

**Enseignant - Responsable du projet :**

**Jonathan Gaudreault**

**Étudiants :**

**Guillaume Genest-Lebel Marc-André Trahan**

**Julienne Petronie Betsama Assolo Maxime Prieur**

**GLO-2004 : VirtuBois**

**Livrable n°3 – Équipe #4**

Date de remise du livrable : ***26/02/2019***

Référence du Livrable : ***GLO-2004, livrable 3, équipe 4***

# 

# 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**.

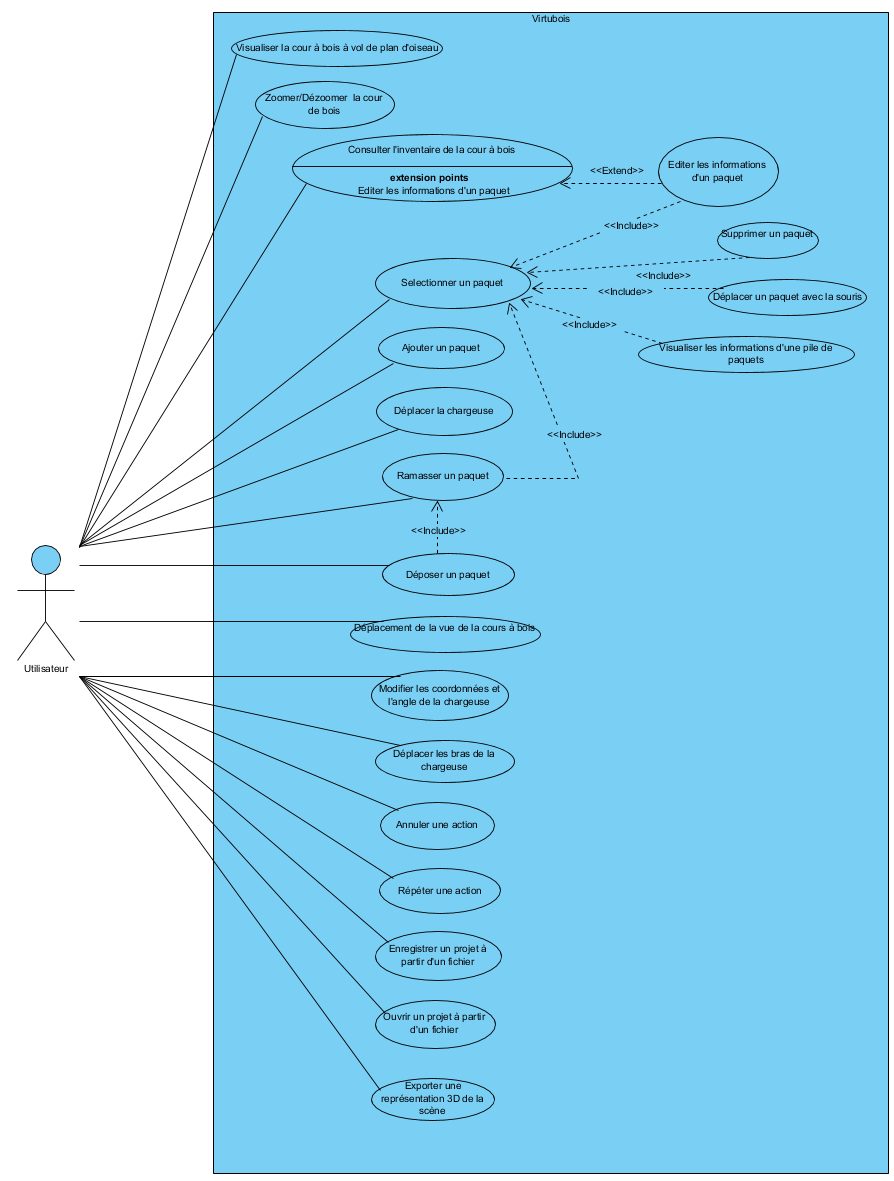
La **PaireDeBras** peut contenir une **Pile.**

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.

