

# **IFT-2007 / GLO-2004**

## **Projet de Session Livrable 2**

Présenté à Anthony Deschênes

### **Équipe 6**

Amenallah Massarra Ouannes

Mohamed Hedi Braham

Fares Majdoub

Katia Bououni

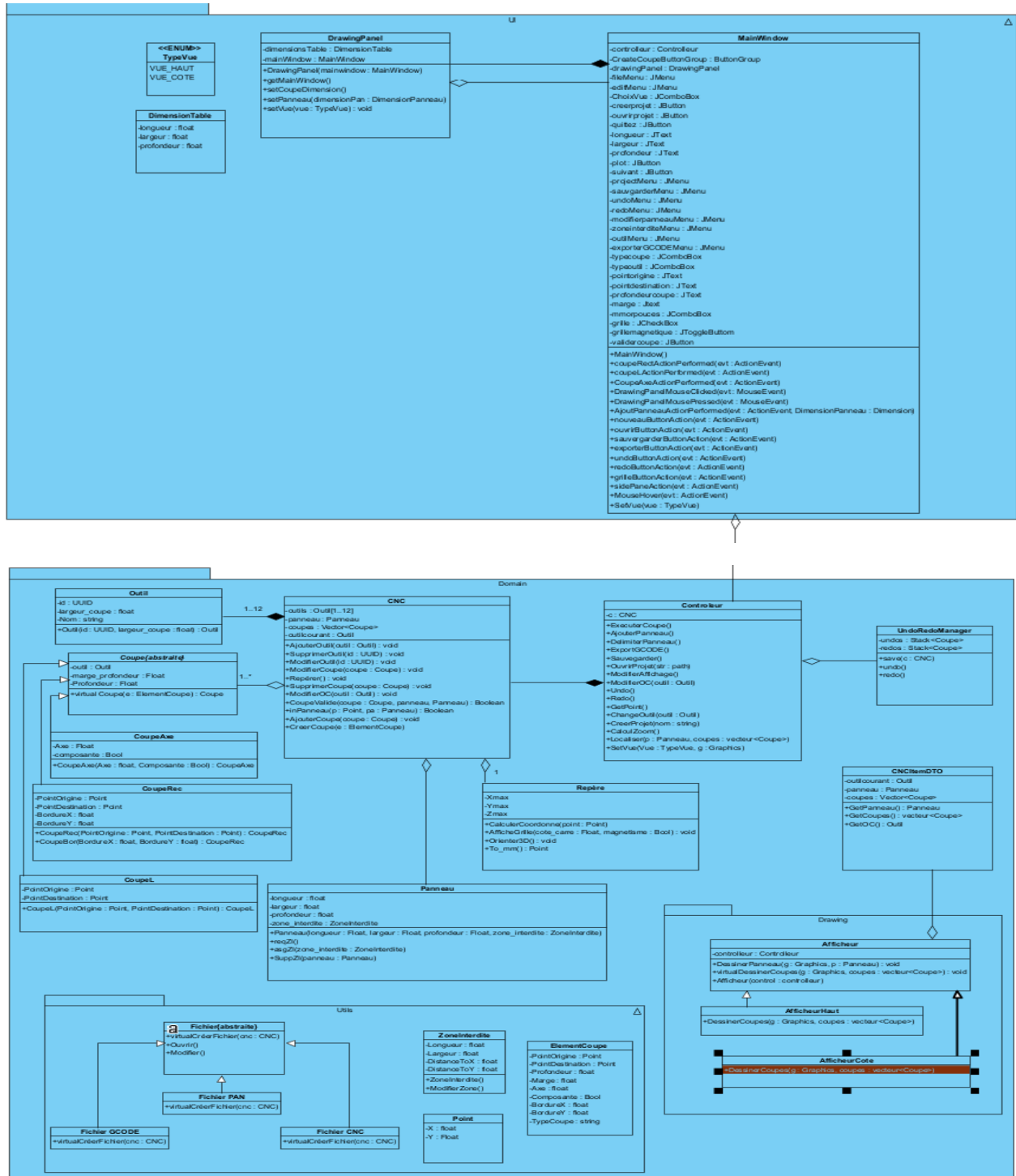
Zied Daly

Session d'Automne 2024

**Table des matières :**

1.	Diagramme des classes de conception .....	3
2.	Architecture logique .....	5
3.	Diagrammes de séquences de conception (DSC) .....	6
	3.1.1 Déterminer l'élément sur lequel a lieu un clic de souris.....	6
	3.1.2 Déterminer l'élément sur lequel a lieu un clic de souris.....	7
	3.2 L'ajout d'une coupe rectangulaire .....	8
	3.3 Affichage de la vue :.....	9
4.	Squelette de l'interface .....	10
5.	Diagrammes de Gantt .....	13
6.	Contributions des membres .....	14

## 1. Diagramme des classes de conception



Le diagramme de classes de conception illustre les principales classes et méthodes de l'application CNCCLT, en mettant en avant celles qui répondent aux exigences du livrable.

