurare una visione coerente e unificata del progetto da realizzare. Il Mock-up permetterà inoltre una parallelizzazione agevolata dei compiti relativi alla modifica dell’interfaccia grafica in Scene Builder.



## 1.4. 1st Sprint Planning

In seguito l'obiettivo è stato quello di pianificare la prima Sprint scegliendo le User Stories da implementare, stimando la velocità e definendo lo Sprint Goal. Prima di procedere a ciò, sono state attribuite ad ogni User Stories degli Story Points che indicano una stima del carico di lavoro necessario a completarle.

### Planning Poker

Seguendo la metodologia Scrum, il gruppo ha discusso degli Story Points tramite la cerimonia del Planning Poker. Ogni User Story nel Product Backlog è stata letta ad alta voce e successivamente discussa dal team. Fatto questo, ogni membro ha espresso allo stesso tempo il valore in Story Points stimato per la Story discussa. I valori da assegnare seguono la sequenza di Fibonacci e, in particolare, sono stati utilizzati i primi 5 numeri della sequenza (1, 2, 3, 5, 8), indicando che le Stories sono state formulate seguendo correttamente le linee guida INVEST.

Solo una volta raggiunta l'unanimità sul valore degli Story Points, questi ultimi sono stati assegnati alle relative Stories. Altrimenti, è avvenuta un'ulteriore votazione dopo aver discusso il proprio punto di vista e le motivazioni dietro alle stime effettuate. Nella seguente tabella sono riportate le diverse User Stories (tramite identificativo) e il relativo valore in Story Points:

User Stories	Story Points
US1.1 - US1.8 - US1.9 - US1.10 - US2.2 - US3.3	1
US1.4 - US1.5 - US1.6 - US1.7 - US3.2 - US3.5	2
US1.2 - US1.3 - US1.11 - US1.12 - US2.1 - US2.4 - US2.5 - US2.7 - US2.8 - US2.9 - US3.1 - US4.2	3
US2.3 - US2.6 - US3.4 - US4.3 - US4.4 - US4.5	5
US2.10 - US2.11 - US4.1	8

### Estimated Velocity

Dopo aver definito gli Story Points per ognuna delle User Story presente nel Product Backlog il gruppo si è prestato a decidere quali e quanti User Stories cercare di svolgere nella prima Sprint. Sono state individuate un totale di 18 User Stories da completare nella prima Sprint. Di seguito è riportata una tabella che le racchiude con i relativi Story Points:

User Stories	Story Points
US1.1 - US1.8 - US1.9 - US1.10 - US2.2	1
US1.4 - US1.5 - US1.6 - US1.7	2
US1.2 - US1.3 - US1.11 - US1.12 - US2.1 - US2.7 - US2.8 - US2.9	3

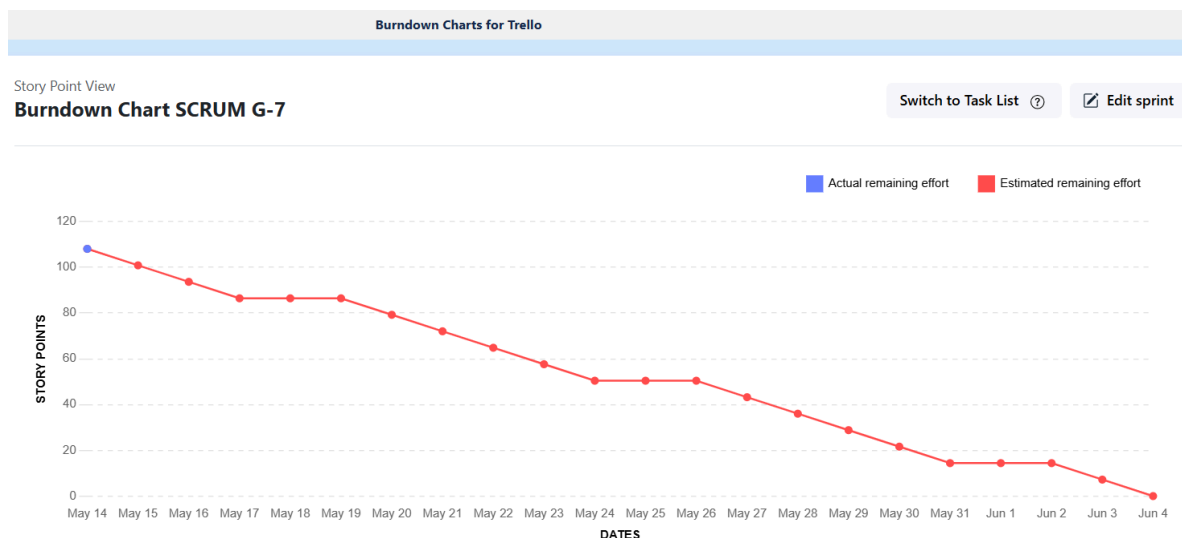
Gli SP riguardanti le User Stories aggiunte alla Sprint sono, poi, stati usati per definire una velocità stimata della Sprint. Questa stima aiuta a comprendere quanto carico di lavoro assegnare al team di sviluppo ad ogni Sprint, e a seconda di quanti SP sono stati soddisfatti andare a diminuire o aumentare il carico. La velocità stimata per la prossima Sprint è calcolata come segue:

$$\sum SP \times N_{US} = (1 \times 5) + (2 \times 4) + (3 \times 8) + (1 \times 5) = 42$$

Il gruppo ha deciso di stimare questa velocità in modo da riuscire ad implementare l'applicativo nel giro di tre Sprint.

## Burndown Chart

Inoltre è stato integrato su Trello anche il Power-Up per la realizzazione del Burndown Chart, uno strumento fondamentale per tracciare l'andamento del lavoro durante la sprint. Il Burndown Chart ci consente di visualizzare in modo chiaro e immediato il lavoro rimanente in relazione al tempo a disposizione, permettendoci di individuare eventuali rallentamenti e migliorare la pianificazione delle prossime sprint. Di seguito è riportato il Burndown Chart prodotto con Trello:



## Sprint Goal

L'obiettivo per la prima Sprint, in linea con le User Stories inserite nello Sprint Backlog, è la creazione di un applicativo JavaFX funzionante che si presenti come un enorme canva con una barra degli strumenti che permetta la selezione della modalità di interazione con il canvas.

Le funzionalità previste sono:

- Inserisci, ossia la possibilità di inserire le figure nell'area di disegno. Le figure disponibili sono:
  - Linea
  - Rettangolo
  - Ellisse
- Seleziona, ossia la possibilità di selezionare le figure all'interno dell'area di disegno. Una volta selezionate è possibile svolgere le seguenti operazioni:
  - Elimina

- Copia
- Taglia

Una volta copiata la figura può anche essere incollata con la funzione Incolla.

- Sposta, ossia la possibilità di spostare le figure all'interno dell'area di disegno.
- Salva, ossia la possibilità di salvare il disegno su un File.
- Carica, ossia la possibilità di caricare un File con un disegno precedentemente salvato.