

**Enseignant - Responsable du projet :**

**Jonathan Gaudreault**

**Étudiants :**

**Marc-André Trahan**

**Maxime Prieur**

**GLO-2004 : VirtuBois**

**Livrable n°4 – Équipe #4**

Date de remise du livrable : ***30/04/2019***

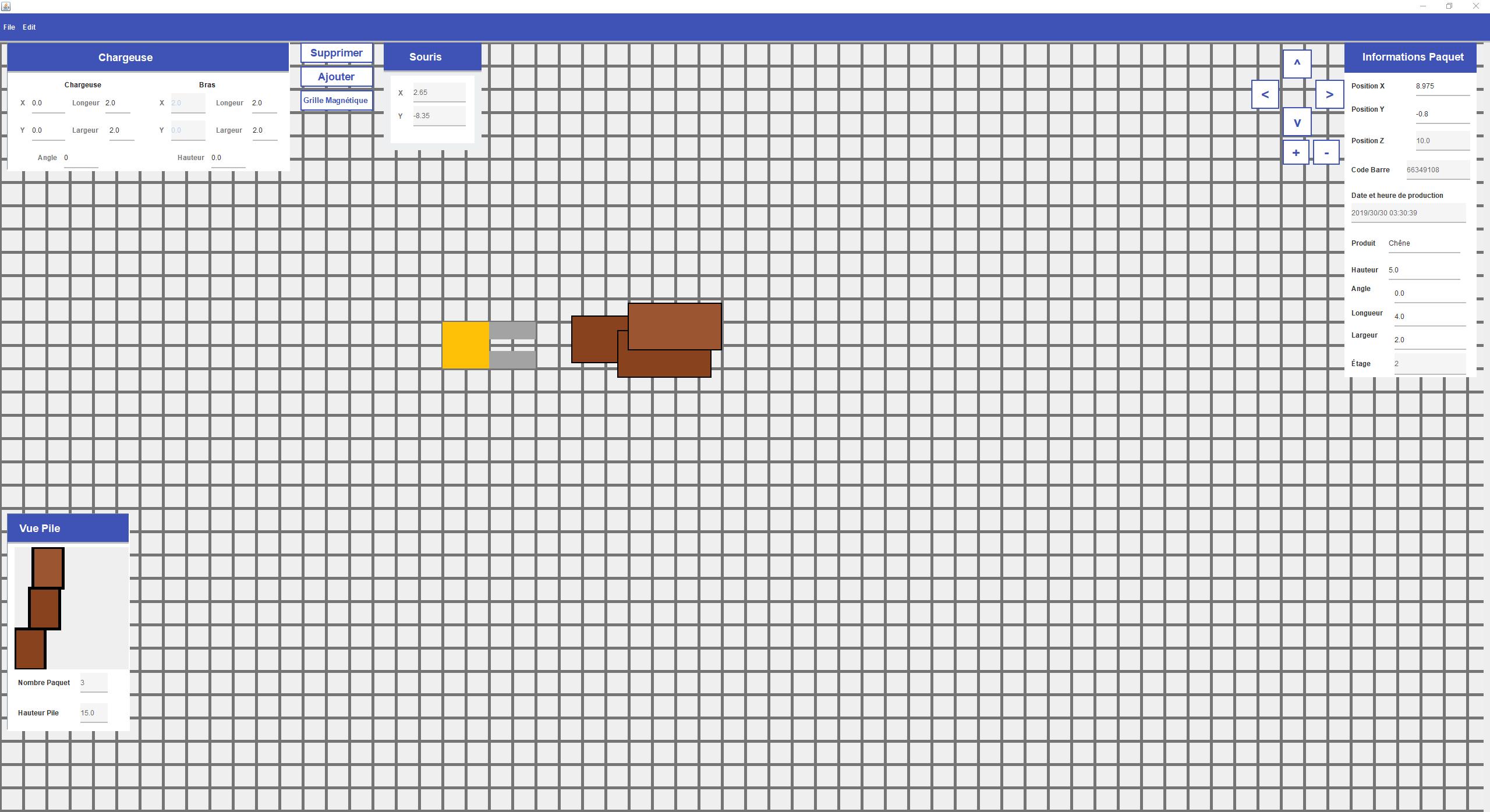
Référence du Livrable : ***GLO-2004, livrable 4, équipe 4***