Au cœur de l'application se trouve la classe `Controleur`, qui représente le contrôleur de Larman dans le package `domaine`. Cette classe est responsable de la gestion des appels aux objets et méthodes du modèle.

La classe `CNC`, quant à elle, occupe une place centrale, car elle contient toutes les informations pertinentes pour le projet. Elle est subdivisée en trois classes principales : la classe `Outil`, qui décrit les outils de coupe, la classe `Panneau`, qui caractérise le panneau, et la classe abstraite `Coupe`. Cette dernière est elle-même divisée en sous-classes (`CoupeRec`, `CoupeL`, et `CoupeAxe`), qui gèrent les différents types de coupes possibles dans le projet.

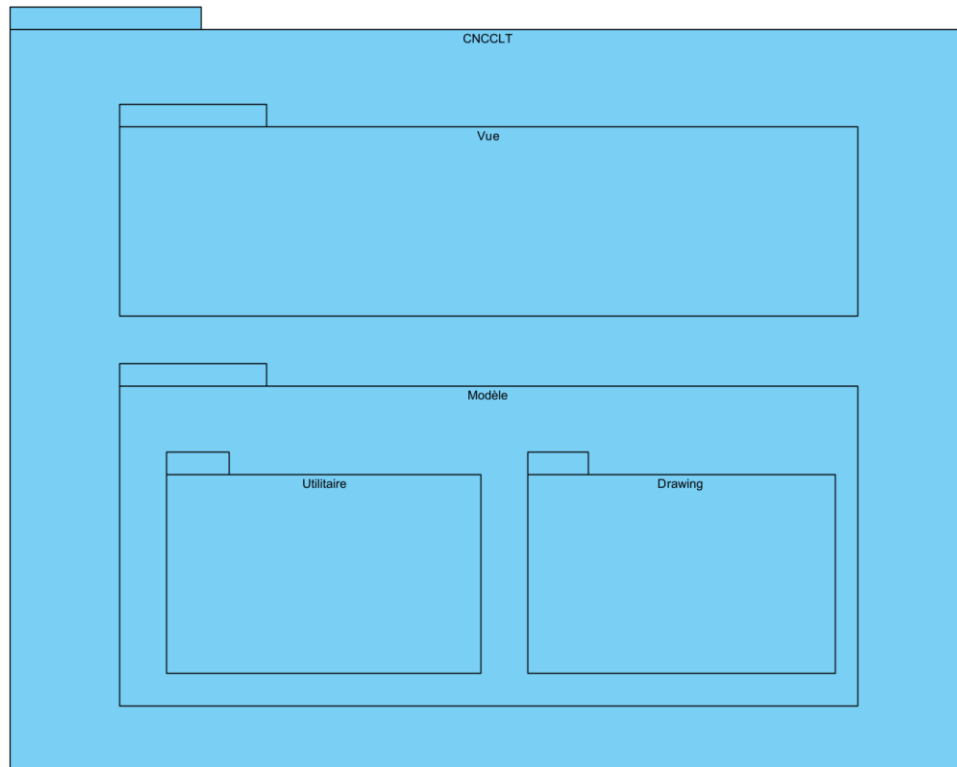
Pour faciliter l'accès aux éléments de la `CNC`, nous utilisons la classe `CNCitemDTO`. Une classe `UndoRedoManager` qui gère les opérations d'annulation (`undo`) et de rétablissement (`redo`) à l'aide de deux vecteurs : l'un pour stocker les coupes annulées dans le vecteur `undo` et l'autre pour enregistrer les coupes rétablies dans le `redo`.

Le package `utils` regroupe des classes auxiliaires essentielles, telles que la classe `Fichier`, nécessaire pour la manipulation des projets, et une classe dédiée à la description de la zone interdite.

Le package `Drawing` est utilisé par le contrôleur pour dessiner des formes, afin de générer visuellement un panneau dans l'interface utilisateur. Il est composé de la classe `Afficheur` et de ses deux sous-classes, `AfficheurHaut` et `AfficheurCote`.

Enfin, le package `GUI` contient deux classes : une énumération nommée `TypeVue`, qui permet de sélectionner la vue, et la classe `DimensionTable`, qui caractérise le `JPanel`. La classe `MainWindow` gère l'interaction avec l'utilisateur grâce à ses méthodes, permet l'accès au contrôleur, et possède divers attributs. La classe `DrawingPanel`, quant à elle, gère la création de la fenêtre interactive.

