First Sprint Planning

Stima velocità iniziale del progetto: 28 story points

User Stories scelte per lo Sprint

Al seguente link nel foglio *User Stories 1st Sprint Planning* è allegata la tabella contenente tutte le User Stories scelte per il primo sprint.

https://docs.google.com/spreadsheets/d/1jYiJdclKfv4hlbiJVQQ3ZSFj0A611gO-KJ7naodZIP0/edit?usp=sharing

Task per ciascuna User Story

User story 0, Esplorazione

0.1 Funzionamento Scene Builder
Formazione riguardo l'utilizzo di Scene Builder.
0.2 Funzionamento JavaFX
Formazione riguardo l'utilizzo delle librerie base di JavaFx
0.3 Funzionamento Serializzazione
Formazione riguardo l'utilizzo delle librerie base di Java per la serializzazione di oggetti.

User story 1.1, Interfaccia iniziale

1.1.1 Prototipo interfaccia
Disegnare un prototipo iniziale usabile dell'interfaccia utente che includa:

- spazio di lavoro bianco
- barra in alto predisposta all'aggiunta dei vari pulsanti.

1.1.2 Design Architettura
Definire il design dettagliato (class diagram) per il modulo ***View***, in particolare per la classe che si occuperà dell'avvio dell'interfaccia grafica. Predisporre le classi dedicate al ***Controller*** e ai ***Moduli***, rispecchiando l'architettura software scelta, ovvero ***MVC***.
1.1.3 Implementazione interfaccia grafica
Implementare l'interfaccia grafica di base dell'applicazione su Scene Builder seguendo il prototipo.
1.1.4 Test funzionale dell' interfaccia
Verificare che l'interfaccia si avvii correttamente senza errori e che sia conforme agli acceptance criteria e al prototipo sviluppato.
1.1.5 Aggiornamento documentazione interfaccia
Aggiornare la documentazione dell'applicazione con il design dell' *MVC*.

User story 1.2, Selezione e creazione della forma geometrica

1.2.1 Creazione classe Forma

Creare la classe astratta ***Forma*** nel package ***Modulo***. La classe deve gestire la creazione delle forme geometriche per cui dovrà avere gli attributi relativi alle coordinate ***x*** e ***y*** dove inserire la forma.

1.2.2 Creazione classe FormaContornoChiuso

Creare la classe ***FormaContornoChiuso*** che estende la classe astratta ***Forma*** nel package Modulo. La classe deve gestire la creazione delle forme geometriche a contorno chiuso.

1.2.3 Creazione classe Linea

Creare la classe ***Linea*** che estende la classe astratta ***Forma*** nel package ***Modulo***. La classe deve gestire la creazione delle linee, quindi dovrà avere altri due attributi per memorizzare le coordinate ***x*** e ***y*** di fine linea.

1.2.4 Creazione classe Ellisse

Creare la classe ***Ellisse*** che estende la classe astratta ***FormaContornoChiuso*** nel package *Modulo*. La classe deve gestire la creazione dell'ellisse, quindi dovrà avere altri due attributi per memorizzare le dimensioni di quest'ultima, ovvero ***raggioX*** e ***raggioY***.

1.2.5 Creazione classe Rettangolo

Creare la classe ***Rettangolo*** che estende la classe astratta ***FormaContornoChiuso*** nel package ***Modulo***. La classe deve gestire la creazione del rettangolo, quindi dovrà avere altri due attributi per memorizzare le dimensioni di quest'ultimo, ovvero ***base*** e ***altezza***.

1.2.6 Creazione classe Factory

Creare la classe ***Factory***, necessaria per l'implementazione del pattern creazionale ***Factory Method*** che si occuperà di istanziare i concrete product mediante il factory method ***creaForma()***.

1.2.7 Creazione bottoni forme

Aggiornamento della toolbar presente nella ***View*** con un'area dedicata all'inserimento delle forme in cui sono inclusi tre bottoni uno per ogni forma in ambiente Scene Builder, rispettivamente per ***linea***, ***ellisse*** e ***rettangolo***, che permetta la scelta della forma geometrica.

1.2.8 Integrazione tra modulo View e creazione forma

Implementazione dei metodi all'interno di ***Model*** necessari alla cattura dell'evento (che avviene nel Controller) di pressione sulla lavagna/finestra integrandolo con la creazione della specifica forma tramite ***Factory Method***. La forma creata deve corrispondere a quella selezionata tramite bottone.

1.2.9 Scrittura test unitari

Scrivere i test unitari per verificare il corretto funzionamento di ogni classe definita nella User Story.

1.2.10 Scrittura test integrazione

Scrivere test di integrazione per verificare che la pressione del pulsante relativo a una figura comporti la corretta creazione della figura selezionata e l'aggiornamento della ***View***.

1.2.11 Aggiornamento documentazione

Aggiornare la documentazione del design (class diagram) con l'analisi delle classi implementate per questa user story.

