

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

1. Pre-Game.....	3
1.1. Environment Configuration.....	3
Gestione del Tempo.....	3
Linguaggio di Programmazione.....	3
Repository Github/Google Docs.....	3
Ambiente di sviluppo.....	3
Trello.....	3
1.2. Initial Product Backlog.....	4
User Stories/Acceptance Criteria.....	4
Definition of Done (DoD).....	0
1.3. Software Architecture.....	0
Design Architetturale.....	0
Considerazioni sui Design Pattern.....	0
Mock-up.....	0
1.4. 1st Sprint Planning.....	0
Planning Poker.....	0
Estimated Velocity.....	0
Burndown Chart.....	0
Sprint Goal.....	0
2. 1st Sprint Release.....	0
2.1. Sprint Backlog.....	0
Class Diagram.....	0
Task Assignment.....	0
2.2. Updated Product Backlog.....	0
2.3. Sprint Review.....	0
User Stories complete alla fine della Sprint:.....	0
Implementation/Testing Issues.....	0
2.4. Sprint Retrospective.....	0
2.5. 2nd Sprint Planning.....	0
Estimated Velocity.....	0
Burndown Chart.....	0
Sprint Goal.....	0
3. 2nd Sprint Release.....	0
3.1. Sprint Backlog.....	0
Class Diagram.....	0
Task Assignment.....	0
3.2. Sprint Review.....	0
User Stories complete alla fine della Sprint:.....	0
Implementation/Testing Issues.....	0
3.3. Sprint Retrospective.....	0
3.4. 3rd Sprint Planning.....	0

Estimated Velocity.....	0
Burndown Chart.....	0
Extended Burndown Chart.....	0
Sprint Goal.....	0
4. 3rd Sprint Release.....	0
4.1. Sprint Backlog.....	0
Class Diagram.....	0
Factory Pattern.....	0
Builder Pattern.....	0
Decorator Pattern.....	0
State Pattern.....	0
Command Pattern.....	0
Memento Pattern.....	0
Composite Pattern.....	0
Task Assignment.....	0
4.2. Sprint Review.....	0
User Stories complete alla fine della Sprint:.....	0
Implementation/Testing Issues.....	0
4.3. Sprint Retrospective.....	0
4.4. Project Release.....	0
Burndown Chart.....	0
Extended Burndown Chart.....	0
Considerazioni Finali.....	0

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
- Task Done
- User Stories 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	Come utente voglio essere accolto da una finestra vuota quando apro il programma in modo da iniziare a disegnare.	Dato che dispongo dell'applicazione quando la apro allora sono accolto da un foglio di disegno vuoto e una barra degli strumenti.
US1.2	Come utente voglio poter creare un disegno composto da una o più forme geometriche in modo da poter disegnare.	Dato che ho l'applicazione aperta quando voglio creare un disegno allora dispongo nella barra degli strumenti di diverse forme geometriche che posso usare per comporlo.

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

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

UE2 - Shape Editing

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

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

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

UE4 - Other Functionalities

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

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

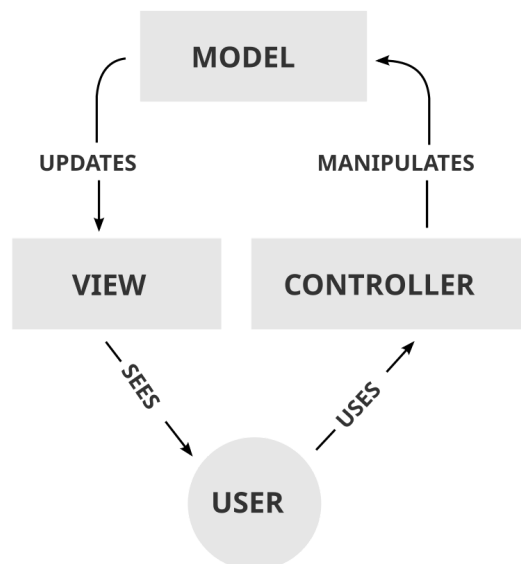
Design Architetturale

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:

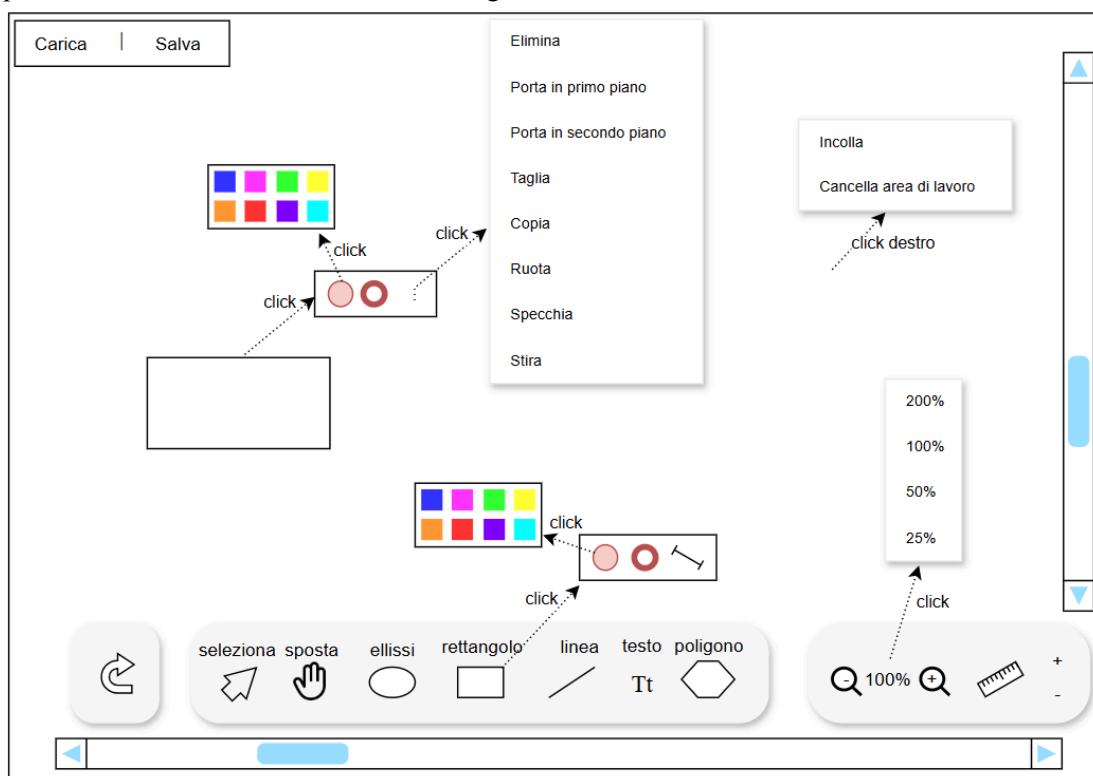
- **Factory Method:** si riconosce il potenziale beneficio dell'utilizzo del Pattern Factory Method al fine di agevolare l'implementazione del processo di creazione dei diversi tipi di forme. In

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

US2.3	5
-------	---

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.

Seguendo il pattern MVC, DrawingAppController è la principale classe di controllo: coordina il comportamento dell'applicazione, contenendo lo stato attuale del canvas (DrawingContext) e gestisce i metodi per cambiare la modalità di disegno e i comandi selezionati. Nel frattempo la classe DrawSnapModel si occupa di gestire le varie informazioni importanti, come la lista di forme presenti nel foglio di disegno, e fornisce metodi per la loro manipolazione.

Si possono notare nel dettaglio i design pattern adottati:

- DrawingContext e DrawingState implementano lo State Pattern, che permette al canvas di comportarsi in modo diverso in base a quale dei diversi stati è selezionato (DrawState, SelectState, MoveCanvasState).
- Per eseguire i comandi come Taglia, Copia, Incolla..., è stato adoperato il Command Pattern tramite la classe CommandInvoker e le implementazioni dell'interfaccia Command.
- Per creare gli oggetti relativi ai vari rettangoli, ellissi e linee, è stato seguito il Factory Pattern, centralizzando e standardizzando la logica di creazione.
- In modo tale da aggiungere alle forme esistenti decorazioni come l'indicazione di selezione, senza modificarne la struttura, è stato infine usato il Decorator Pattern.