## 2. Architecture logique



Ce diagramme représente l'architecture de haut niveau du logiciel CNC composé d'une interface graphique et d'un domaine (la logique derrière l'application) qui regroupe toutes les fonctionnalités du programme. L'interface graphique est le moyen par lequel l'utilisateur interagit avec le domaine(l'application

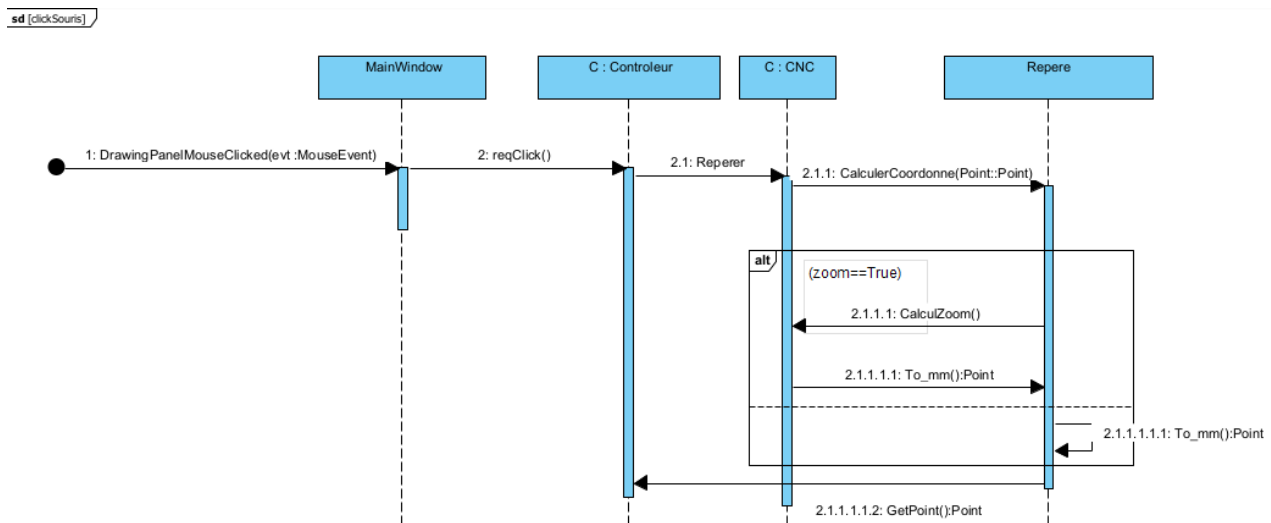
Dans le package du domaine, on a une composante utilitaire qui sert à créer les fichiers nécessaires ainsi que définir les zones interdites etc. Les classes utilitaires sont réutilisées dans les diverses parties du domaine c'est pour ça qu'on les a séparés dans leur propre package. Une autre composante nommée Drawing qui gère le traçage dans notre MainWindow et les type de vues.

Finalement dans le reste du domaine on a les diverses classes de l'application qui effectuent le travail réel