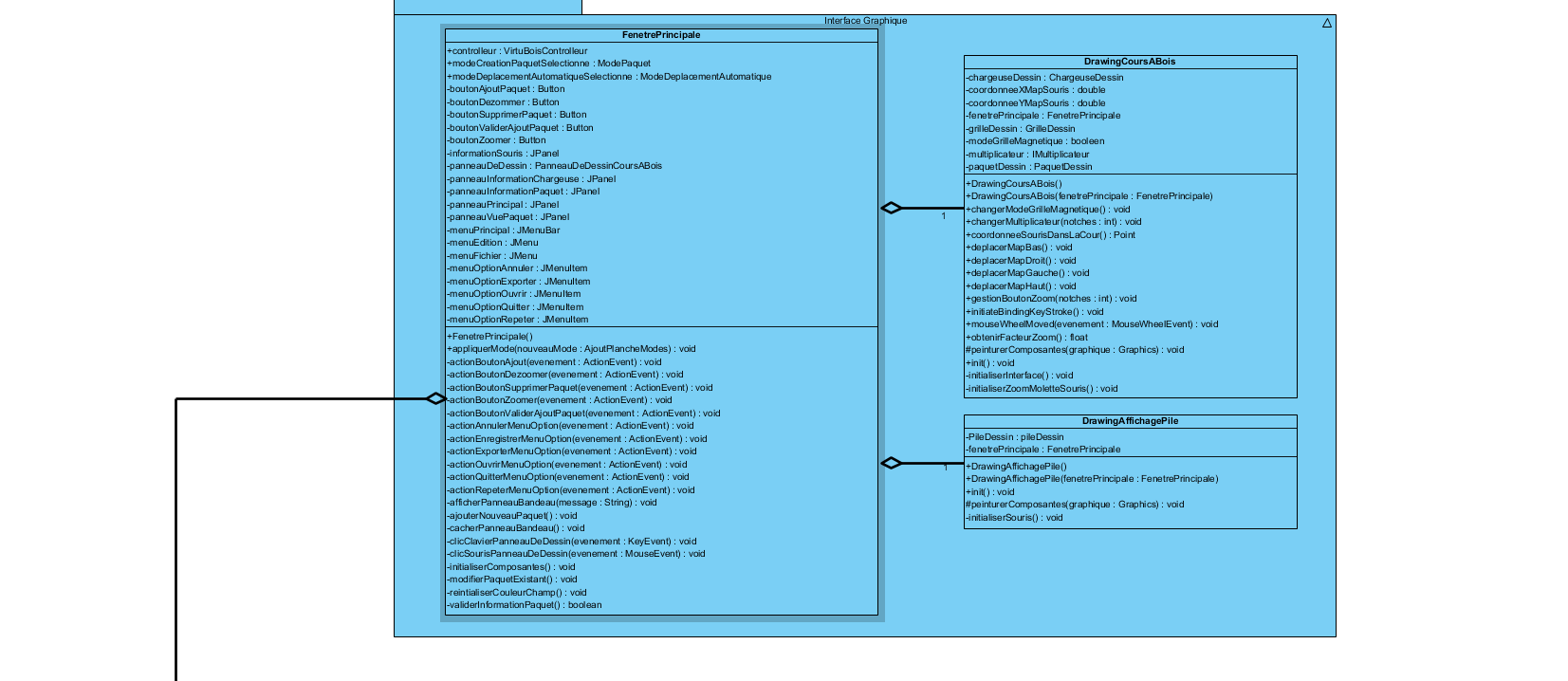
# Modèle des cas d’utilisations : diagramme des cas d’utilisations

