

# ING1 - MIM

# RAPPORT

PROJET  
CHROMAT-YNK

```
y {  
  switch (command) {  
    case "FWD":  
      if (parts.length < 2)  
        throw new IllegalArgumentException(  
          "distance must be at least 2"  
        );  
      double distance = Double.parseDouble(parts[1]);  
      double newX = x + distance * Math.cos(Math.toRadians(0));  
      double newY = y + distance * Math.sin(Math.toRadians(0));  
      gc.strokeLine(x, y, newX, newY);  
      x = newX;  
      y = newY;  
      break;
```

EFE TATAR, NATHAN SIGNOUD,  
FRANÇOIS BRETAGNE, ALIZE  
FORTUNEL

# Sommaire

---

Présentation du groupe

---

Organisation

---

Diagramme UML

---

Problèmes rencontrés

---

Solution apportées

---

Résultats

# PRESENTATION DE NOTRE PROJET

Ce projet vise à créer un programme pour dessiner des formes à partir d'instructions (..), avec la possibilité de modifier l'épaisseur et la couleur des traits. Il inclura un langage d'instructions et un interpréteur pour les exécuter. Le programme permettra également de sauvegarder et charger des créations, tout en les dessinant.

Un groupe de quatre étudiants de première année du programme  
Double Diplôme Ingénieur-Manager (ING1 - MIM)



EFE  
TATAR



FRANÇOIS  
BRETAGNE



NATHAN  
SIGNOUD



ALIZE  
FORTUNEL

Efe et François ont travaillé sur la tokenisation et l'analyse syntaxique, le parser.

Optimisation de la performance - Gestion des erreurs - Étendre le parser

Nathan et Alizé ont travaillé sur l'interface ainsi que sur le rapport.

Amélioration de l'interface utilisateur - Intégration de fonctionnalités - Tests

Les deux équipes ont travaillé en collaboration notamment en communiquant de manière ouverte et régulière pour partager les problèmes rencontrés ainsi que les solutions apportées. Elles ont commenté leurs codes afin qu'ils soient compréhensibles pour l'autre équipe.

# ORGANISATION :

Rendez-vous avec Redouane Bouhamoum	7 MAI : 11h-11h30	16 MAI : 16h-16h30	22 MAI : 17h-17h30	24 MAI : 10h-10h30
Mise en commun :	7 MAI : 11h30-13h	17 MAI : 17h-18h30	22 MAI : 17h30-18h30	25 MAI : 16h-19h

T  
E  
A  
F  
T  
E  
A  
R

7 MAI : 15h-17h	8 MAI : 10h-11h	15 MAI : 14h-16h30	17 MAI : 16h30-18h30	18 MAI : 14h-17h30	20 MAI : 17h-19h30
21 MAI : 15h-18h30	22 MAI : 17h30-20h30	23 MAI : 13h-17h	24 MAI : 15h-19h30	25 MAI : 15h - 21h	26 MAI : 14h - 21h

7 MAI : 15h-17h	15 MAI : 17h-19h30	17 MAI : 17h-18h30	19 MAI : 14h-16h	20 MAI : 16h30-18h	21 MAI : 16h-18h30
	22 MAI : 14h-18h30	23 MAI : 15h30-18h30	24 MAI : 16h-18h	25 MAI : 16h - 20h	26 MAI : 15h - 19h