Task Assignment

Dopo aver abbozzato il class diagram il gruppo si è occupato di dividere le diverse User Story selezionate per la prima Sprint in diversi task. Quest'ultime sono state aggiunte alla lista Task-to-do sulla piattaforma Trello e assegnate ai diversi componenti del team. Nella seguente tabella è riportato in maniera esaustiva e compatta le assegnazioni effettuate e se esse sono state completate o meno.

Task State	Members			
	Anzivino Giuseppe	Calabrese Raffaele	Cirillo Francesco Pio	Fasolino Alessandra
Assigned and completed tasks	T1.1.1 - T1.1.2 - T1.1.3 - T1.1.4 - T1.2.1 - T1.2.2 - T1.2.3 - T1.2.4 - T1.12.1 - T1.12.2 - T1.12.3 - T1.12.4 - - T1.12.5 - T1.12.6 - T1.12.7 - - T2.7.1 - T2.7.2 - T2.7.3 - T2.7.4 - T2.7.5	T1.4.1 - T1.4.2 - T1.4.3 - T1.8.1 - T1.8.2 - T1.8.3 - T1.8.4 - T1.8.5 - T1.9.1 - T1.9.2 - T1.9.3 - T1.9.4 - T1.10.1 - T1.10.2 - - T1.10.3 - T2.8.1 - - T2.8.2 - T2.8.3 - T2.8.4 - T2.8.5 - T2.9.1 - T2.9.2 - T2.9.3 - T2.9.4 - T2.9.5	T1.3.1 - T1.3.2 - T1.3.3 - T1.11.1 - T1.11.2 - T1.11.3 - - T1.11.4 - T1.11.5 - T1.11.6 - - T1.11.7 - T2.3.1 - - T2.3.2 - T2.3.3	T1.5.1 - T1.5.2 - T1.5.3 - T1.5.4 - T1.5.5 - T1.6.1 - T1.6.2 - T1.6.3 - T1.6.4 - T1.6.5 - T1.7.1 - T1.7.2 - T1.7.3 - T1.7.4 - T1.7.5 - T2.1.1 - T2.1.2 - T2.1.3 - T2.1.4 - T2.1.5 - T2.2.1 - T2.2.2 - T2.2.3 - T2.2.4 - T2.2.5
Tasks assigned but not completed	×	×	×	×
Tasks not assigned to any team member	×	×	×	×

Nella tabella sono state riportati solo gli identificativi delle diverse task, ma sulla piattaforma Trello ognuna di essa è accompagnata da un nominativo che descrive quale sarà l'obiettivo dell'attività da svolgere. Dividendo le User Story in task abbiamo anche deciso di definire dei checkpoint che consistono praticamente in codice compilabile, senza errori e che svolgesse alcune delle funzionalità richieste.

2.2. Updated Product Backlog

Durante la Sprint sono state individuate nuove User Story fornite dal Product Owner e riguardanti una quinta User Epic. Nella seguente tabella sono riportate le nuove User Story aggiunte con le relative descrizioni e i criteri di accettazione definite con lo stesso formato scelto in precedenza:

UE5 - GROUPS AND SHAPES LIBRARIES		
ID	User Story	Acceptance Criteria
US5.2	Come utente voglio poter “raggruppare” le forme selezionate insieme in modo che si comportino come un'unica forma.	Dato che ho bisogno di usare un gruppo di forme come forma unica quando clicco sul tasto raggruppa allora le figure selezionate diventano raggruppate in una figura unica.
US5.3	Come utente voglio poter “separare” un insieme di forme precedentemente raggruppate in modo che si possano fare modifiche sulle singole forme.	Dato che ho bisogno di usare una forma singola che è inserita in un gruppo di forme quando clicco sul gruppo di forme e successivamente sul pulsante apposito nella barra delle modifiche allora devo poter annullare il raggruppamento.
US5.4	Come utente voglio poter “creare una nuova forma” e attribuirgli un nome in modo da poterla aggiungere successivamente al foglio di disegno nello stato in cui era al momento del salvataggio.	Dato che voglio salvare una forma per usarla successivamente nello stato in cui è al momento del salvataggio quando seleziono una forma e poi clicco sul pulsante apposito allora devo poter scegliere il nome della figura e salvarla.
		Dato che voglio usare una forma che ho precedentemente salvato quando seleziono la forma che ho creato nella barra apposita allora posso inserirla nel foglio di disegno.

US5.5	Come utente voglio che al salvataggio di un disegno vengano salvate tutte le forme personalizzate create al suo interno in modo da poterle riutilizzare quando si carica il file nel programma successivamente.	Dato che ho creato nuove figure durante la creazione di un disegno e voglio salvare il disegno in questione quando salvo il file del disegno allora si salvano anche "i comandi di creazione forma" delle nuove figure al suo interno.
US5.6	Come utente voglio poter salvare i "comandi di creazione forma" all'interno del "file librerie forme" in modo da esportare e importare una libreria di figure create da me.	Dato che ho creato nuove figure e voglio salvarle in un file "libreria forme", quando clicco sul pulsante apposito allora si salvano i "file di libreria di forme".
		Dato che ho salvato un "file di libreria di forme" in passato, quando clicco sul pulsante apposito allora posso importare questo file e usare le figure che ho creato in un altro disegno.

Dopo aver definito le diverse User Story è stato previsto anche un piccolo Planning Poker per assegnare gli Story Point alle nuove aggiunte. Come fatto precedentemente questi ultimi sono stati aggiunti su Trello solo dopo l'unanimità del gruppo, o una breve discussione. Nella seguente tabella sono riportate le diverse User Stories (tramite identificativo) e il relativo valore in Story Points:

User Stories	Story Points
US5.3 - US5.5	3
US5.1 - US5.6	5
US5.2 - US5.4	8

2.3. Sprint Review

Dopo aver completato il processo di sviluppo della Sprint e aver passato le diverse User Story nella lista Done è stata svolta una fase di Sprint Review in cui sono state individuate quelle completate ed è stata svolta una piccola discussione su problematiche riscontrate durante l'implementazione e il testing dell'applicativo.

User Stories complete alla fine della Sprint:

Al termine della Sprint sono state completate tutte le User Story aggiunte alla seguente Sprint inizialmente. Nella seguente tabella sono riportate le User Story completate tramite i titoli definiti sulla piattaforma Trello e i relativi Story Point:

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
US2.3	5

Tutti gli Story Points pianificati sono stati consumati, per cui la Project Velocity effettiva è risultata pari a quella stimata.

Implementation/Testing Issues

Nella Sprint Review è stata svolta anche una breve discussione riguardante i problemi riscontrati durante l'implementazione e il testing dell'applicativo. Sostanzialmente le problematiche principali emerse sono state due:

- Durante la fase di implementazione ci sono stati dubbi sul modificare o meno il class diagram per implementare alcune specifiche funzionalità. Questa problematica è stata risolta attraverso un meeting del team in cui sono state trovate soluzioni per non alterare il diagramma iniziale ed allontanarsi dalla logica definita.
- Durante la fase di testing, a causa del fatto che dovessimo testare un applicativo JavaFX, sono state aggiunte delle dipendenze per il testing (es. Mockito, TestFX) e, per via di un' incompatibilità tra esse e la versione scelta di JavaFX, alcuni test non andavano come previsto. Per ovviare a questa problematica sono stati inseriti dei comandi appositi che ignorassero questa compatibilità attraverso la sezione VMOptions all'interno di configurazione di Esecuzione definite appositamente per i Test.