User Story 1.3, Creazione della forma con il colore del bordo selezionato

1.3.1 Aggiornamento classe Forma
Aggiornare la classe ***Forma***, seguendo il design, per l'inclusione del nuovo attributo "coloreBordo", riflettendo la modifica in tutte le sottoclassi.
1.3.2 Creazione tavolozza colori per bordo
Aggiornamento della toolbar presente nella ***View*** con un'area dedicata alla scelta del colore del bordo in ambiente Scene Builder.
1.3.3 Integrazione tra scelta colore di bordo e creazione forma
Aggiornamento dei metodi all'interno di ***Model*** responsabili della creazione della specifica forma tramite factory method con integrazione del colore di bordo scelto.
1.3.4 Estensione Unit test
Estensione e aggiornamento dei test unitari delle classi coinvolte, ovvero ***Forma*** e quelle che la estendono.
1.3.5 Estensione test integrazione
Estensione e aggiornamento dei test di integrazione per verificare che la pressione del pulsante relativo a una figura comporti la corretta creazione della figura selezionata con il relativo colore di bordo scelto e l'aggiornamento della ***View***.
1.3.6 Aggiornamento documentazione
Aggiornamento della documentazione relativa alle modifiche apportate per la User Story implementata.

User Story 1.4, Creazione della forma con il colore di riempimento selezionato

1.4.1 Aggiornamento classe FormaContornoChiuso
Aggiornare la classe ***FormaContornoChiuso***, seguendo il design, per l'inclusione del nuovo attributo "coloreRiempimento", riflettendo la modifica in tutte le sottoclassi.
1.4.2 Creazione tavolozza colori per riempimento
Aggiornamento della toolbar presente nella ***View*** con un'area dedicata alla scelta del colore del riempimento in ambiente Scene Builder.
1.4.3 Integrazione tra scelta colore di riempimento e creazione forma
Aggiornamento dei metodi all'interno di ***Model*** responsabili della creazione della specifica forma tramite factory method con integrazione del colore di riempimento scelto.
1.4.4 Estensione Unit test
Estensione e aggiornamento dei test unitari delle classi coinvolte, ovvero ***FormaContornoChiuso*** e quelle che la estendono.
1.4.5 Estensione test integrazione
Estensione e aggiornamento dei test di integrazione per verificare che la pressione del pulsante relativo a una figura comporti la corretta creazione della figura selezionata con il relativo colore di riempimento scelto e l'aggiornamento della ***View***.
1.4.6 Aggiornamento documentazione
Aggiornamento della documentazione relativa alle modifiche apportate per la User Story implementata.

User story 2.1, Eliminazione della forma selezionata

2.1.1 Aggiunta componente grafico

Aggiunta del componente grafico, ad ogni forma creata e inserita nella ***View***, in maniera tale che al click destro sulla forma presenti l'elemento predisposto per l'eliminazione.

2.1.2 Implementazione della logica di cancellazione

Implementazione della logica necessaria a eseguire la cancellazione della forma selezionata, rimuovendola dalla ***View***, nel momento in cui il componente grafico viene premuto.

2.1.3 Esecuzione Test funzionale cancellazione

Eseguire functional test cases rispetto alla funzionalità di cancellazione implementata.

User story 2.2, Spostamento della forma selezionata

2.2.1 Implementazione della logica di spostamento

Implementazione della logica necessaria a eseguire lo spostamento della forma selezionata, all'interno della ***View***, nel momento in cui la forma viene trascinata.

2.2.2 Scrittura test unitari modifica posizione

Scrivere i test unitari per verificare il corretto funzionamento dell'operazione di modifica delle coordinate legate allo spostamento della forma.

2.2.3 Esecuzione test funzionale spostamento

Eseguire functional test cases rispetto alla funzionalità di spostamento implementata.

User story 2.5, Spostamento della forma selezionata

2.5.1 Aggiunta componente grafico

Aggiunta del componente grafico, ad ogni forma creata e inserita nella ***View***, in maniera tale che al click destro sulla forma presenti l'elemento predisposto per la modifica delle sue dimensioni (che dipendono dalla forma selezionata).

2.5.2 Implementazione della logica di modifica delle dimensioni

Implementazione della logica necessaria a eseguire la modifica delle dimensioni della forma selezionata, nel momento in cui vengono specificate le dimensioni e il componente grafico viene premuto.

2.5.3 Scrittura test unitari modifica dimensioni

Scrivere i test unitari per verificare il corretto funzionamento dell'operazione di modifica delle dimensioni.