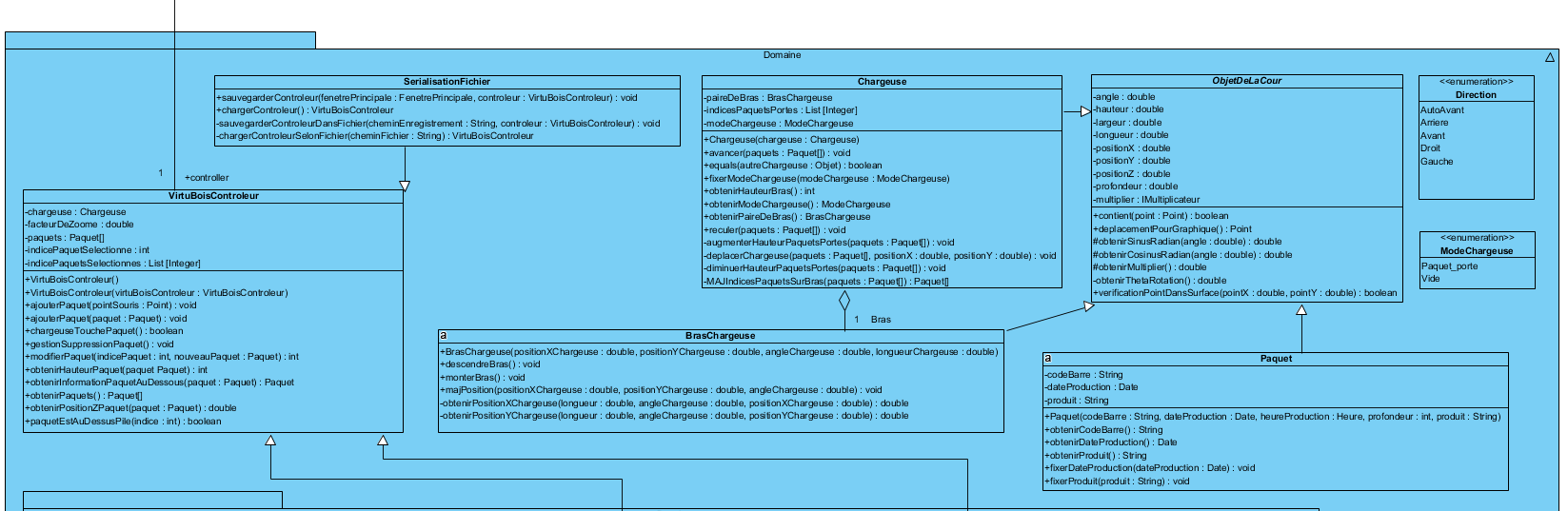


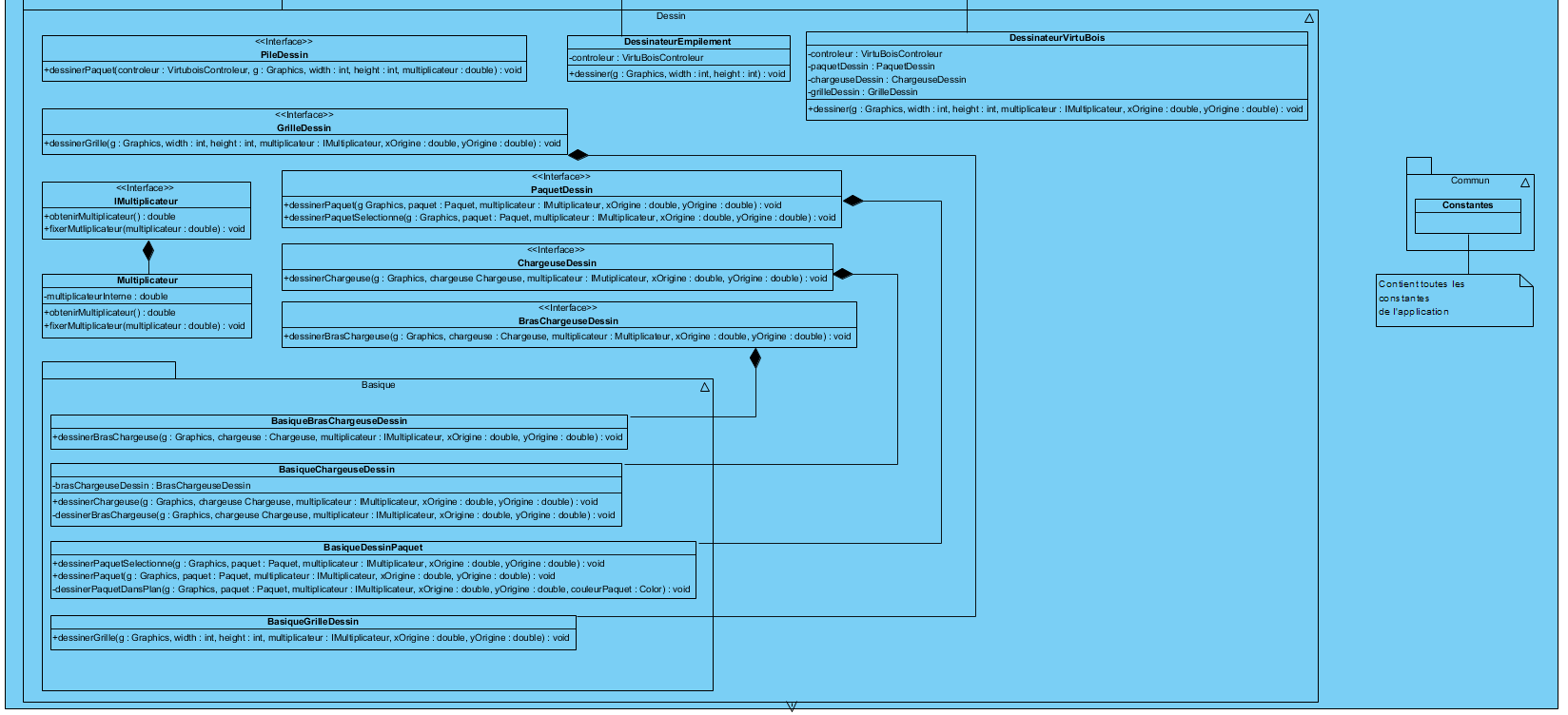
# Modèle de conception

## 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

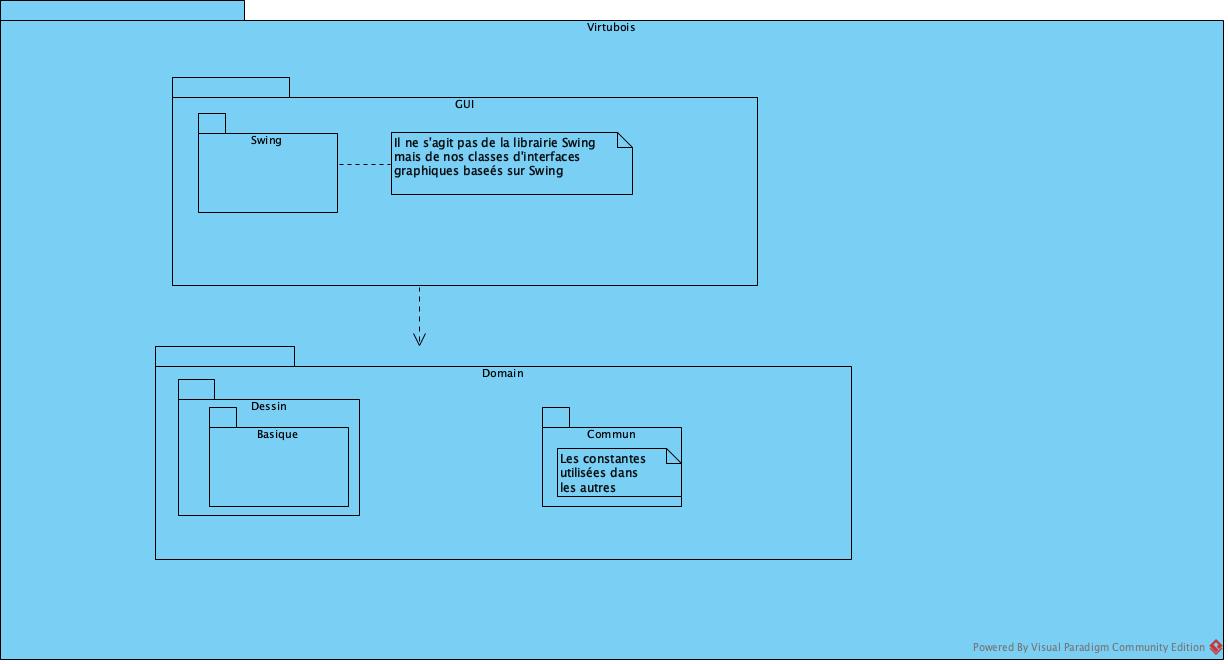
Afin de faire les communications entre la couche *Interface Graphique* et la couche *Domaine*, nous utilisons le **VirtuBoisControlleur**. Son rôle est donc d’appeler les différentes classes de la couche *Domaine* selon les actions réalisé dans l’interface visuelle.

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 eviter le hard codage.