Nei test che prevedono un Mock, è inserita l'opzione:

```
-Dnet.bytebuddy.experimental=true
```

Per i test relativi a DrawStateTestUI, è inserita l'opzione

```
--add-exports=javafx.graphics/com.sun.javafx.application=ALL-UNNAMED.
```

Durante l'esaminazione della release sono stati riscontrati dei bug minori prontamente risolti. Inoltre non ci sono stati problemi con il Product Backlog, quindi non c'è stato bisogno di aggiungere o modificare User Stories.

2.4. Sprint Retrospective

Seguendo lo schema dello Starfish Diagram e concentrandoci sul processo di sviluppo, è stata effettuata la Sprint Retrospective evidenziando come è possibile migliorare il development dell'applicato a partire dalla Sprint successiva. Di seguito è riportata una tabella nella quale vengono riportate le proposte formulate durante queste fase:

Stop Doing	Less Of	Keep Doing	More Of	Start Doing
------------	---------	------------	---------	-------------

Prioritizzare la verifica del codice altrui prima di finire i propri task	Conversazioni su whatsapp per sostituire meeting	Raggiungimento dello Sprint Goal	Pianificazione più dettagliata	Tener traccia dei bug e debito tecnico
	Focus sulla quantità di Codice	Decisioni di gruppo		
		Resilienza agli ostacoli		

2.5. 2nd Sprint Planning

In seguito è stato svolto un planning per la Sprint a venire, focalizzandosi sulla scelta delle User Story, la velocità prevista e lo Sprint Goal.

Estimated Velocity

Come primo obiettivo il gruppo si è prestato a decidere quali e quanti User Story cercare di svolgere nella prossima Sprint. Sono state individuate un totale di 9 User Story da completare nella seconda Sprint. Di seguito è riportata una tabella che le racchiude con i relativi Story Points:

User Stories	Story Points
US3.3	1
US3.2	2
US2.4 - US2.5 - US3.1	3
US2.6 - US3.4	5
US2.10 - US2.11	8

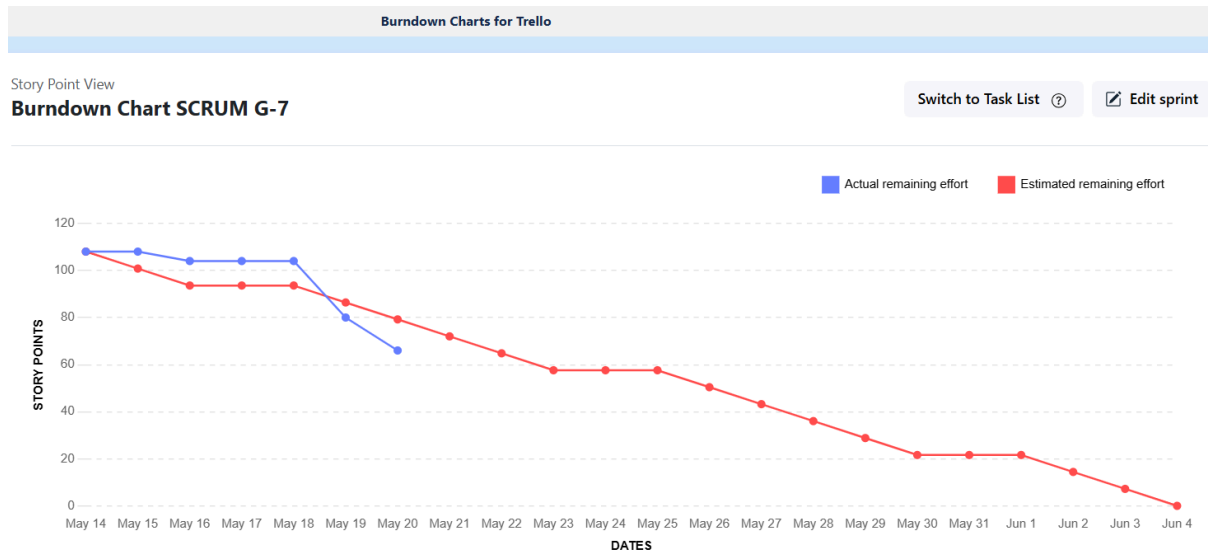
Gli SP riguardanti le User Stories aggiunte alla Sprint sono, poi, stati usati per definire una velocità della Sprint seguendo la stessa modalità usata precedentemente. La velocità stimata per la prossima Sprint è calcolata come segue:

$$\sum SP \times N_{US} = (1 \times 1) + (2 \times 1) + (3 \times 3) + (2 \times 5) + (2 \times 8) = 38$$

Il gruppo ha deciso di rimanere su una velocità simile a quella della precedente Sprint in quanto non ha riscontrato evidenti problematiche riguardanti la consegna e la scadenza fissata.

Burndown Chart

Di seguito è riportato il burndown chart aggiornato dell'intero progetto in seguito alla prima sprint:



Da come è possibile vedere il lavoro sta seguendo un andamento simile a quello previsto, indice che il gruppo sta lavorando al progetto con sinergia e in modo corretto. Anche se verso la fine della Sprint sono state aggiunte delle User Story, queste non sono state prese in considerazione per il Burndown Chart relativo alla fine di questa Sprint in modo da rispecchiare meglio l'andamento corrente del progetto. Naturalmente verrà aggiornato nella prossima.

Sprint Goal

L'obiettivo per la seconda Sprint, in linea con le User Stories inserite nello Sprint Backlog, è la creazione di un applicativo JavaFX funzionante che si presenti come definito precedentemente con l'aggiunta di alcune funzionalità, quali sono:

- la possibilità di modificare le proprietà delle forme presenti nel foglio di disegno (Cambio colore interno, Cambio colore bordo, Cambio dimensione)
- la possibilità di annullare un'operazione precedentemente svolta
- la possibilità di spostare in primo o in secondo piano una forma quando si sovrappone ad altre
- la possibilità di usufruire di zoom e di una griglia per facilitare il posizionamento sul foglio.

3. 2nd Sprint Release

3.1. Sprint Backlog

In seguito alla Sprint Review precedente è emerso che il work flow per l'utilizzo delle funzionalità di Taglia, Copia e Incolla non era soddisfacente per il Product Owner. Quindi è stata aggiunta un'ulteriore User Story per esaudire le richieste avanzate e inserita con la massima priorità nello Sprint Backlog.

UE1 - Shape Editing		
ID	User Story	Acceptance Criteria
US2.12	Come utente voglio che l'interazione per le operazioni di taglia, copia e incolla abbiano lo stesso workflow di funzionamento in modo da rendere più intuitivo lo svolgimento di tali operazioni.	Dato che ho un foglio di disegno con più forme, quando svolgo le operazioni di taglia copia e incolla allora voglio che si trovino nella stessa sezione.

Dopo aver definito la User Story e l'Acceptance Criteria associata è stato previsto anche un piccolo Planning Poker per stimare l'effort necessario al refactoring. Come fatto precedentemente, è stato aggiunti su Trello solo dopo l'unanimità del gruppo.

User Stories	Story Points
US2.12	3

Gli SP riguardanti la User Stories aggiunta alla Sprint sono stati aggiunti alla Estimated Velocity precedentemente calcolata per questa Sprint. La velocità stimata per questa Sprint è calcolata come segue:

$$\sum SP \times N_{US} = (1 \times 1) + (2 \times 1) + (4 \times 3) + (2 \times 5) + (2 \times 8) = 41$$

Il gruppo ha deciso di non eliminare nessuna User Story precedentemente inserita nello Sprint Backlog, per sostituirla con la US2.12, dato che la stima della Velocity attuale è comunque in un intorno della Velocity riscontrata del Team per la Sprint precedente.