# 

# Enoncé de vision

Notre équipe fut sollicitée par une scierie afin de faire la réalisation d’un projet informatisé dénommé « VirtuBois ».  L’objectif principal de cette application sera de transformer les méthodes de travail actuel, soit la gestion de la cour à bois de façon manuelle, vers une solution informatisée qui sera en mesure d’automatiser certaines opérations, de maintenir une cartographie de la cour à jour et ainsi de faciliter le travail des employés.  Afin de réaliser ce projet, l’application sera munie d’un plan cartésien qui représentera les limites physiques de la cour à bois. À l’intérieure de celle-ci se trouveront une multitude de paquets de bois, ayant des grandeurs, des tailles et des orientations différentes, réparti de manière à représenter une maquette en deux dimensions de la cour à bois vue de haut.  Étant donné que plusieurs paquets peuvent être les uns par-dessus les autres, notre solution offrira également une vue de face en deux dimensions lors de la sélection d’une pile de paquet. De surcroît, la chargeuse aura elle aussi une représentation dans le plan cartésien et il sera possible de la faire naviguer, à l’écran, afin d’effectuer les opérations que fera la réelle chargeuse, le tout en temps réel.  En addition, les bras de la chargeuse seront également contrôlables afin d’avoir la possibilité d’obtenir les paquets déposés à différentes hauteurs. Bien évidemment, des options d’ajout, de modification, de lecture d’information et de suppression seront de la partie afin de maintenir la maquette et tenir l’inventaire de la cour à bois à jour. Finalement, il sera possible de sauvegarder toutes les informations du plan dans un fichier et, par le fait même, de charger les données dans un nouveau plan via un fichier préalablement sauvegardé.

# Rendu Graphique final de l’application



# Modèle du domaine

## Diagramme

Diagramme des classes conceptuelles 1

## Texte Explicatif

La **CoursABois** peut contenir une ou plusieurs **Chargeuses** et de 0 à plusieurs **Pile**s de **Paquet**s.

La **Chargeuse** possède une **Coordonnée**, est déplaçable via l’interface utilisateur et peut posséder une **PaireDeBras** pour le chargement et le déchargement des **Pile**s de **Paquet**s

La **PaireDeBras** de la **Chargeuse** possède ses propres **Coordonnée**s à des fins d'alignement entre les **Pile**s de **Paquet**s et la **chargeuse**.