### 3. Diagrammes de séquences de conception (DSC)

#### 3.1.1 Déterminer l'élément sur lequel a lieu un clic de souris

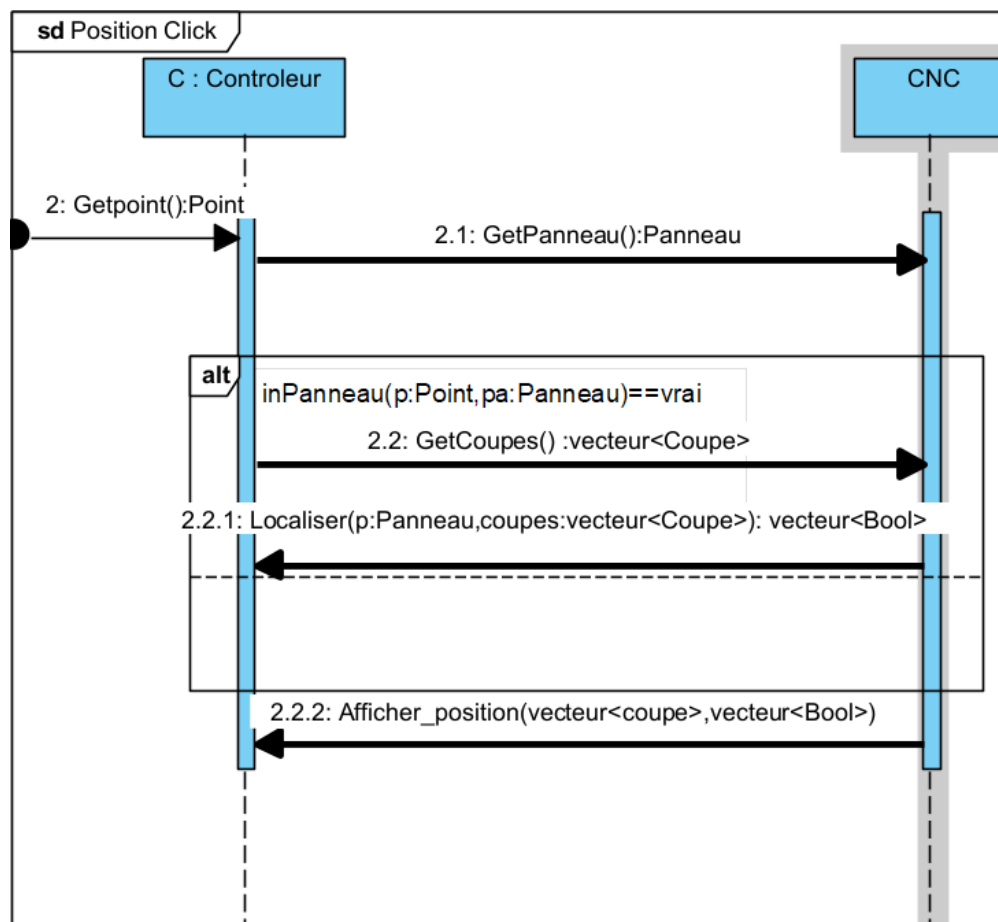
- 1- L'utilisateur fait un clic de souris dans le JPanel.
- 2- L'application Java détecte l'événement qui est le clic et nous allons faire appel à la fonction `MouseListener` dans `JPanel` pour avoir les coordonnées x et y suivant l'axe du `JPanel`
- 3- Calculer les coordonnées suivant axe du panneau qui est le coin en bas à gauche du panneau.
- 4- Dans le cas où il y a eu un zoom nous devons calculer nos nouvelles coordonnées suivant le zoom grâce à la fonction `getPreciseWheelRotation()` aussi présente dans `JPanel`.
- 5- Convertir les dimensions récupérées de pixel en millimètre.
- 6- Retourner les coordonnées en millimètre.



Dans ce processus nous allons créer un objet `JPanel` qui va nous permettre d'avoir les coordonnées en pixel ainsi que le zoom s'il y a lieu grâce aux fonctions présentes dans le `JPanel` présenté en haut. Nous allons aussi utiliser la bibliothèque `Swing (Java17)` pour la gestion d'événement. En effet cette bibliothèque va permettre le passage de l'utilisateur vers l'interface. Le zoom aussi est un événement qui va être calculé grâce à cette bibliothèque car c'est événement ainsi que le clic sur le point sont déclenchés par l'utilisateur à travers la souris.

### 3.1.2 Déterminer l'élément sur lequel a lieu un clic de souris

- 1- Le point est déjà présent dans le Contrôleur.
- 2- Appeler l'objet Panneau déjà existant afin d'avoir ces dimensions ainsi que celle de la zone interdite.
- 3- Parcourir le vecteur des coupes et vérifier si le point appartient à cette coupe.
- 4- Afficher la position de la souris (Panneau, hors panneau, coupe x, zone interdite).