Il Class Diagram è stato aggiornato con l'aggiunta del pattern Memento per la realizzazione della funzionalità di “Annulla Modifica” e i Command per le altre funzionalità da implementare. Il Class Diagram aggiornato si presenta come segue:



24

Task State	Members			
	Anzivino Giuseppe	Calabrese Raffaele	Cirillo Francesco Pio	Fasolino Alessandra
Assigned and completed tasks	T2.12.1 - T2.12.2 - T2.12.3 - T2.12.4 - T3.1.1 - T3.1.2 - T3.1.3 - T3.1.4 - T3.4.1 - T3.4.2 - T3.4.3 - T3.4.4	T2.11.1 - T2.11.2 - T2.11.3 - T2.11.4 - T3.3.1 - T3.3.2 - T3.3.3 - T3.3.4	T2.10.1 - T2.10.2 - T2.10.3 - T2.10.4 - T2.10.5 - T3.2.1 - T3.2.2 - T3.2.3 - T3.2.4	T2.4.1 - T2.4.2 - T2.4.3 - T2.4.4 - T2.5.1 - T2.5.2 - T2.5.3 - T2.5.4 - T2.6.1 - T2.6.2 - T2.6.3 - T2.6.4
Tasks assigned but not completed	×	×	×	×
Tasks not assigned to any team member	×	×	×	×

3.2. Sprint Review

Come per la Sprint precedente, dopo aver completato il processo di sviluppo e aver passato le diverse User Story nella lista Done è stata svolta la fase relativa alla Sprint Review in cui sono state individuate quelle completate ed è stata svolta una piccola discussione su eventuali problematiche riscontrate durante l'implementazione e il testing dell'applicativo.

User Stories complete alla fine della Sprint:

Anche al termine di questa Sprint sono state completate tutte le User Story aggiunte inizialmente. Nella seguente tabella sono riportate le User Story completate tramite i titoli definiti sulla piattaforma Trello e i relativi Story Point:

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

Tutti gli Story Points pianificati sono stati consumati, per cui la Project Velocity effettiva è risultata pari a quella stimata. Questi risultati confermano che il team sta lavorando in modo efficace e coordinato, e che la stima iniziale della velocità di progetto è stata ben calibrata rispetto alle attività effettivamente svolte.

Implementation/Testing Issues

Durante la fase di development il gruppo ha riscontrato diverse problematiche e ha deciso di tenere traccia anche dei bug più insidiosi e complicati che potevano portare le User Story a non essere accettate. I bug principali riscontrati sono stati:

- L'utilizzo del tasto di Undo per l'operazione di "Annulla Modifica" non disattiva la barra degli strumenti. I pulsanti presenti al suo interno possono essere utilizzati e portare a comportamenti indesiderati.
- Anche se le dimensioni della forma non sono impostate a 100, quando si usa il ridimensionamento le dimensioni sono messe a 100 invece che alle dimensioni reali della forma.
- Anche se il colore delle forme e dei bordi sono diverse da bianco e nero, le funzionalità di cambio colore li presentano come colore di default invece dei colori reali della forma.
- Inizialmente quando veniva fatto lo zoom, se si disegnava una figura in fondo al foglio canvas, se si aumentava lo zoom, la figura scompariva. Il problema è stato risolto gestendo bene il coordinamento tra il canvas e lo scrollpane.
- L'attivazione e la disattivazione della griglia comportava l'abilitazione del pulsante di Undo.




Inoltre una problematica importante emersa in questa Sprint è stata quella del non funzionamento del Power-Up di Trello precedentemente usato per monitorare il progetto attraverso il Burndown Chart. Per ovviare a questo problema, il gruppo ha deciso di utilizzare Excel per aggiornare l'andamento del Burndown Chart. Sono stati individuati anche bug minori, che sono stati prontamente risolti dai diversi componenti del gruppo e per i quali non c'era bisogno di tenerne traccia. Inoltre a causa di diverse User Story che si sono rivelate più complicate di quello che si pensava è stato deciso di aumentare il valore in Story Points di due User Story a causa di dubbi e difficoltà di coordinazione implementativa con le altre funzionalità implementate. In particolare sono state modificate le seguenti User Story:

US4.2 : 3 → 5

US5.1: 5 → 8

3.3. Sprint Retrospective

Continuando ad utilizzare lo Starfish Diagram è stata effettuata una Sprint Retrospective relativa alla seconda Sprint. Di seguito è riportata una tabella nella quale vengono riportate le proposte formulate durante queste fase:

Stop Doing	Less Of	Keep Doing	More Of	Start Doing
		Raggiungimento dello Sprint Goal	Tener traccia dei bug e debito tecnico	
		Resilienza agli ostacoli		

3.4. 3rd Sprint Planning

In seguito è stato svolto un planning per la Sprint a venire, focalizzandosi sulla scelta delle User Story, la velocità prevista e lo Sprint Goal.

Estimated Velocity

Come primo obiettivo il gruppo si è prestatato a decidere quali e quanti User Story cercare di svolgere nella prossima Sprint. Sono state individuate un totale di 7 User Story da completare nella terza Sprint. Di seguito è riportata una tabella che le racchiude con i relativi Story Points:

User Stories	Story Points
US3.5	2
US4.2 - US4.3 - US4.4 - US4.5	5
US4.1 - US5.1	8

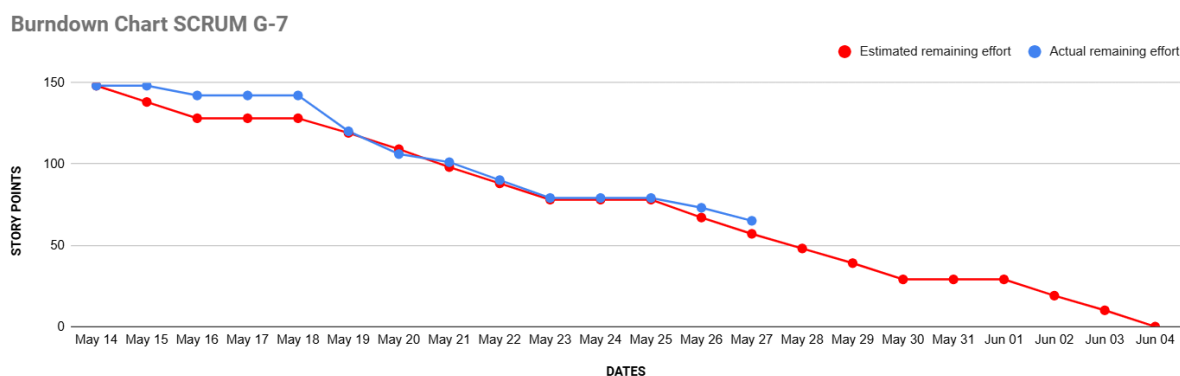
Gli SP riguardanti le User Stories aggiunte alla Sprint sono, poi, stati usati per definire una velocità della Sprint seguendo la stessa modalità usata precedentemente. La velocità stimata per la prossima Sprint è calcolata come segue:

$$\sum SP \times N_{US} = (1 \times 2) + (4 \times 5) + (2 \times 8) = 38$$

Il gruppo ha effettuato la scelta conservativa di rimanere su un intorno inferiore rispetto alla velocità della precedente Sprint, in modo da lasciare eventuale effort per la realizzazione di nuove User Story aggiunte dal Product Owner, come successo in precedenza.