F  
B  
R  
R  
A  
E  
N  
T  
C  
A  
O  
G  
I  
N  
S  
E

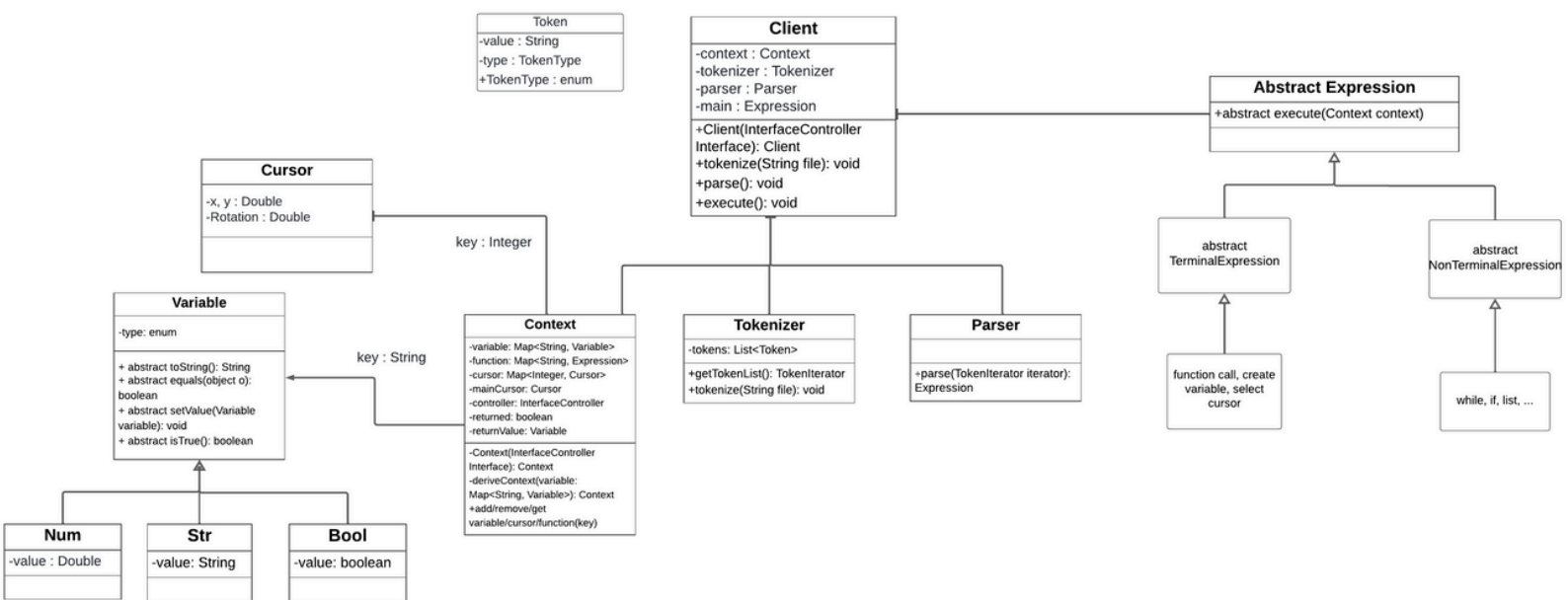
N  
S  
I  
G  
N  
A  
T  
H  
A  
N  
D

7 MAI : 15h-17h	15 MAI : 16h-17h30	17 MAI : 17h-19h30	19 MAI : 17h-19h	20 MAI : 19h30-21h	21 MAI : 18h-19h30
	22 MAI : 15h30-18h30	23 MAI : 18h30-20h	24 MAI : 16h-18h30	25 MAI : 16h - 21h	26 MAI : 14h - 21h

7 MAI : 15h-17h	13 MAI : 16h-17h30	15 MAI : 16h-18h30	17 MAI : 17h-19h30	20 MAI : 16h30-18h30	21 MAI : 16h-19h30
	22 MAI : 14h-18h30	23 MAI : 15h30-18h30	24 MAI : 10h30-13h	25 MAI : 16h - 20h	26 MAI : 16h - 20h

F  
O  
R  
T  
U  
N  
E  
L  
A  
L  
I  
Z  
E

# DIAGRAMME



# PROBLÈMES RENCONTRÉS

```
( For animation a
import javafx.anim
import javafx.anim
( JavaFX image co
import javafx.embe
import javafx.fxml
( Various compone
import javafx.scen
import javafx.scen
import javafx.scen
import javafx.scen
import javafx.scen
import javafx.scen
import javafx.scen
import javafx.scen
import javafx.scen
```

