**Relazione Progetto**

**Matteo Toma, matricola 116781**

Per realizzare un programma che permette di “programmare disegni” sono state utilizzate Java 17.0.1 e per quanto riguarda Gradle, il plugin org.openjfx.javafxplugin con versione 0.0.10.

E’ stato inoltre definito un file module-info.java contenent e le componenti necessarie ad avviare un’applicazione grafica: javafx.controls e javafx.fxml. Contiene anche la componente per visualizzare i logger: java.logging .

**Gestione delle responsabilità**

Come pattern di progettazione si è scelto l’MVC (**Model-View-Controller),** in quanto è abbastanza semplice da applicare al progetto non essendo particolarmente complesso.

**Model**

In questo package bisogna definire tutte le classi e le interfacce necessarie per rappresentare l’applicazione d’interesse.

Servono classi che rappresentino e gestiscano il piano cartesiano, altre per la rappresentazione della penna, altre per la sua direzione.

Fondamentale sarà la classe che si occupa di implementare tutti quelli che saranno le istruzioni dei comandi da interpretare.

Servirà una classe per poter importare tutti i colori, della penna e dello sfondo del piano.

Serviranno classi per poter memorizzare/calcolare informazioni quali direzione del cursore e informazioni sull’area.  
Per poter capire come sono configurati e come rappresentare i punti delle rette nel piano del disegno, verrà usata una classe per rappresentare dei grafi non orientati.

**View**

Le classi del package View si occuperanno di istanziare tutte le componenti grafiche dell’interfaccia, inizializzarle e resettarle, infine dovranno gestire gli eventi associati ad ogni pulsante.

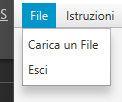
**Controller**

Le classi e interfacce del package controller, costruiranno il piano di lavoro in base alle dimensioni specificate.   
Dovranno permettere di poter mostrare la precedente configurazione del piano e quindi permettere di mostrare il piano DOPO e PRIMA di un’istruzione letta dal file.  
Dovrà inoltre gestire i casi in cui le istruzioni nel file non esistano tra quelle esistenti, e controllare che la sintassi sia giusta.

**INTERFACCE**

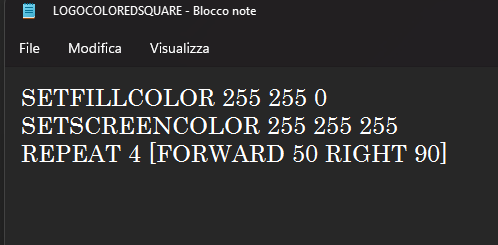
**Funzionamento del programma**

Una volta avviata l’interfaccia grafica, premendo su File e scegliendo un file contenente le istruzioni necessarie a creare un’immagine, verranno mostrati i seguenti comandi:

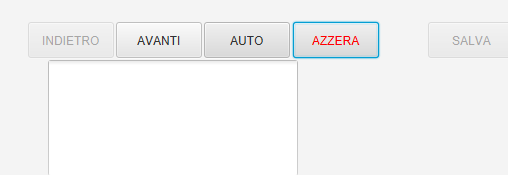


Premendo carica File, si potrà scegliere il file .txt contenente le informazioni in linguaggio Logo.

Un esempio di file è questo:



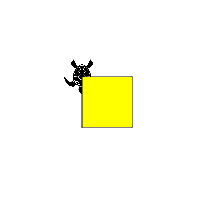
Una volta caricato il file, verrà mostrata la seguente schermata:



Premendo “avanti”, verranno mostrate le istruzioni step by step (contemporaneamente, verranno interpretate ed eseguite), mentre premendo auto verranno interpretate ed eseguite tutte insieme.



Il risultato finale è questo:



**Esempi di comandi.**

Ad esempio, per disegnare un cerchio colorato, si possono scrivere i seguenti comandi:

**SETFILLCOLOR** 255 102 255 #per impostare il colore di riempimento dell’area di rosa

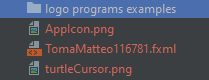
**SETPENCOLOR** 255 255 255 #per impostare il colore del tratto della penna di bianco

**SETSCREENCOLOR** 153 153 255 #per impostare il colore di sfondo di azzurro.

**LEFT** 90 #per ruotare il cursore verso sinistra (di 90 gradi)

**REPEAT** 360 [ FORWARD 2 RIGHT 1 ] #per ripetere l’operazione 360 volte, andando in avanti di due e una volta a destra. Infatti aumentando, ad esempio, il forward a 3, aumenterà il raggio del cerchio.

**File presenti nella cartella “resources”**



1. I programmi Logo di esempio
2. L’icona dell’app
3. Il file fxml che rappresenta la grafica.
4. L’icona del cursore.

**Descrizione delle classi usate per rappresentare i programmi Logo**

**Descrizione delle classi usate per rappresentare il disegno prodotto da un programma Logo**

**Descrizione di come le gerarchie possano essere “estese” per aggiungere nuovi comandi o disegni.**