## Diagramme de séquence de conception

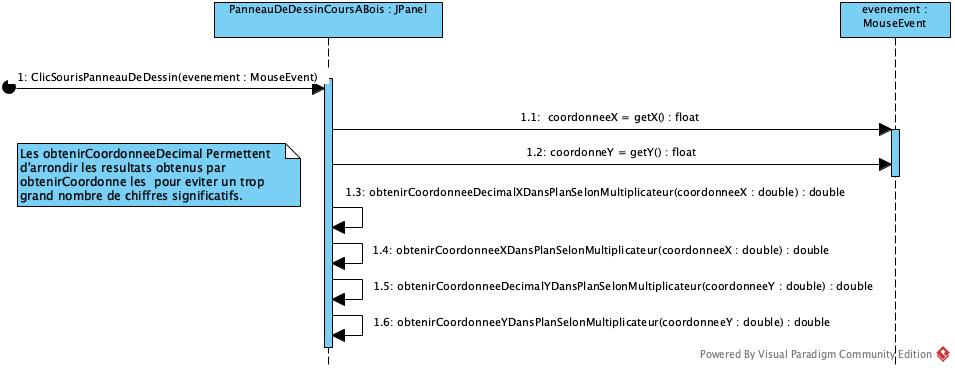


Diagramme de séquence de conception pour déterminer les coordonnées du point cliqué par l'utilisateur

L’utilisateur en cliquant sur la cour à bois lance un appel au JPanel avec comme paramètre un objet MouseEvent. Le JPanel appelle récupère alors les coordonnées x et y stockées dans le MouseEvent. La méthode ClicSourisPanneauDeDessin appelle ensuite successivement obtenirCoordonneeDecimalXDansPlanSelonMultiplicateur et obtenirCoordonneeDecimalYDansPlanSelonMultiplicateur. Dans ces 2 méthodes on appelle obtenirCoordonneeXDansPlanSelonMultiplicateur nous renvoyant les coordonnées transformées pour correspondre à ce qui est affiché dans le drawingPanel en utilisant le multiplicateur et la différence entre le point d’origine (coin supérieur gauche de la fenêtre) avec les coordonnées de la souris. Les méthodes tronquent ensuite les résultats pour ne pas afficher trop de chiffres significatifs.

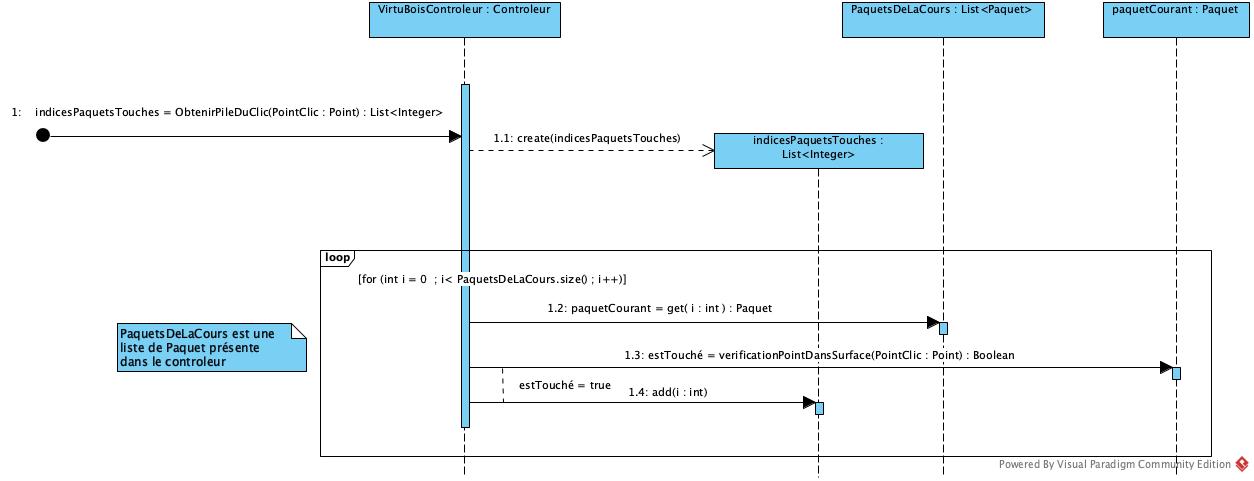


Diagramme de séquence de conception pour déterminer la pile sélectionnée par un clic de souris

Le contrôleur reçoit les coordonnées d’un point (x, y) par un appel afin de fournir les indices permettant de retrouver les paquets touchés dans la liste des paquets de la cours (ensemble de paquet contenant le point cliqué). On renvoie une liste d’indice vide si l’endroit cliqué par l’utilisateur est vide.

Le contrôleur créer une Liste d’entier : indicesPaquetsTouches. Il itère ensuite sur tous les paquets de la cour à bois. Pour chacun des paquets on appelle la fonction vérifiant si le point est situé dans le paquet. Si c’est le cas, on ajoute l’indice du paquet à indicesPaquetsTouches qui sera retourné une fois l’itération terminée.