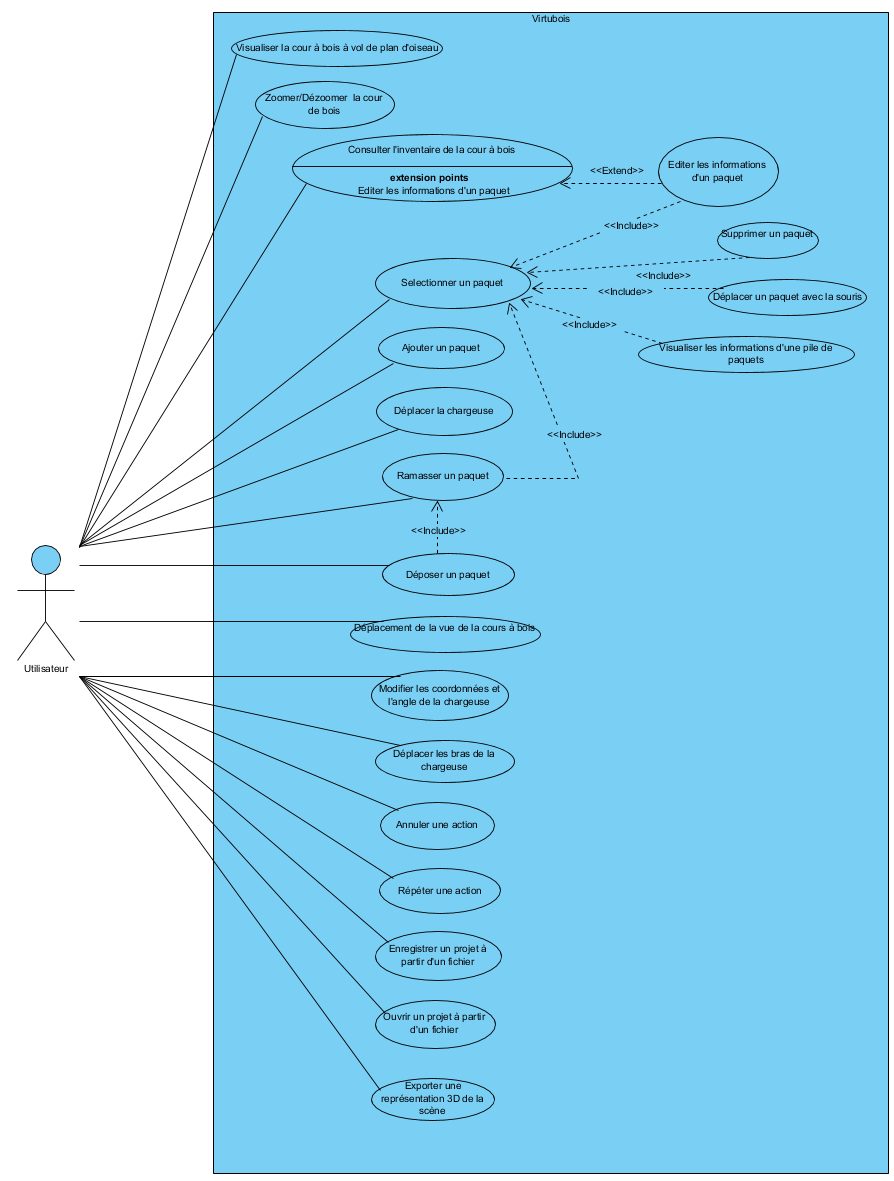
La **PaireDeBras** peut contenir une **Pile.**

Cette pile estconstituée de **paquet**s.

Les **Paquet**s possèdent leurs propres **Coordonnée**s et leurs propres informations (code barre, date et heure de production, dimension, produit contenu).

Les **Coordonnée**s de chaque élément déplaçable dans la **CoursABois** permettent de retrouver la position exacte et l'angle de chacun de ceux-ci.