2.5.4 Esecuzione test funzionale modifica dimensioni

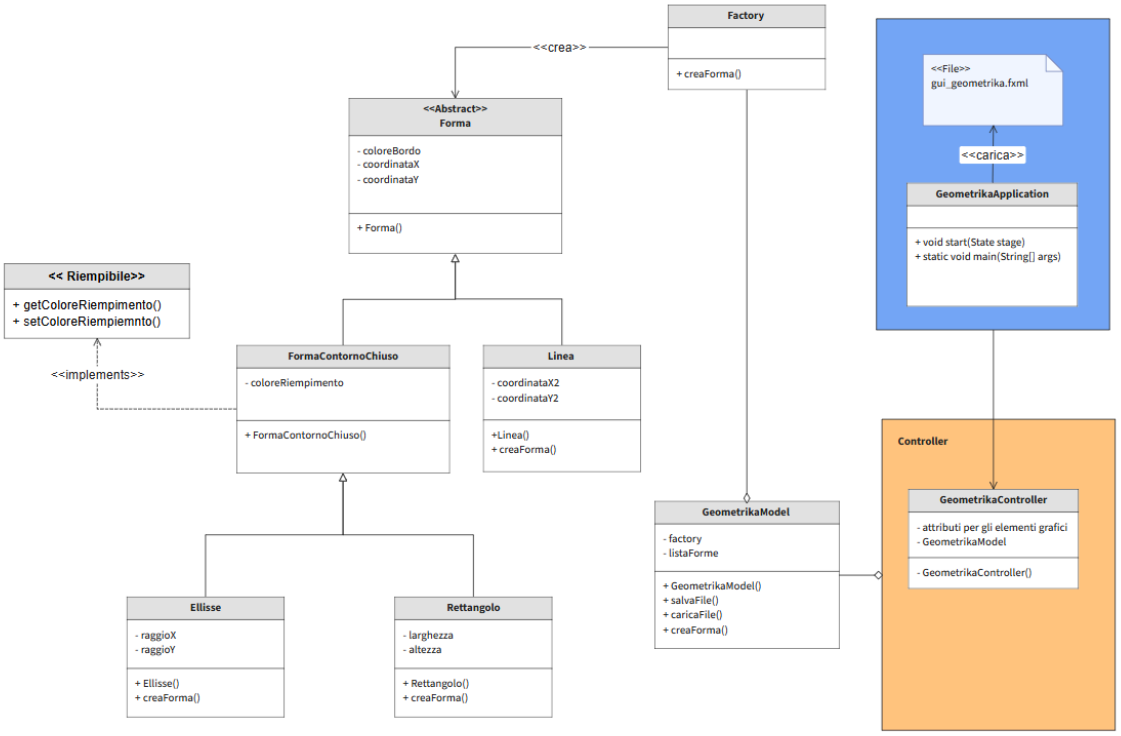
Eseguire functional test cases rispetto alla funzionalità di modifica delle dimensioni implementata.

Sprint Goal

L'obiettivo di questa prima Sprint è permettere all'utente di utilizzare l'applicazione di disegno geometrico che si sta sviluppando per effettuare le operazioni basiche, quali creazione della forma desiderata con i colori scelti, e le operazioni di modifica, spostamento ed eliminazione di forme precedentemente create.

Design

Cliccando sull'immagine seguente è possibile analizzare a schermo intero il Class Diagram costruito durante la fase di design iniziale.



In base all'architettura software scelta, ovvero MVC, è possibile analizzare il diagramma delle classi raggruppando queste nelle tre componenti, ovvero Moduli, Vista e Controllore.

Modelli

I **modelli** si occupano della logica di business e della gestione dei dati.
La classe centrale è ***GeometrikaModel***, che gestisce lo stato dell'applicazione e contiene una lista di oggetti ***Forma*** e un'istanza della classe ***Factory***, utilizzata per la creazione delle forme tramite il pattern **Factory Method**.

La classe astratta ***Forma*** rappresenta la base per tutte le forme geometriche e contiene attributi comuni come il colore del bordo e le coordinate. Da essa derivano le classi concrete:

- ***Linea***, che aggiunge le coordinate di un secondo punto, necessarie per la creazione del segmento;

- ***FormaContornoChiuso***, che introduce il concetto di riempimento tramite l'attributo ***coloreRiempimento*** e l'implementazione dell'interfaccia ***Riempibile***, la quale definisce i metodi ***getColoreRiempimento()*** e ***setColoreRiempimento()***.

A loro volta, ***Ellisse*** e ***Rettangolo*** estendono ***FormaContornoChiuso***, aggiungendo rispettivamente ***raggioX*** e ***raggioY*** per l'ellisse, e ***base*** e ***altezza*** per il rettangolo.
Infine, la classe ***Factory*** contiene il vero e proprio factory method, ovvero ***creaForma()***, responsabile della creazione delle istanze concrete delle varie forme.

Controllore

Il **controllore**, rappresentato dalla classe ***GeometrikaController***, funge da intermediario tra vista e modelli. È composto da un'istanza di ***GeometrikaModel*** e contiene riferimenti agli elementi dell'interfaccia grafica (che verranno definiti in fase di sviluppo).

Vista

La **vista** è rappresentata dalla classe ***GeometrikaApplication*** contiene i metodi ***main()*** e ***start()*** per avviare l'app e caricare l'interfaccia definita nel file FXML.

Il file ***gui_geometrica.fxml*** descrive graficamente l'interfaccia utente (layout, bottoni, ecc.), separando completamente la logica applicativa dalla presentazione.

Questa struttura rispetta i principi dell'architettura MVC, separando chiaramente i compiti tra gestione dei dati, interfaccia utente e coordinamento delle interazioni.