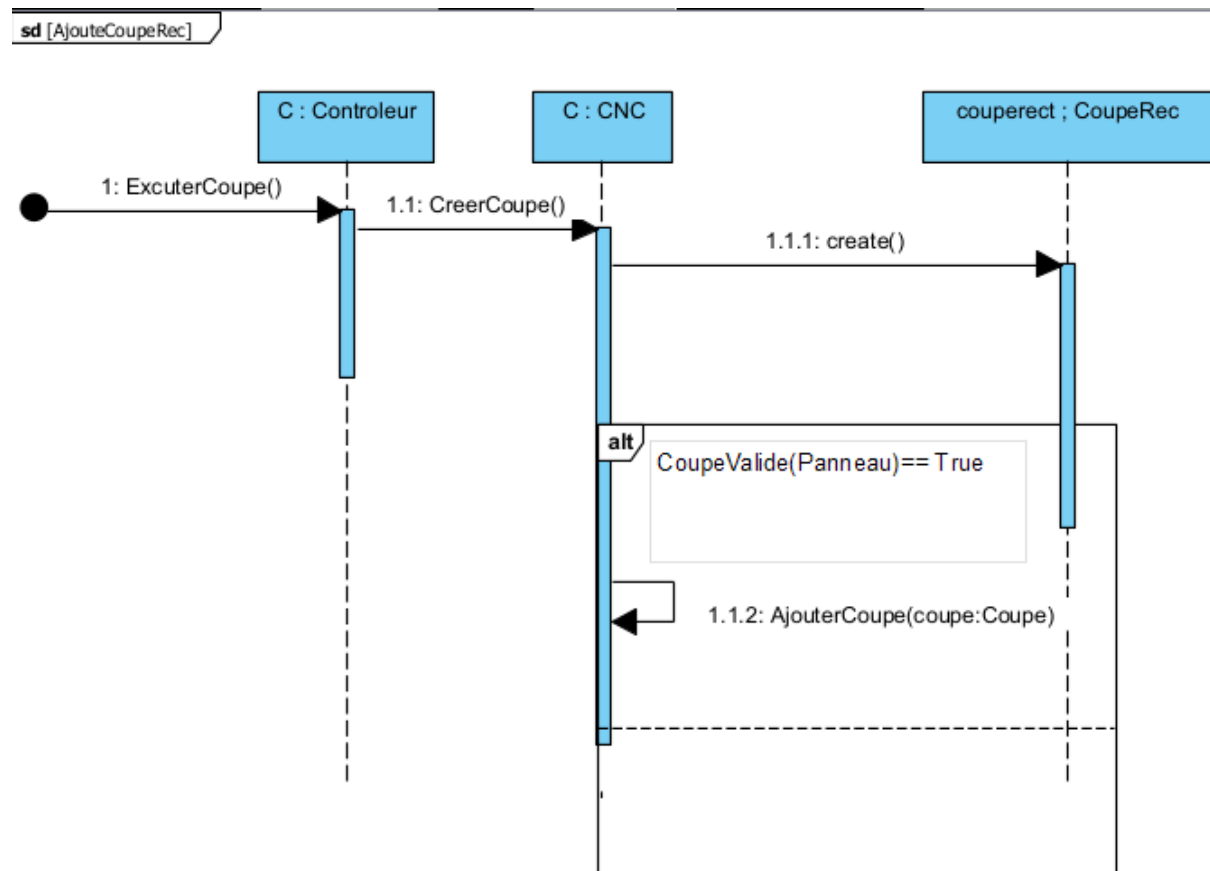


Ce diagramme peut être considéré comme une succession du précédent diagramme, nous avons un point déjà présent en millimètre et accessible grâce à la fonction Getpoint. Ce processus va comparer entre les différentes positions et dimensions des éléments de notre panneau pour savoir dans quel endroit se situe ce point. Pour cela nous allons au début voir si le point est situé dans le panneau ou dehors.

S'il est dans le panneau nous allons vérifier si on se situe dans la zone interdite puis parcourir notre vecteur coupe pour avoir la position exacte de notre point. Pour les coupes axiales nous allons avoir une droite de suivant une composante x ou y, on vérifie si une des droites passe par ce point. Dans le cas où c'est une coupe rectangulaire ou en L nous allons distinguer entre la bordure et la partie que nous allons garder (panneau). Nous affichons enfin les positions récupérées depuis la fonction localiser dans la méthode Afficher\_position.

### 3.2 L'ajout d'une coupe rectangulaire

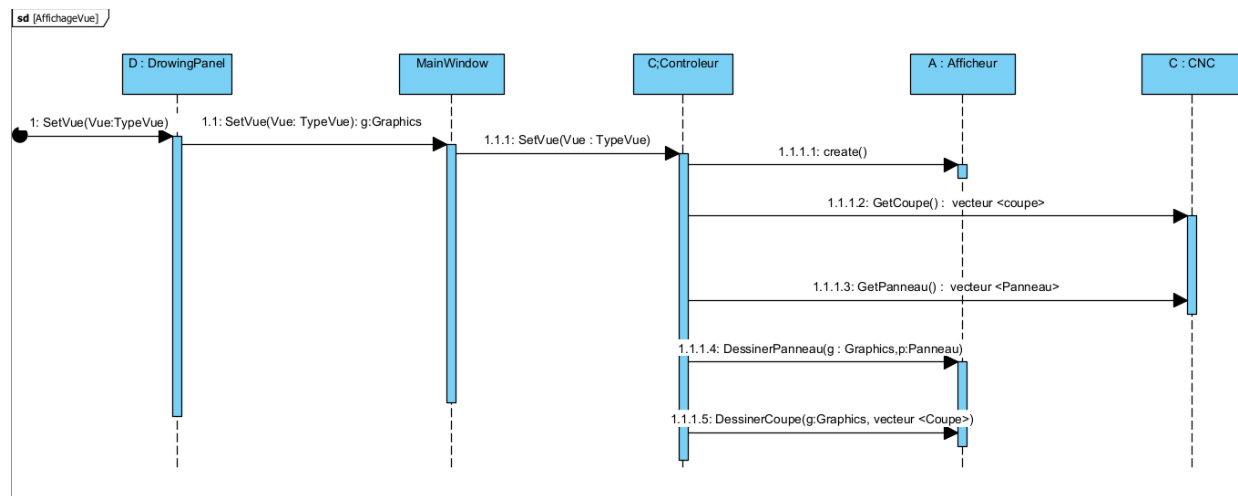
- 1-L'utilisateur choisit de faire une coupe rectangulaire
- 2-Extraire les éléments de la coupe pour la créer
- 3- Le CNC génère un objet coupe.
- 4- On valide si notre coupe ne se superpose pas avec notre zone interdite.
- 5- On ajoute notre coupe à l'ensemble des coupes.