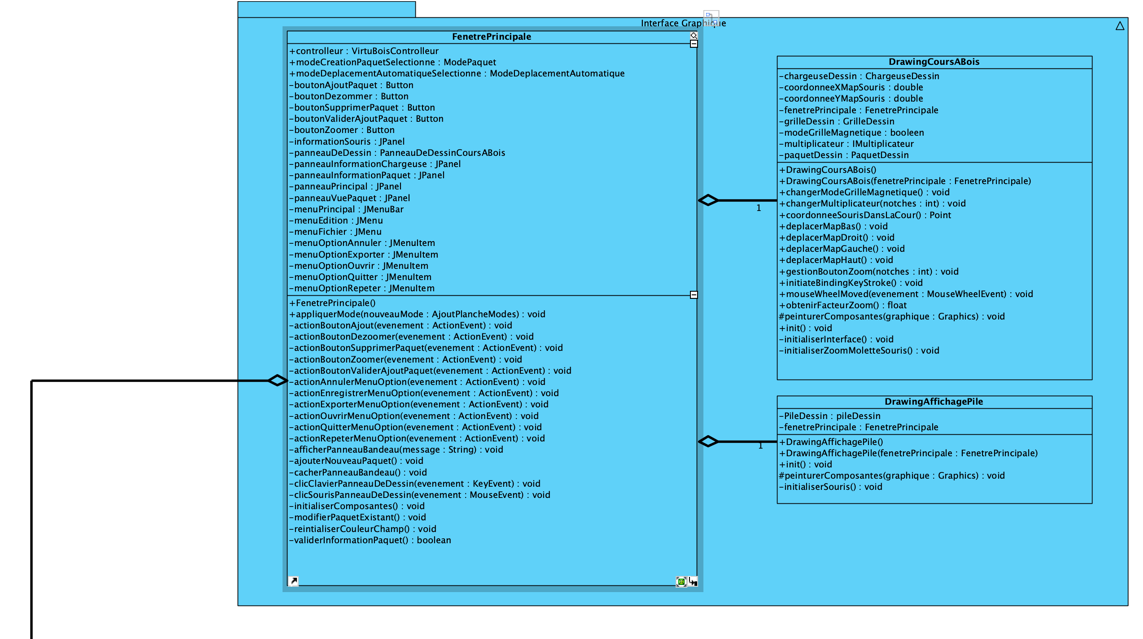
# Modèle des cas d’utilisations : diagramme des cas d’utilisations

