Graff Mathieu
Salzard Manech
Vela-Mena Antoine
Klingler Emma

S3D

SAE - Développement d'une application

LES FONCTIONNALITES DE NOTRE APPLICATION :

- Introspection textuelle
- Modification de l'affichage des classes
- Exportation en image et en Plantuml
- Drag and Drop
- Création de classes via l'interface utilisateur
- Changer la couleur du bloc

Itération 1 :

- Implémentation des patrons MVC et Fabrique
- Implémentation du squelette de chaque classe et de leur test
- Implémentation de l'interface graphique avec l'apparition de menu contextuels

Itération 2 :

- Manech
  - Implémentation de la vue Textuelle
  - Implémentation de la classe Fichier +importation du patron Composite
- Mathieu
  - Ajouter bloc
  - Implémentation d'un bouton de suppression d'un bloc
- Emma
  - Exportation en image
- Antoine
  - Implémentation du treeview

Itération 3

- Manech
  - Génération Plantuml
- Mathieu
  - Implémentation des flèches
- Emma
  - Modifier Bloc
- Antoine
  - Implémentation du Drag and Drop

Graff Mathieu
Salzard Manech
Vela-Mena Antoine
Klingler Emma

Itération 4

- Manech
  - Visualisation des classes sur l'interface (petite image et sous forme de bloc)
- Mathieu
  - Affichage des différentes flèches
- Emma
  - Vue simple/Vue complexe des classes en globalité et individuellement
- Antoine
  - Création de classes via l'interface utilisateur

Itération 5

- Manech
  - Importation d'un projet depuis n'importe quel répertoire
- Mathieu
  - Affichage des courbes des flèches et des cardinalités
- Emma
  - Changement de couleur
- Antoine
  - Bouton de suppression de toutes les classes
  -

LES FONCTIONNALITES QUI FONT NOTRE FIERTE

Manech : Importation d'un projet depuis n'importe quel répertoire sur la machine

- Récupération des fichiers
- Suppression du dossier projClass s'il existe
- Compilation des fichiers
- Enregistrement des fichiers compiler dans un dossier projClass
- Chargement des class avec ClassLoader afin de pouvoir les lires

Emma : L'exportation en Image

L'exportation en image est gérée par la méthode exportAsImage(). Elle utilise la méthode snapshot() pour capturer l'affichage du Pane (le viewport) sous forme d'un WritableImage. Ce dernier est ensuite converti en BufferedImage afin de pouvoir l'enregistrer. Un dialogue de fichier permet à l'utilisateur de choisir l'emplacement et le format de l'image (PNG ou JPEG). Enfin, l'image est sauvegardée dans le format sélectionné grâce à la méthode ImageIO.write().

Graff Mathieu
Salzard Manech
Vela-Mena Antoine
Klingler Emma

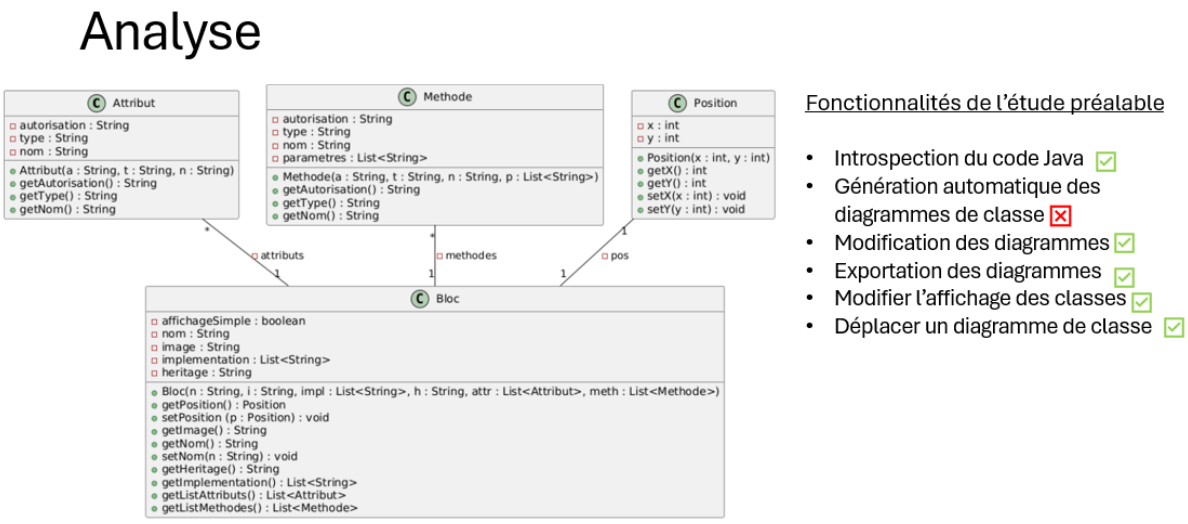
Mathieu : Calcul de la position des flèches

Pour calculer la position, je prends la position du centre des blocs auxquels elle est liée et je calcule une fonction affine qui correspond à la courbe qui va relier les blocs. J'utilise ensuite une variable x qui va avancer sur la courbe jusqu'à ce qu'elle soit au bord du bloc. Ça sera la position d'un côté et le même processus sera utilisé pour l'autre côté.