Lors de l'exécution d'une coupe, la méthode ExecuterCoupe du contrôleur déclenche l'appel à la fonction CreerCoupe, qui reçoit les paramètres nécessaires à la définition de la coupe. Cette fonction crée ensuite un objet CoupeRec décrivant la coupe à effectuer. Une fois cet objet créé, la méthode CoupeValide intervient pour vérifier que la coupe respecte les dimensions du panneau ainsi que les zones interdites. Si la validation est réussie, la coupe est ajoutée à l'ensemble des coupes déjà planifiées.

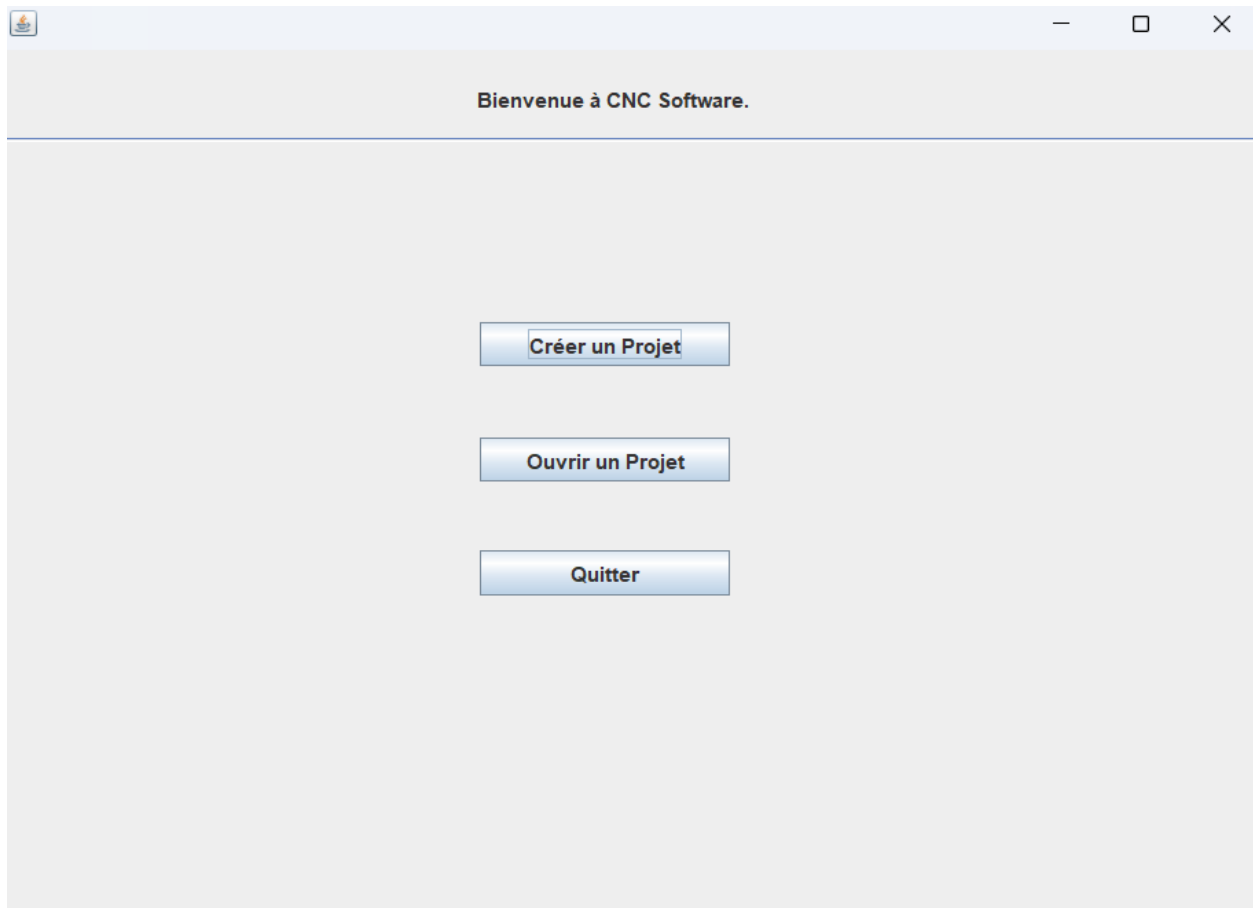
### 3.3 Affichage de la vue :