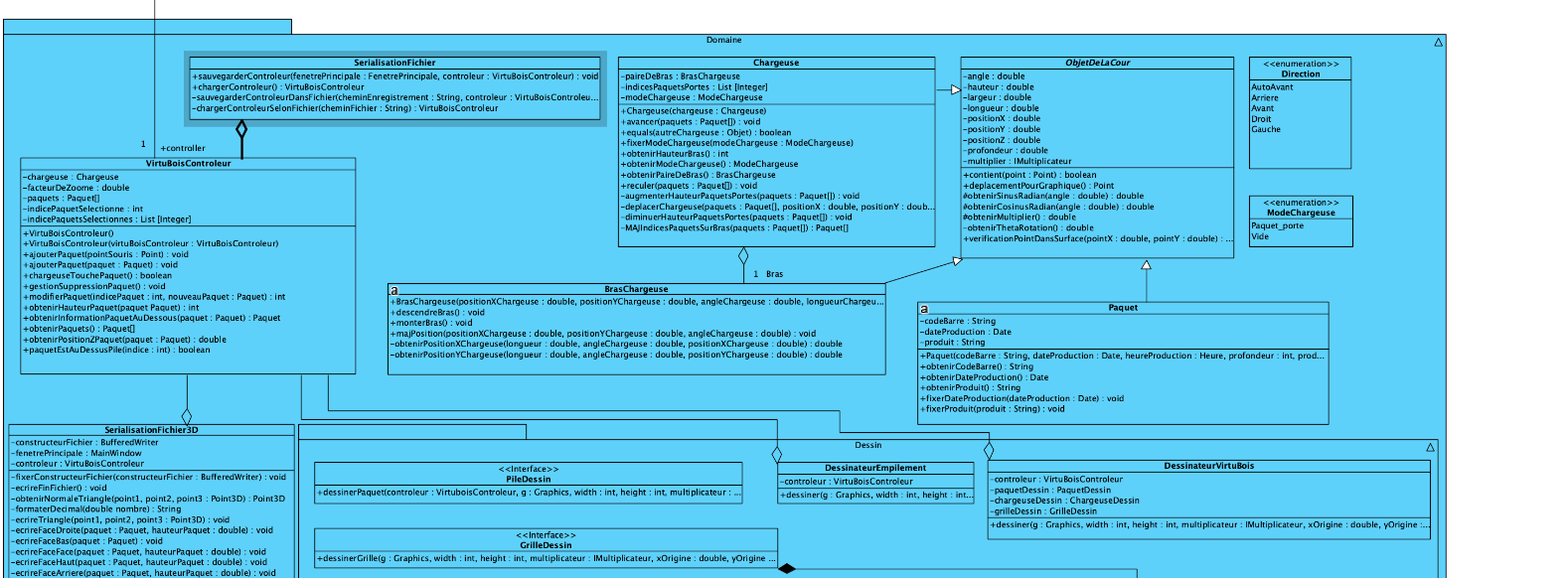


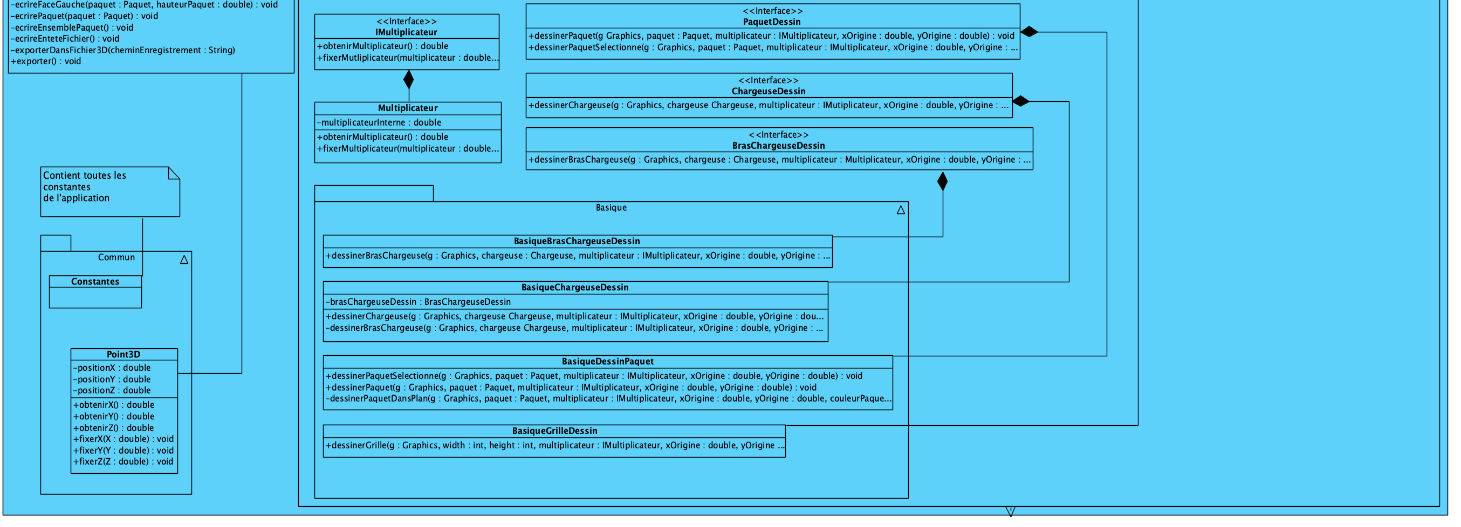
# Modèle de conception

## Diagramme de classe de conception

### Diagramme







### Texte Explicatif

L’interface graphique de l’application est décomposée en 3 sections distinctes. Tout d’abord, nous avons la **FenetrePrincipale** qui agit comme un conteneur pour les différentes sections graphique de l’application. Elle est également en charge des différentes actions en lien avec l’interface graphique, soit la barre de menu et les divers boutons. Afin de pouvoir visualiser la cour à bois en vue à vol d’oiseau, nous utilisons le **DrawingCoursABois**. Finalement, nous avons le **DrawingAffichagePile** qui permet de voir les paquets empilés en pile lors de la sélection d’une pile de paquet dans la cour à bois