## Parsing des arguments de fonctions

Problème de limite de token

## Récurtivité des fonctions

Ne fonctionne pas dans notre programme. Pour des raisons d'organisation temporelle nous avons classé ce problème comme mineur

## Parsing des opérations arithmétique

Algorithme de Shunting Yard

## Conception de l'architecture du projet

## Gestion de la vitesse

Gestion de la vitesse d'exécution du dessin dans l'interface et lors de la fusion des différents codes, les boutons ont disparus

## Ajout d'un fichier CSS

Problème de chargement avec le fichier CSS pour appliquer les styles à l'interface graphique

## Définition des chemins

les fichiers FXML et CSS étaient mal définis ce qui génèrait des erreurs

## Gestion des exceptions

Problème car des exceptions n'étaient pas levé

# SOLUTIONS APPORTÉES



## Parsing des arguments de fonctions

Extraction d'une sous liste de token

## Récursivité des fonctions

Nous n'avons pas trouvé de solutions à ce problème

## Parsing des opérations arithmétique

Nous nous sommes renseigné sur internet et notamment avec l'algorithme de Shunting Yard qui nous a permis de régler ce problème.

## Gestion de la vitesse

Apport de 3 boutons, lent, moyen et rapide

## Ajout d'un fichier CSS

Correction du chemin vers le fichier CSS en m'assurant qu'il était correctement défini dans le fichier FXML et que le fichier CSS soit présent dans le bon répertoire.