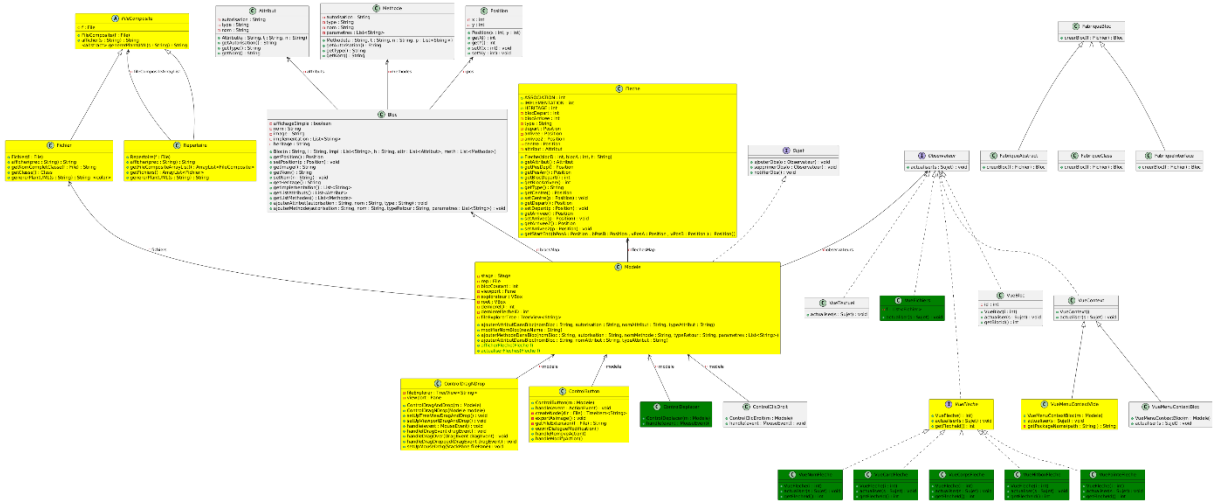
```
// Pour calculer y du départ et de l'arrivée, je vais utiliser ax + b qui part du milieu du premier bloc et arrive au milieu du deuxième
// a -> coefficient directeur (yb - ya)/(xb - xa)
double a = (bCentre.getY() - aCentre.getY()) / (bCentre.getX() - aCentre.getX());
// b = y - ax
double b = aCentre.getY() - aCentre.getX() * a;
```

Antoine : Drag and Drop

Le drag and drop était la partie fun dans un sujet sérieux, pouvoir déplacer des fichiers d'un coté à un autre, avoir l'aperçue de l'amélioration au fur et à mesure du projet avec l'ajout des blocs, des textes, des couleurs.

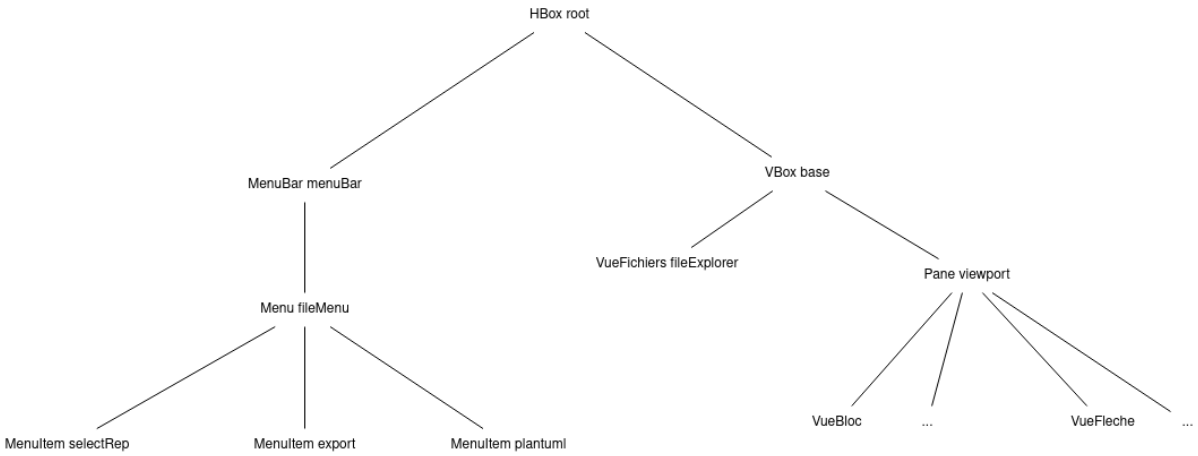


Le diagramme de classe final : (nous avons gardé les mêmes patrons que lors de l'analyse)



Graff Mathieu
Salzard Manech
Vela-Mena Antoine
Klingler Emma

le graphe de scène



Mode d'emploi :

Mettre le dossier image en "ressource root" sur intelliJi et utiliser la version JDK 20 et JavaFx 20

Fonction de création d'un squelette de classe depuis le logiciel:

Il faut d'abord se placer à l'endroit ou se trouve votre projet java (plus précisément là où vous voulez que le fichier se créé)

Pour éviter les erreurs :

-mettre des types primitifs ou des classes déjà existant pour les attributs

-mettre des virgules entre chaque attributs et chaque méthodes

-respecter l'ordre d'écriture suivant pour les méthodes sans oublier les espaces(il n'y a pas d'espace entre le nom de la méthode et les paramètres : [modificateur d'accès] [type retourné par la méthode] [nom de la méthode]([paramètre de la méthode])

-les paramètres doivent être séparé par des virgules et sont entourés par des parenthèses et ils ont la syntaxe suivante : [type] [nom] Exemple d'écriture d'attributs : int ok,String nom Exemple d'écriture de méthodes : public void setNom(String n,int age), public String getNom() Si jamais ces consignes ne sont pas respecté, le fichier risque de ne pas compiler, il n'apparaîtra donc pas sur la vue des fichiers et ne sera donc pas utilisable sur le diagramme. Néanmoins le fichier sera quand même créé dans le répertoire de votre projet. Il est impossible d'ouvrir un projet java si l'un des fichiers java ne compile pas. Pour les flèches il est possible de cliquer sur une fleche et de la déplacer afin de modifier sa courbe et l'emplacement de l'association