Burndown Chart

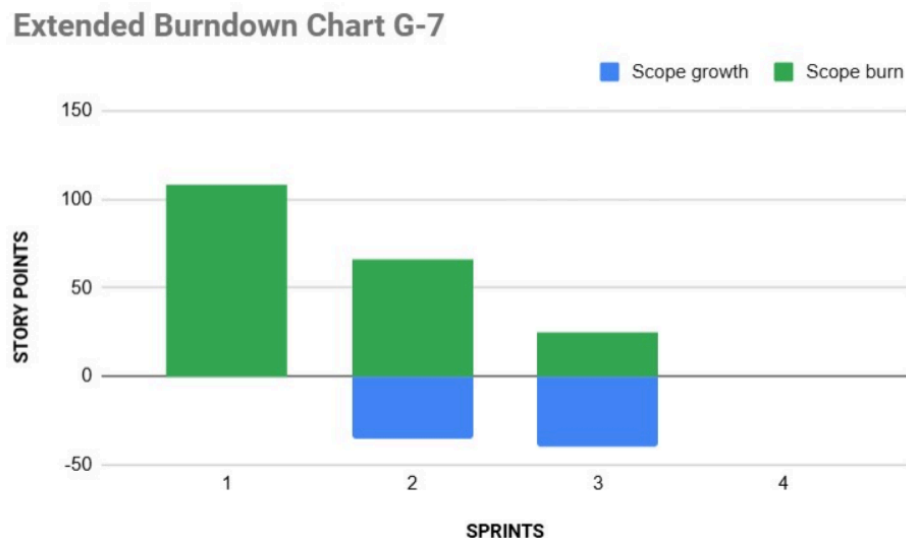
Di seguito è riportato il burndown chart aggiornato dell'intero progetto in seguito alla seconda sprint:



Da come è possibile vedere il lavoro sta seguendo un andamento simile a quello previsto, indice che il gruppo sta lavorando al progetto con sinergia e in modo corretto.

Extended Burndown Chart

Di seguito è riportato l'extended burndown chart aggiornato dell'intero progetto in seguito alla seconda sprint:



Come è possibile riscontrare per mezzo della consultazione dell'Extended Burndown Chart, nella seconda sprint il product owner ha aggiunto nuove user stories che hanno portato alla crescita del numero di story points complessivi del progetto.

Inoltre nella terza sprint è possibile vedere un ulteriore aumento del numero di story points comportato dal fatto che durante la seconda sprint il team ha effettuato delle rivalutazioni della difficoltà di determinati task resa possibile dall'esperienza maturata durante il progetto.

Sprint Goal

L'obiettivo per la terza Sprint, in linea con le User Stories inserite nello Sprint Backlog, è la creazione di un applicativo JavaFX funzionante che si presenti come definito precedentemente con l'aggiunta di alcune funzionalità, quali:

- la possibilità di modificare la dimensione della griglia secondo diversi livelli di precisione
- la possibilità di inserire poligoni arbitrari e stringhe
- la possibilità di applicare modifiche alle figure presenti nel foglio di disegno quali rotazione, specchiamento e stiramento.
- possibilità di selezionare più di una figura contemporaneamente per l'applicazione di modifiche simultanee.

4. 3rd Sprint Release

4.1. Sprint Backlog

Durante lo Sprint, e in particolare nella fase implementativa, il gruppo ha finito per completare prima del previsto le task assegnategli. Per tale motivo sono state aggiunte le User Story relative al raggruppamento e alla separazione delle forme. Gli SP riguardanti le User Stories aggiunte alla Sprint sono stati aggiunti alla Estimated Velocity precedentemente calcolata per questa Sprint. La velocità stimata per questa Sprint è calcolata come segue:

$$\sum SP \times N_{US} = (1 \times 2) + (1 \times 3) + (4 \times 5) + (3 \times 8) = 49$$

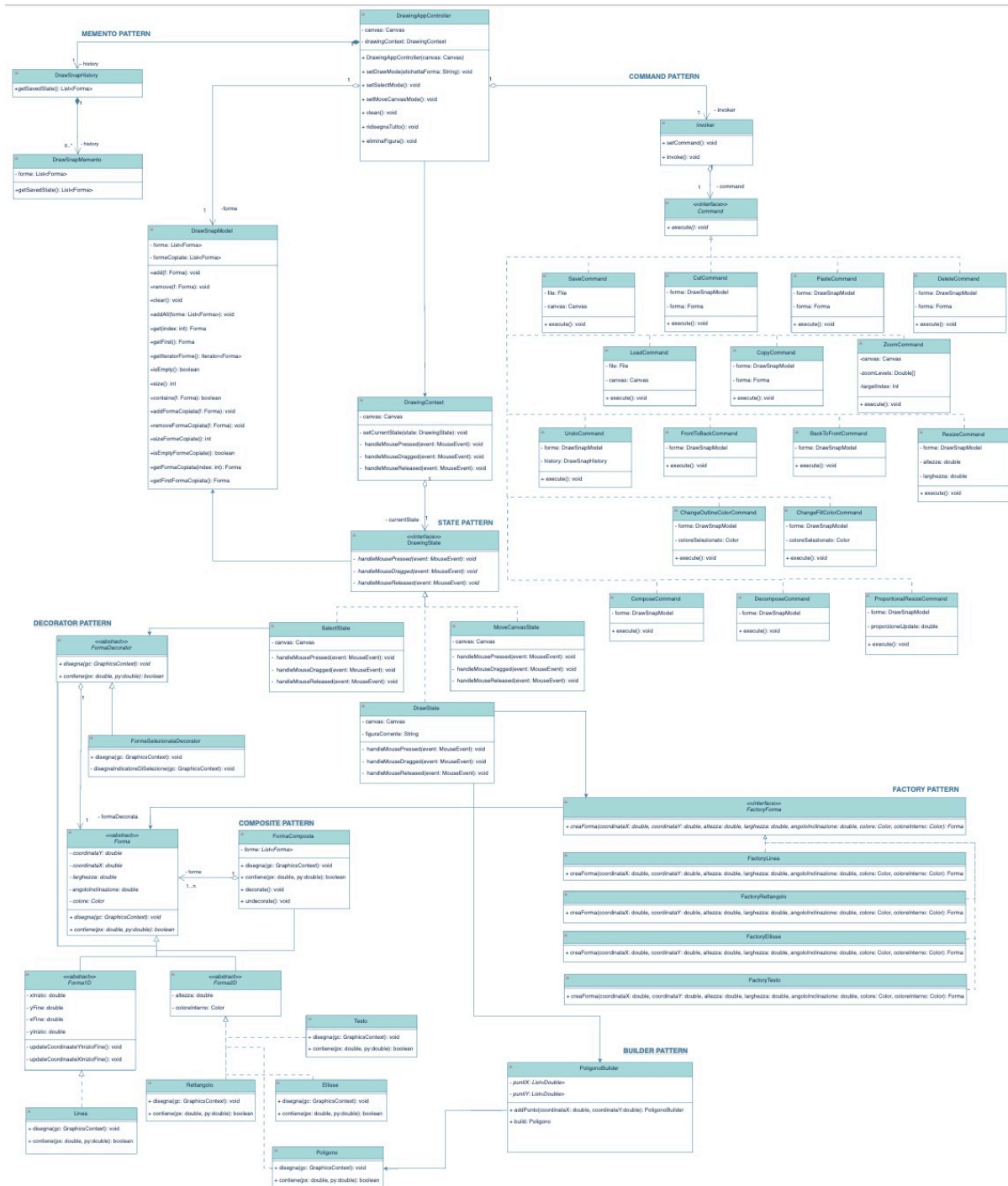
Questo aumento della velocità rispetto alle precedenti Sprint è stato causato principalmente da due motivi:

- L'aumento di Story Points svolto nella Sprint precedente per le US4.2 e 5.1 ha portato ad un aumento di 6 Story Points. Alla fine tale aumento si è rivelato fugace e le funzionalità non hanno comportato grandi complicazioni per la loro implementazione.
- L'aggiunta della User Story relativa al raggruppamento US5.2 ha necessariamente comportato anche l'aggiunta della User Story relativa alla separazione US5.3. Il prodotto finale non sarebbe stato completo senza la possibilità di separare le forme composte dopo averle create. Per permettere lo sforzo di implementare entrambe le User Stories, mantenendo comunque un prodotto finale funzionante e completo, è stato necessario implementare versioni semplificate delle funzioni:
 - ruota forma composta;
 - specchio forma composta;
 - ridimensiona forma composta.