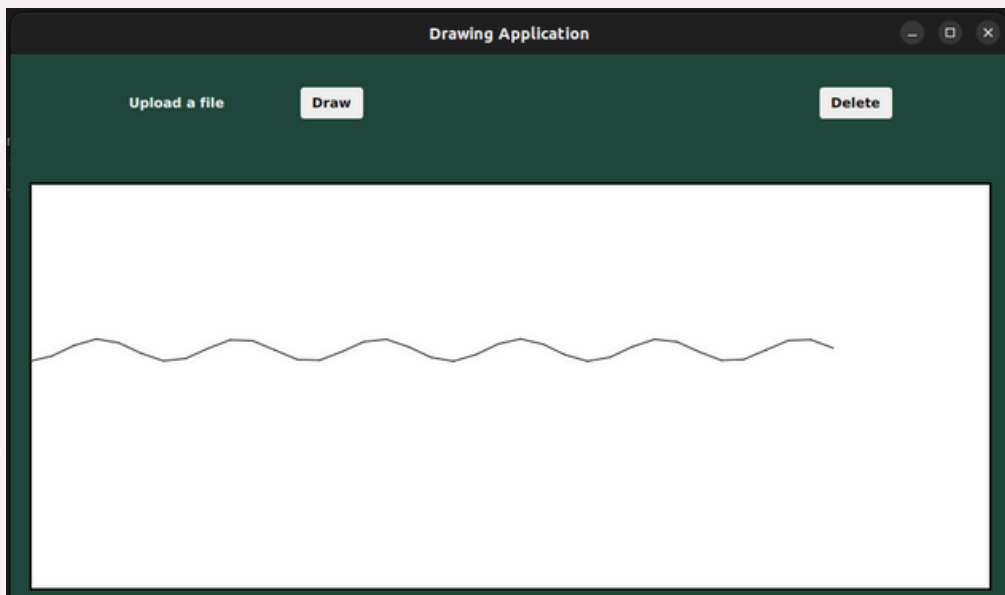
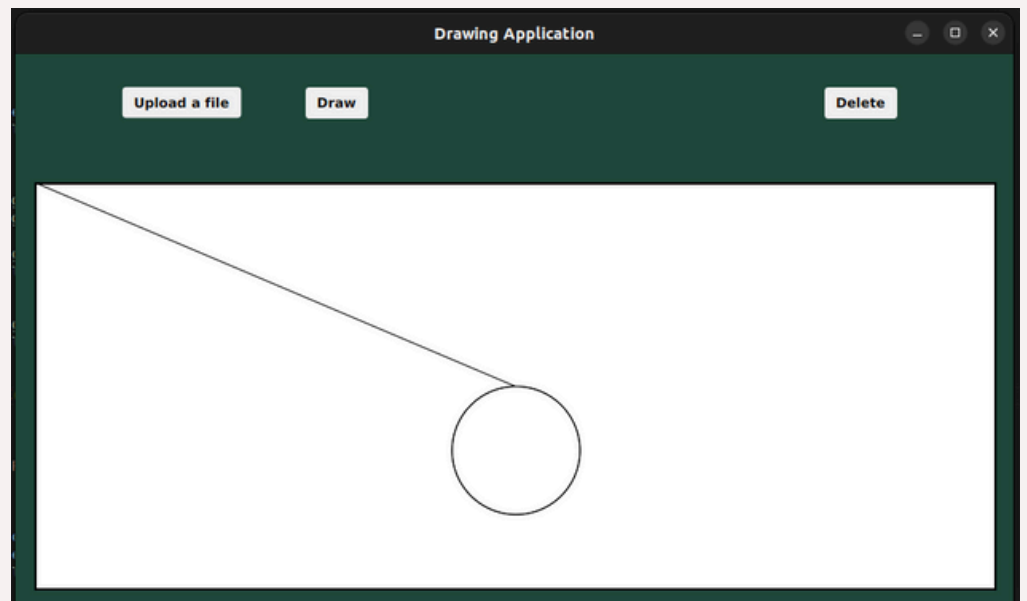
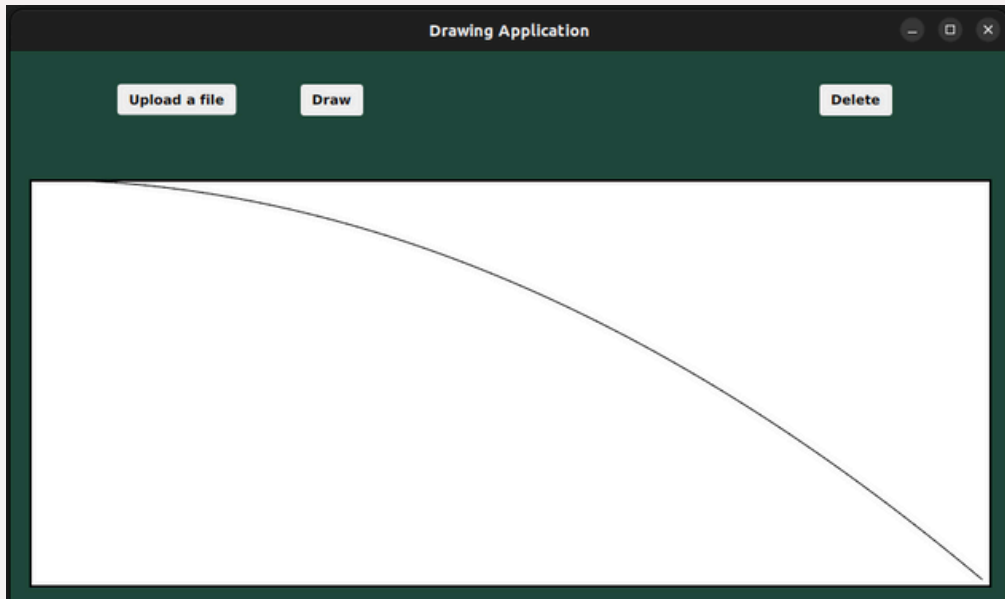
## Définition des chemins

Utilisation de la méthode `getClass().getResource()`

## Gestion des exceptions

Ajout de blocs try-catch pour gérer les exceptions lors du chargement des fichiers FXML