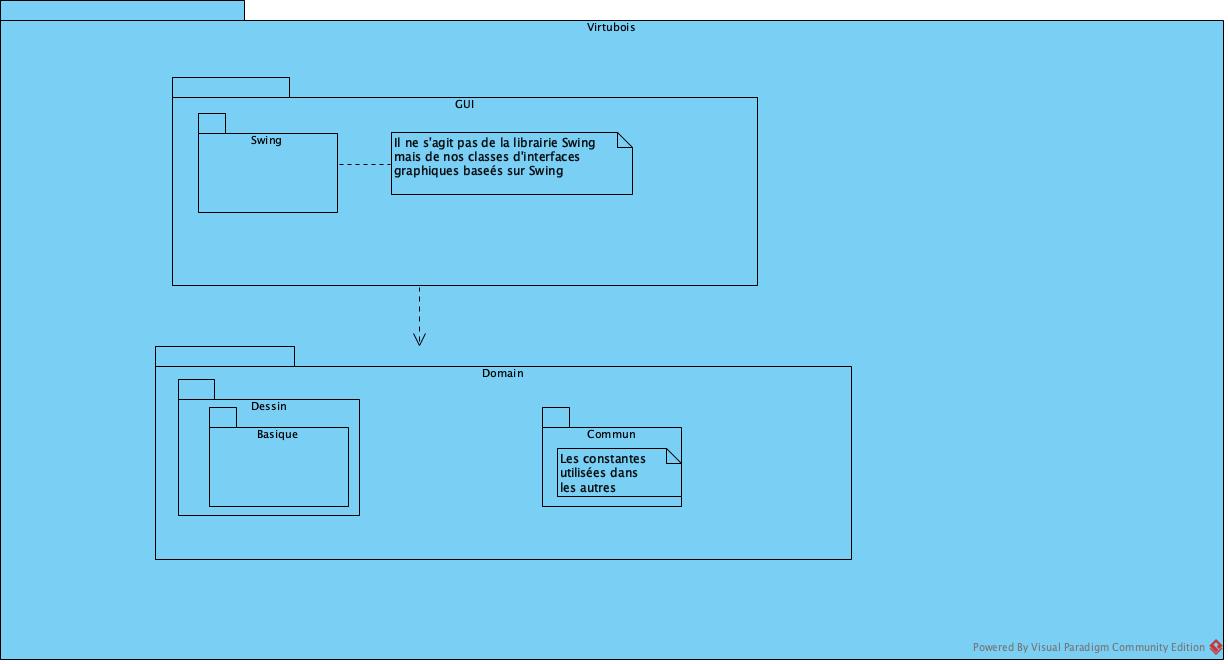
Afin de faire les communications entre la couche *Interface Graphique* et la couche *Domaine*, nous utilisons le **VirtuBoisControlleur**. Son rôle est donc d’appeler les différentes classes de la couche *Domaine* selon les actions réalisé dans l’interface visuelle.

Le package *Dessin* contenu dans le package *Domaine* permet de faire afficher la chargeuse, les paquets ainsi que la pile de paquets à tout moment. Chaque élément présent dans l’interface sera pris en compte par son propre dessinateur et chaque dessinateur implémente une interface. Donc, pour les paquets, c’est **BasiquePaquetDessin** qui fera l’affichage et elle implémentera **PaquetDessin** comme interface. Pour la chargeuse, ce sera **BasiqueChargeuseDessin** qui fera le dessin et implémenteral’interface **ChargeuseDessin**. De plus, la classe **BasiqueChargeuseDessin** s’occupera de faire afficher la paire de bras de la chargeuse. En effet, c’est au rôle de la chargeuse à dessiner toutes ces propres composantes car si nous voulons, par exemple, ajouter une seconde chargeuse dans notre plan de cours, il faudrait uniquement dessiner une seconde chargeuse et sa paire de bras sera dessiner automatiquement. Par la suite, nous avons **BasiqueGrilleDessin** qui implémente l’interface **GrilleDessin** et qui se charge d’afficher la grille du plan de cours. Nous avons également une classe **Multiplicateur** et une interface **IMultiplicateur** qui nous permet de gérer le zoom et de toujours afficher les paquets et la chargeuse à l’échelle. Afin d’afficher l’empilement des paquets, nous avons à faire à la classe **DessinateurEmpilement** et pour faire afficher les diverses objets dans le plan de cours, nous utilisons la classe **DessinateurVirtuBois.**

Nous avons un deuxième package dans le package *Domaine* qui se nomme *Commun*. Nous avons présentement uniquement une classe à l’intérieur, **Constantes**, et elle nous permet de rassembler toutes nos constantes dans notre application au même endroit.