Nonostante ciò il prodotto realizzato è completo, funzionante e aderente a quanto discusso con il Product Owner, e in future release resta possibile implementare versioni più complesse delle funzioni sopra esposte. Prima di procedere con la definizione e l'assegnazione dei Task, il gruppo si è poi focalizzato sull'aggiornamento del diagramma delle classi facendo riferimento a quello che sarebbe dovuto essere il prodotto finale della terza Sprint.

Class Diagram

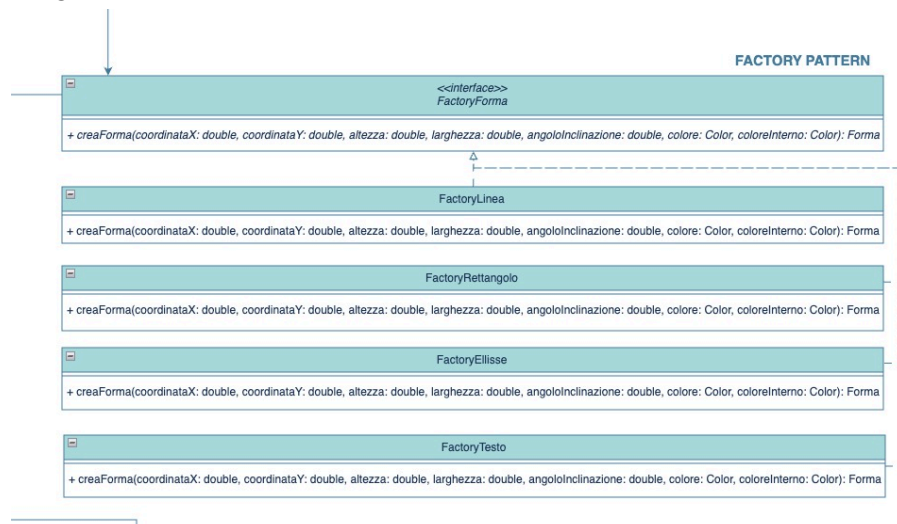
Il class diagram è stato aggiornato con il pattern Builder per la creazione dinamica della classe Poligono, il pattern Composite il raggruppamento delle figure e altri Command per l'implementazione di ulteriori funzionalità. Il Class Diagram aggiornato si presenta come segue:



In particolare possiamo vedere come sono stati implementati i diversi pattern:

Factory Pattern

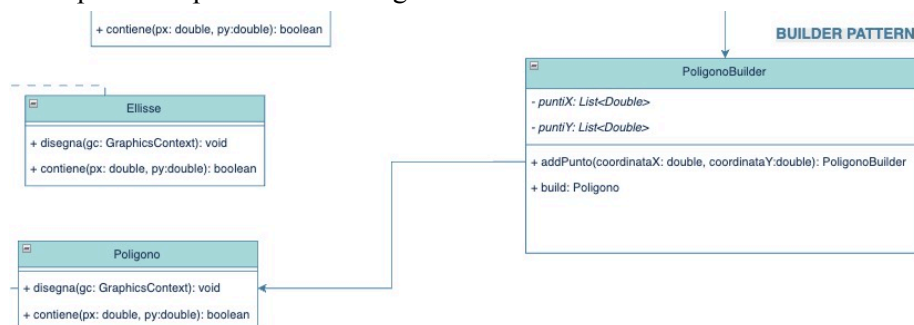
Il Factory Pattern è stato utilizzato per garantire una gestione flessibile e più modulare per la creazione degli oggetti forma permettendo di isolare la logica di istanziazione delle diverse forme geometriche (Forma, Rettangolo, Ellisse, Linea, Testo) all'interno di specifiche classi factory dedicate (FactoryForma, FactoryRettangolo, FactoryEllisse, FactoryLinea, FactoryTesto). In questo modo l'aggiunta di nuove forme o la modifica delle modalità di creazione non richiedono interventi diffusi in tutto il sistema, ma solo l'implementazione o l'adattamento delle classi factory. Il pattern si presenta nel seguente modo:



Esso favorisce anche un'architettura orientata ai principi SOLID, migliorando la separazione delle responsabilità e riducendo le dipendenze tra le componenti.

Builder Pattern

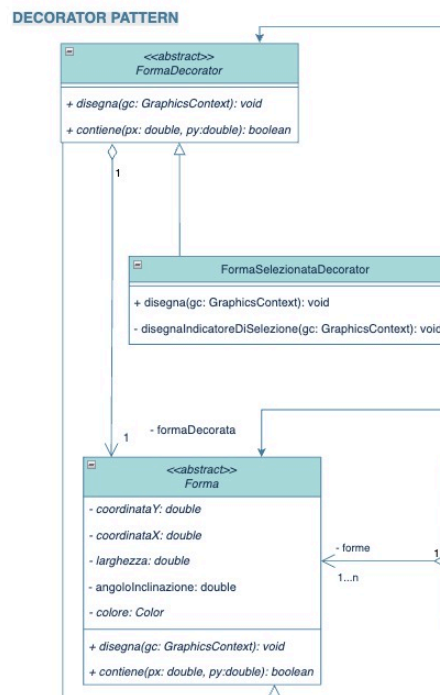
A differenza delle altre forme, la creazione di un poligono richiede la configurazione di molteplici componenti, come la definizione dei vertici e la possibilità di costruire forme complesse in modo graduale e flessibile. Per tale motivo, è stato adottato il Builder pattern nella classe Poligono (PoligonoBuilder). Questo pattern consente di separare la costruzione dell'oggetto dalla sua rappresentazione finale e permette di assemblare passo dopo passo le sue componenti in modo chiaro e controllato. Il pattern si presenta come segue:



Decorator Pattern

Per estendere dinamicamente il comportamento delle forme selezionate senza modificare le loro classi originali, è stato utilizzato il Decorator Pattern nella classe `FormaSelezionataDecorator`. Questo pattern consente di "decorare" un oggetto con funzionalità aggiuntive, come evidenziare proprietà

visive legate allo stato di selezione mantenendo invariata la struttura base dell'oggetto originale. Il pattern si presenta come segue:



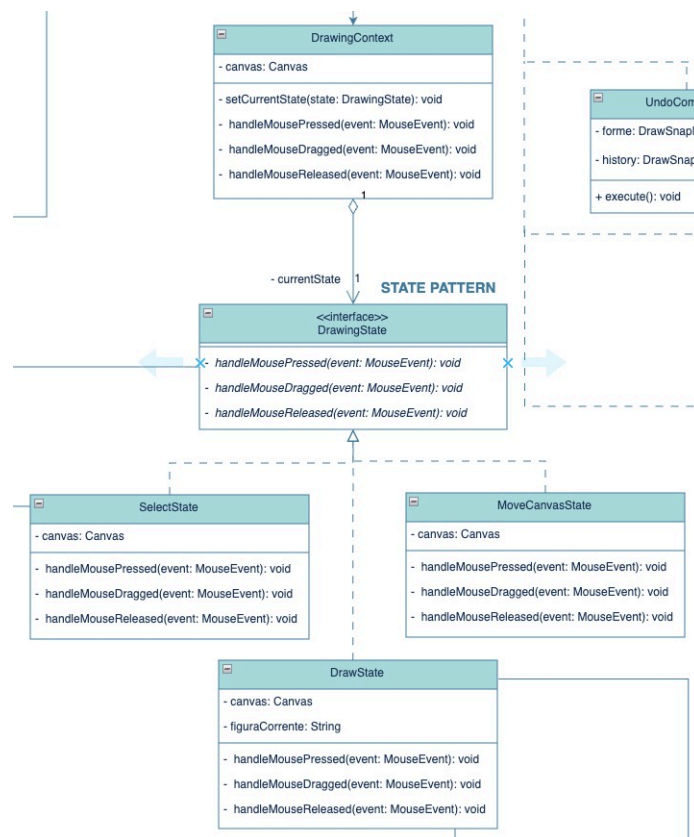
In questo modo è stato possibile modificare i comportamenti delle forme rispettando il principio di responsabilità singola e favorendo la riusabilità del codice.