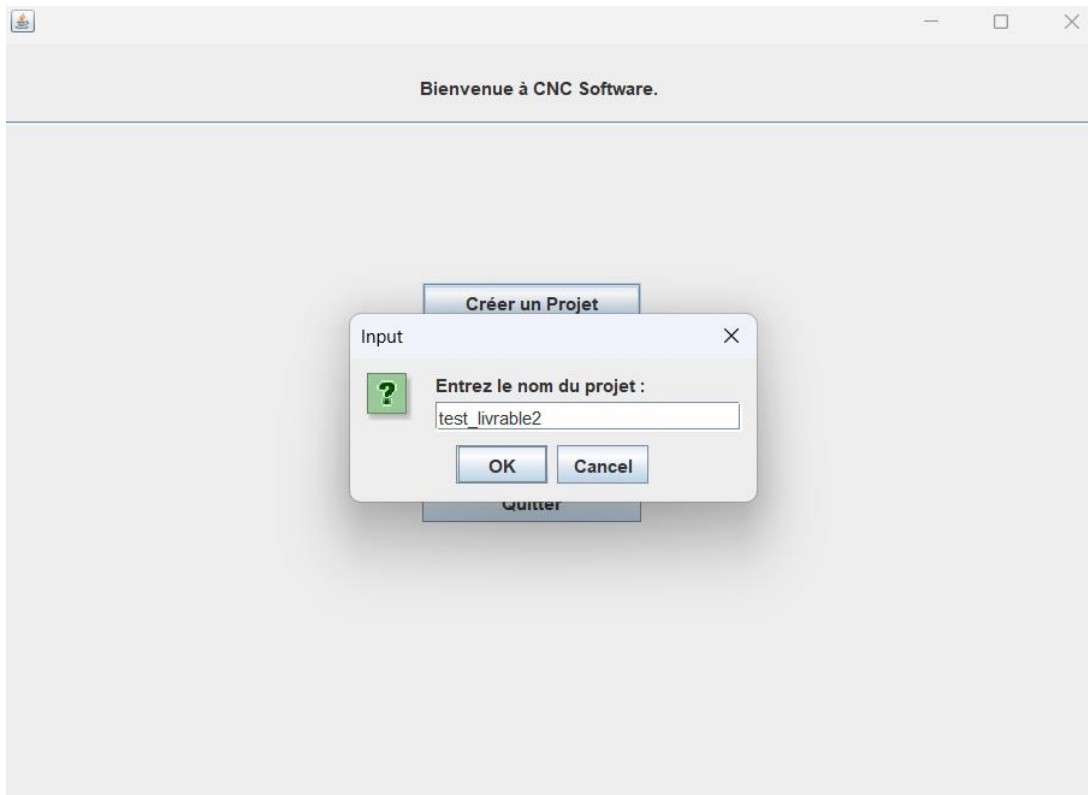


L'utilisateur sélectionne la vue souhaitée dans un menu déroulant de type JComboBox. L'événement ItemListener déclenché appelle la méthode setVue(vue: TypeVue) du DrawingPanel, qui à son tour invoque la même méthode dans la classe MainWindow. Cette dernière appelle ensuite la méthode setVue(vue: TypeVue, g: Graphics) du contrôleur de Larmann, en lui passant un objet Graphics pour permettre le rendu de l'interface graphique. Un objet Afficheur correspondant au type de vue sélectionné est alors créé. Ensuite, les coupes et le panneau sont extraits depuis la CNC. Les méthodes DessinerPanneau et DessinerCoupe sont appelées pour actualiser l'affichage en fonction de la nouvelle configuration.

## 4. Squelette de l'interface

Afin de créer le squelette de notre interface, nous avons suivi l'esquisse de la première livrable. Une fois le programme est lancé, nous pouvons créer un projet, ouvrir un projet existant ou quitter l'application. Pour créer un nouveau projet, l'utilisateur sera redirigé vers une nouvelle fenêtre, où il pourra nommer le projet et choisir son emplacement.





Une fois il écrit le nom du projet il sera redirigé vers l'étape où il configure son panneau.