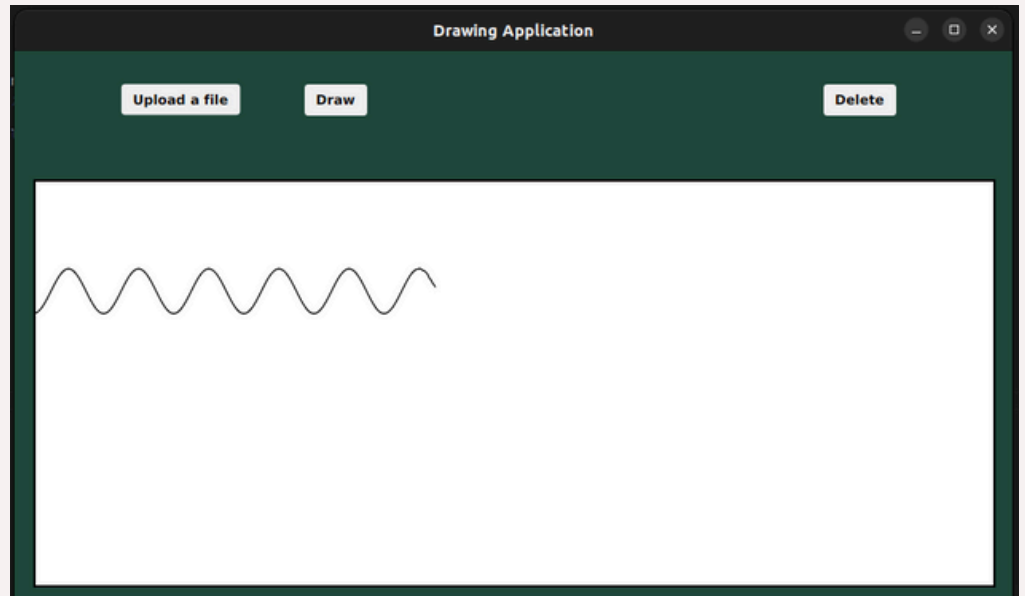
# RESULTS





# RESULTS

Nous avons obtenu ce graphique grâce à la fonction suivante



```
DEF power{  
  IF(argc < 2){  
    RETURN(0-1);  
  }  
  
  NUM x = arg0;  
  NUM n = arg1;  
  NUM pwr = 1;  
  
  IF(n == 0)  
  {  
    RETURN(1);  
  }  
  
  NUM i = 1;  
  
  WHILE(i <= n){  
    pwr = pwr * x;  
    i = i + 1;  
  }  
  
  RETURN(pwr);  
}
```

```
DEF fact{  
  IF(argc < 1){  
    RETURN(0-1);  
  }  
  
  NUM i = 1;  
  NUM fact = 1;  
  
  WHILE(i <= arg0){  
    fact = fact * i;  
    i = i+1;  
  }  
  RETURN(fact);  
}
```

```
DEF cos{  
  IF(argc < 1){  
    RETURN(0-2);  
  }  
  
  NUM x = arg0;  
  
  NUM pwr;  
  NUM poly;  
  NUM fact;  
  
  NUM sum = 1;  
  NUM n = 1;  
  
  WHILE(n < 100){  
    pwr = power( (0-1), n );  
    poly = power(x, 2*n);  
    fact = fact(2 * n);  
    sum = sum + pwr * poly / fact;  
    n = n + 1;  
  }  
  
  RETURN sum;  
}
```

```
CURSOR 1;  
SELECT 1;  
  
NUM i = 0;  
NUM res;  
  
WHILE(i < 800){  
  res = cos(i/10);  
  POS([i, res*20 + 100]);  
  i = i+1;  
}  
  
RETURN(1);
```

# CONCLUSION

Ce projet créer une application de dessin capable d'interpréter et d'exécuter des instructions à partir de fichiers texte, d'instructions par l'utilisateur.

Malgré quelques difficultés, le projet est donc fonctionnel.

L'objectif principal étant de permettre aux utilisateurs de créer des dessins à partir de simples instructions.