State Pattern

Per distinguere in maniera chiara e modulare i diversi comportamenti dell'interfaccia grafica durante l'interazione con il canvas, è stato adottato lo State Pattern. Questo pattern consente di rappresentare gli stati distinti del sistema come oggetti separati che incapsulano la logica specifica di ognuno di loro. Gli stati definiti sono:

- DrawState, stato di disegno sul Canvas
- SelectState, stato per la selezione delle figure presenti nel Canvas
- MoveCanvasState, stato per lo spostamento della visuale all'interno del Canvas

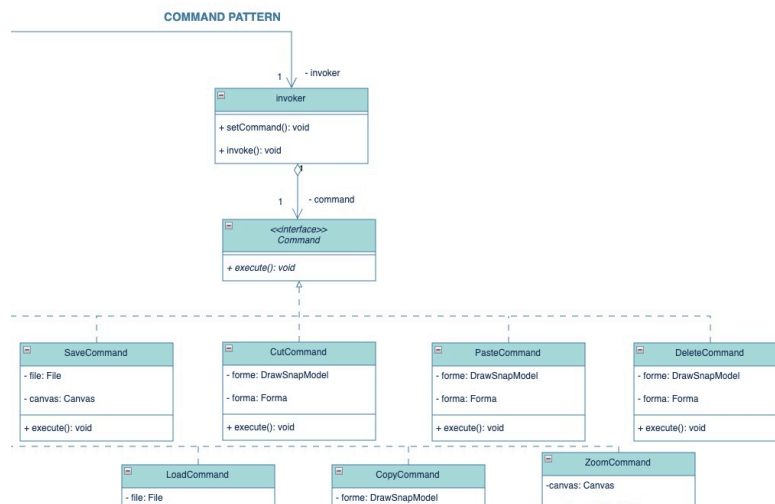
Grazie allo State Pattern, il sistema può cambiare comportamento dinamicamente in base allo stato corrente senza ricorrere a complessi blocchi condizionali e migliorando la chiarezza del codice. Il pattern si presenta come segue:



Inoltre, l'utilizzo di questo pattern permette una facile estensione futura per l'introduzione di nuovi stati impattando in maniera minima sul codice esistente.

Command Pattern

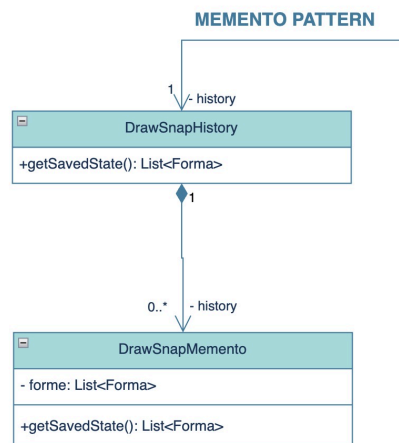
Per gestire in modo flessibile le diverse funzionalità dell'applicativo (es. copia, taglia incolla), è stato adottato il Command Pattern. Questo pattern incapsula ogni azione in un oggetto comando separato permettendo di trattare in modo uniforme le diverse operazioni e facilitandone l'esecuzione. Nel design implementato, un **Invoker** si occupa di ricevere ed invocare i comandi orchestrando la loro esecuzione, mentre i **Receiver** corrispondono ai componenti che effettivamente eseguono le azioni. In alcuni casi il Receiver corrisponde al Controller, che gestisce la logica di interazione con l'utente e gli attributi grafici dell'interfaccia. In altri si tratta del Model che si occupa di rappresentare lo stato e i dati dell'applicativo. Il pattern si presenta come segue:



Anche in questo caso, l'utilizzo di tale pattern permette future e facili estensioni tramite l'introduzione di nuovi comandi o modificando il comportamento senza impattare su altri componenti.

Memento Pattern

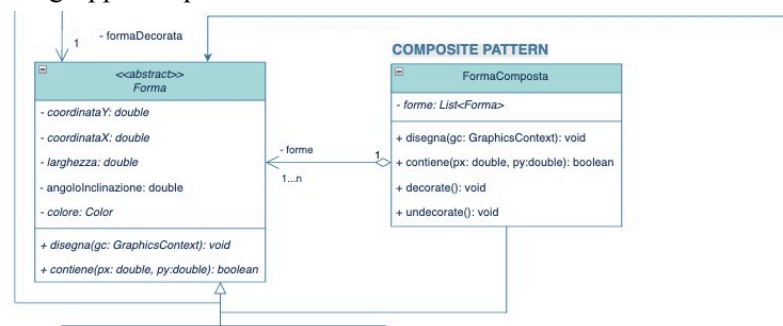
Per implementare la funzionalità di undo, ovvero la possibilità di annullare le operazioni eseguite dall'utente, è stato adottato il Memento Pattern. Questo pattern permette di catturare e salvare lo stato interno degli oggetti interessati senza violarne l'incapsulamento e creando così dei "mementi" che possono essere memorizzati e recuperati per tornare a stati precedenti. Il pattern si presenta come segue:



Questa soluzione permette la consistenza dei dati e una gestione pulita dello storico delle operazioni, migliorando l'esperienza utente e la robustezza dell'applicazione.

Composite Pattern

Per gestire insiemi di forme come singole entità, è stato adottato il Composite Pattern nella classe FormaComposta. Questo pattern consente di trattare oggetti semplici (forme elementari) e composizione di oggetti (forme composte) in modo uniforme, offrendo un'interfaccia comune per le diverse funzionalità offerte dal sistema. Grazie al Composite Pattern, FormaComposta può contenere al suo interno altre forme, anche a loro volta composte, creando così una struttura ad albero gerarchico che facilita la gestione di gruppi complessi.



Tale pattern permette di manipolare singole forme o interi gruppi con la stessa semplicità e senza differenziazioni.

Task Assignment

Dopo aver modificato il class diagram, il gruppo si è occupato di dividere nuovamente le diverse User Story selezionate per la terza Sprint in diversi task. Quest'ultime sono state aggiunte alla lista Task-to-do sulla piattaforma Trello e assegnate ai diversi componenti del team. Nella seguente tabella è riportato in maniera esaustiva e compatta le assegnazioni effettuate e se esse sono state completate o meno.

Task State	Members			
	Anzivino Giuseppe	Calabrese Raffaele	Cirillo Francesco Pio	Fasolino Alessandra
Assigned and completed tasks	T3.5.1 - T3.5.2 - T3.5.3 - T3.5.4 - T4.1.1 - T4.1.3 - T4.1.5 - T4.2.1 - T4.2.2 - T4.2.3 - T4.2.4 - T4.2.6	T4.1.2 - T4.1.4 - T4.1.6 - T4.5.1 - T4.5.2 - T4.5.3 - T4.5.4 - T5.2.1 - T5.2.2 - T5.2.3 - T5.2.4	T5.1.1 - T5.1.2 - T5.1.3 - T5.1.4 - T5.2.1 - T5.2.2 - T5.2.3 - T5.2.4	T4.3.1 - T4.3.2 - T4.3.3 - T4.3.4 - T4.4.1 - T4.4.2 - T4.4.3 - T4.4.4 - T5.3.1 - T5.3.2 - T5.3.3 - T5.3.4
Tasks assigned but not completed	×	×	×	×
Tasks not assigned to any team member	×	×	×	×

4.2. Sprint Review

Anche dopo aver terminato questa Sprint, completando le relative User Story, vi è stata una Sprint Review, nella quale sono stati evidenziati gli Story Points ultimati e le eventuali problematiche emerse durante l'implementazione delle ultime funzionalità.