La classe **CoursABois** est la classe contenant la liste des paquets ainsi que les actions nécessaires afin d’ajouter, modifier ou supprimer un **Paquet** de la cour à bois. Les informations relatives aux paquets sont dans la classe **Paquet**. Cette dernière hérite de la classe **ObjetDeLaCour** afin de pouvoir savoir ces coordonnées x, y, z ainsi que l’angle la hauteur et la longueur. Nous avons deux autres classes qui héritent de celle-ci soit la **Chargeuse** ainsi que **PaireDeBras**. La **Chargeuse** permet de déplacer la chargeuse dans la cour à bois et à tester les collisions avec les paquets. La **PaireDeBras** sert à sélectionner un ou des paquets à l’aide de la chargeuse.

## Architecture Logique



L’architecture logique de notre système se compose en deux couches.

* **La couche GUI** qui représente l’interface utilisateur. Cette couche est composée de nos classes d’interfaces graphiques basées sur Swing.

**La couche Domain** qui représente notre logique applicative et les objets. Elle est composée de nos classes publiques, d’un package Dessin et du package « Commun ». Le package dessin est notre dessinateur. Il contient toutes les classes qui nous permettent de dessiner nos objets. Le package Commun contient globalement une interface de constantes pour éviter le hard codage.

# Conclusion

Nous voilà rendu à la fin de ce projet. L’application aboutis répond ainsi à l’ensemble des caractéristiques demandées par le client. Il faut pourtant faire attention au fait que les lois de la physique n’ont pas été complétement respectées.

En effets la loi de la gravité qui ferait que des paquets ne pourrait pas être porté sans risque de s’effondrer n’a pas été prise en compte. Les limites physiques des bras de la chargeuse laissent également à désirer puisque ceux-ci n’ont grossièrement pas de de hauteur et sont selon notre modèle complètement plats. Il aurait été utile de spécifier cette dimension mais aurait demandé plus de précision quant à la façon de superposer les paquets les uns sur les autres. Les bras peuvent aussi monter à l’infini ce qui bien qu’idéal est peut-être trop extravagant.

Les différents affichages pourraient très bien être amélioré comme par exemple l’affichage de pile de paquets puisque nous avons eu un bref aperçus d’autre groupe ayant obtenu un rendu réussi de cet affichage en vue 3D. Malheureusement la main d’œuvre dont nous disposions était peut-être trop juste pour couvrir l’ensemble des aspects de l’application.

Pour un contexte industriel, il serait utile d’y ajouter un répertoire des paquets présents dans la cour qui permettrait de gérer les stocks plus facilement. Il faudrait bien sur permettre l’implémentation / la connexion à un GPS qui donnerait les informations de la chargeuse en temps réel.

Enfin concernant le projet. Ce développement entier d’une application en équipe a été très riche en expériences, tout d’abord sur la façon d’avancer dans son élaboration et sa construction mais aussi pour ce qui est du travail en équipe et de la communication au sein d’un groupe.

# Contribution de chacun des membres de l’Équipe

|  |  |
| --- | --- |
| **Tâche** | **Nom du membre de l’équipe** |
| Énoncé de vision | Marc-André Trahan |
| Rendu graphique de l’application | Marc-André Trahan |
| Modèle du domaine à jour | Maxime Prieur |
| Modèle des cas d’utilisation | Maxime Prieur |
| Architecture Logique | Maxime Prieur |
| Diagramme de classe de conception | Marc – André Trahan |
| Conclusion | Maxime Prieur |