Dans la configuration du panneau on peut mettre les dimensions soit en pouces soit en mm d'où la liste déroulante à cote du champ. Puis il peut cliquer sur Plot qui va représenter le

panneau (coloré dans le programme). Grace à cette représentation initiale, l'utilisateur aura une vision préliminaire de la coupe, qui est l'étape suivante.

The screenshot shows a software window titled 'Configurer votre coupe:'. The window has a menu bar with the following items: 'Projet', 'Sauvegarder', 'Undo', 'Redo', 'Modifier Panneau', 'Zone Interdite', 'Outil', and 'Exporter (GCODE)'. The main area is divided into two sections. The left section contains configuration options: 'Type de coupe:' with a dropdown menu, 'Type d'Outils:' with a dropdown menu, 'Entrez Point d'origine:' with a text input field, 'Entrez Point d'arrivée:' with a text input field, 'Entrez la Profondeur:' with a text input field, and 'Marge:' with a text input field containing '0.5' and a unit dropdown menu set to 'mm'. A yellow button labeled 'Valider Coupe' is at the bottom left. The right section is a large empty rectangular area for a preview. Below this area is the text '("X: " + cncX + ", Y: " + cncY + ", Z: " + cncZ)'. At the bottom of the window are two checkboxes: 'Grille' (unchecked) and 'Grille Magnétique' (checked).

C'est ici que le programme offre toute ces fonctionnalités, allant de montrer la grille à configurer le type de coupe à effectuer et le type d'outils à choisir. Aussi on a ajouté une barre en haut de la fenêtre pour avoir accès à plus de fonctionnalité comme sauvegarder le projet actuel, définir les outils ou la zone interdite, undo redo, etc... Nous pensons à ajouter des icones à un niveau plus avancé dans le projet pour améliorer la navigation d'un programme pas mal compliqué. Une fois l'utilisateur a tout configuré, il va voir ce message pour confirmer ou non la coupe avant de la valider sur son panneau. Finalement dans la tab Projet, l'utilisateur peut toujours quitter, ouvrir ou créer un nouveau projet.

The screenshot shows a 'Confirmation' dialog box with a close button (X) in the top right corner. It features a green square icon with a white question mark on the left. The text in the center reads 'Est-ce que vous voulez confirmer la configuration de votre coupe ?'. At the bottom, there are two buttons: 'Yes' and 'No'.

## 5. Diagrammes de Gantt

Amenallah Massarra Ouannes	
Mohamed Hedi Braham	
Zied Daly	
Fares Majdoub	
Katia Bououni	

Fonction :	S 1	S 2	S 3	S 4	S 5	S 6	S 7	S 8	S 9	S 10	S 11
UC1 : Créer un panneau											
UC2 : Visualiser un panneau											
UC3 : Définir, afficher, voir une coupe											
UC4 : Modifier/déplacer une coupe											
UC5 : Gestion du zoom autour de la souris											
UC6 : Support des valeurs numériques métriques											
UC7 : Configurer les propriétés des outils											
Tests d'intégration des fonctionnalités											
Ajustements finaux et validation											

Pas de code

Détails des itérations :

- Semaine 5-6 :

Conception, implémentation et tests de base des fonctionnalités de création (UC1) et visualisation de panneaux (UC2). Ces fonctionnalités sont les fondations de l'application.

- Semaine 7-8 :

Implémentation des fonctionnalités liées aux coupes (UC3), comme la définition, l'affichage, la sélection, et la suppression de coupes.

Modification et déplacement des coupes (UC4) seront également intégrés à cette étape, avec le panneau d'édition.

- Semaine 8-9 :

Intégration de la gestion du zoom autour de la souris (UC5) et du support des valeurs métriques (UC6), afin que l'utilisateur puisse manipuler les coupes avec précision.

- Semaine 9-10 :

Mise en place de la fonctionnalité de configuration des propriétés des outils (UC7) pour offrir plus de flexibilité aux utilisateurs.

Tests d'intégration pour vérifier que toutes les fonctionnalités précédemment implémentées fonctionnent ensemble.

- Semaine 10-11 :

Ajustements finaux et validation de l'application.

## 6. Contributions des membres

	Amen Ouannes	Mohamed Hedi Braham	Fares Majdoub	Zied Daly	Katia Bououni
DCC	40%	25%		10%	25%
Description DCC	100%				
Architecture Logique	30%	30%	40%		
DSC	30%	30%		10%	30%
Description DSC	20%	50%			30%
Interface Exécutable			75%	25%	
Description Interface			100%		
Diagrammes de Gantt				100%	
Assemblage projet	30%	30%			40%
Vérification linguistique		20%			80%