User Stories complete alla fine della Sprint:

Anche al termine di questa Sprint sono state completate tutte le User Story aggiunte inizialmente. Nella seguente tabella sono riportate le User Story completate, identificate tramite i titoli definiti sulla piattaforma Trello e i relativi Story Points:

User Stories	Story Points
US3.5	2
US5.3	3
US4.2 - US4.3 - US4.4 - US4.5	5
US4.1 - US5.1 - US5.2	8

Durante la terza Sprint, la velocità effettiva ha superato quella stimata, grazie all'aggiunta e al completamento di alcune User Story extra rispetto al piano iniziale. Questo risultato evidenzia non solo un'elevata efficienza operativa del team, ma anche una buona capacità di adattamento e reattività alle esigenze emerse durante il development della Sprint. Inoltre il superamento della stima non ha nemmeno compromesso la qualità del lavoro o i tempi di consegna.

Implementation/Testing Issues

Come già fatto in precedenza, anche in questa Sprint il gruppo ha tenuto traccia dei bug e delle problematiche maggiori affrontate durante la fase di development che potevano compromettere o impedire il completamento delle User Story. Le situazioni affrontate di cui è importante tenere traccia sono essenzialmente due:

- La realizzazione della classe Poligono ha visto l'introduzione di un nuovo pattern per la sua creazione. Infatti è stato introdotto e utilizzato il pattern Builder che permette la creazione passo dopo passo dell'oggetto.
- La presenza di diversi metodi molto simili tra loro tra diverse classi concrete di Forma ha portato il gruppo a svolgere un'ulteriore distinzione di Forma in forme ad una dimensione (Forma1D) e forme a due dimensioni (Forma2D) in modo da implementare meglio il polimorfismo, rendere più leggibile e manutenibile.

Naturalmente sono stati presenti anche altri bug minori che sono stati prontamente risolti dai diversi componenti del team durante l'implementazione delle diverse funzionalità.

4.3. Sprint Retrospective

Nella Retrospective abbiamo appurato che le linee guida seguite fino ad ora erano ottimali al fine di completare la Sprint. Come si può vedere in tabella, non ci sono cambiamenti rispetto alla seconda Sprint. fatta eccezione per una nuova linea guida inserita in "More Of", ovvero quella di aggiornare il Class Diagram in concomitanza alle Commit. Questo si è reso necessario in quanto sono stati aggiunti più Command ed è stato seguito il Builder Pattern per la costruzione del Poligono.

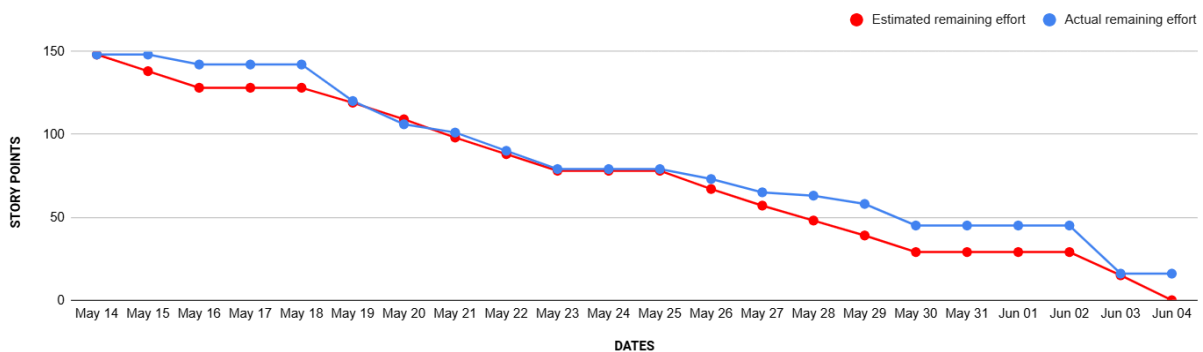
Stop Doing	Less Of	Keep Doing	More Of	Start Doing
X	X	Raggiungimento dello Sprint Goal	Aggiornare il Class Diagram in concomitanza alle Commit	X
		Resilienza agli ostacoli		
		Tener traccia dei bug e debito tecnico		

4.4. Project Release

Burndown Chart

Come è possibile constatare dall'osservazione del Burndown Chart, siamo riusciti a mantenere la velocity stimata e a restare in linea con le aspettative riguardanti il grado di completamento che avremmo raggiunto. Si fa presente che ad ogni sprint è stata tenuta traccia del carico di lavoro implicato dalle modifiche da apportare a funzioni già implementate su richiesta del product owner.

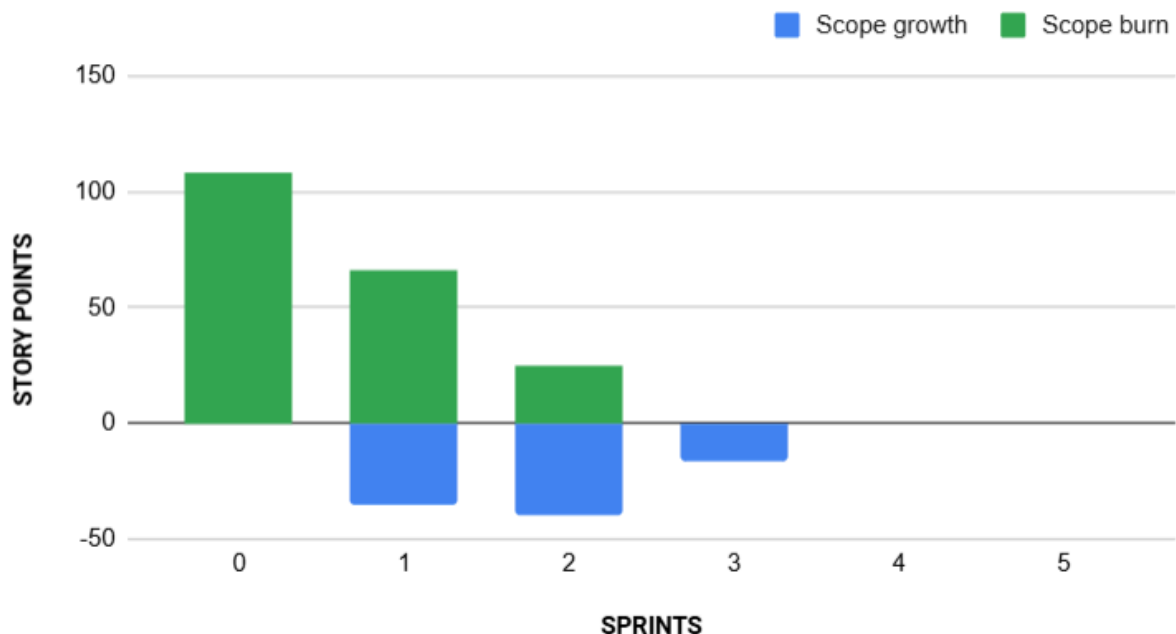
Burndown Chart SCRUM G-7



Extended Burndown Chart

Visto che durante la prima sprint il product owner ha aggiunto delle User Stories abbiamo ritenuto significativo tenerne traccia per mezzo di un Extended Burndown Chart che è visibile di seguito.

Extended Burndown Chart G-7



Considerazioni Finali

Il progetto ha seguito la metodologia Scrum, dimostrando una notevole flessibilità e adattabilità nella gestione dei requisiti. Partendo da un obiettivo di 108 Story Points, l'ambito di lavoro si è esteso a 148 grazie all'integrazione di 40 Story points aggiuntivi durante il ciclo di vita del progetto.

Il team ha mantenuto una velocità di consegna sostenuta e costante, completando un totale di 132 Story Points distribuiti in 3 Sprint (41, 41, 49). Questo ha permesso di consegnare un valore significativo in modo incrementale, rispondendo efficacemente ai cambiamenti affrontati. Nonostante l'aumento delle funzionalità da aggiungere, solo 16 Story Points non sono stati completati, evidenziando l'efficienza complessiva e fornendo preziose indicazioni per ottimizzare ulteriormente la pianificazione e la gestione dei backlog nei futuri impegni.