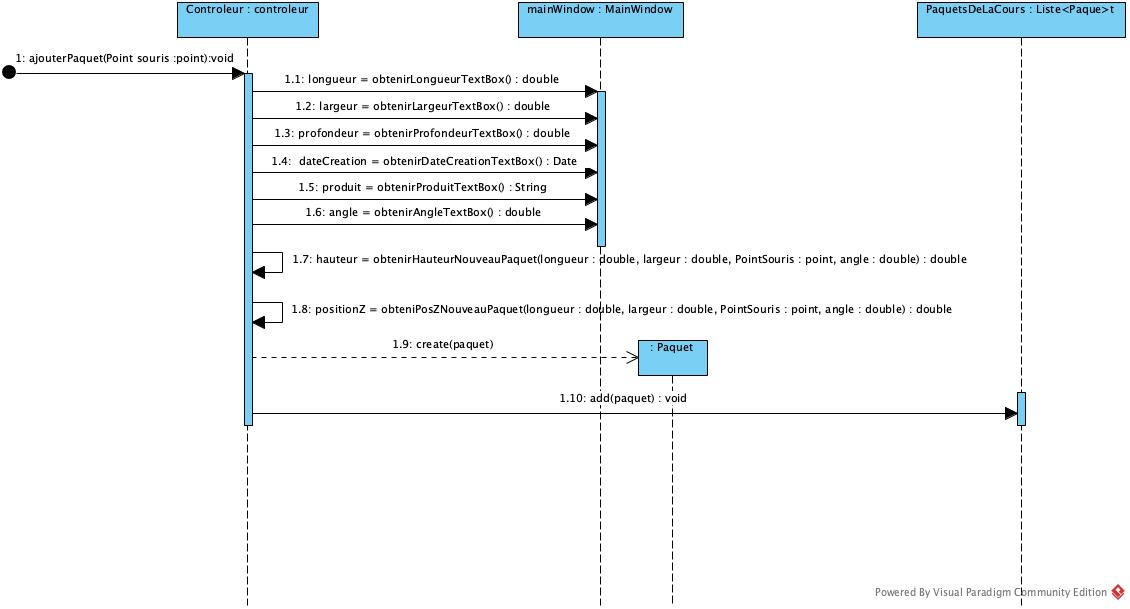


Diagramme Séquence de conception de création d'un nouveau paquet

L’utilisateur clique sur le bouton Ajout Paquet, il va ensuite cliquer sur le plan d’affichage. Le système va prendre les coordonnées de point souris (x,y) et renvoyer le tout au contrôleur. Le contrôleur va récupérer l’ensemble des paramètres ayant été entrés par l’utilisateur et étant nécessaires à l’ajout d’un paquet. Le paquet sera créé et sera ajouté à la liste de paquets.

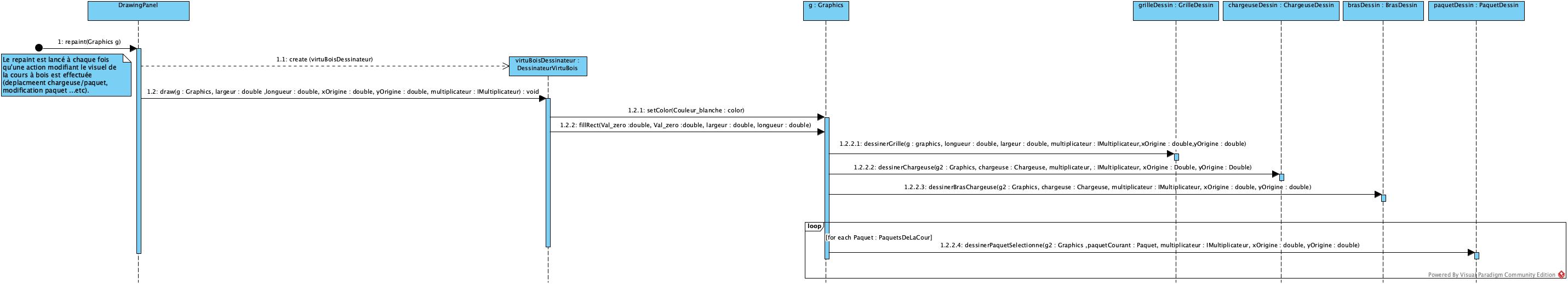
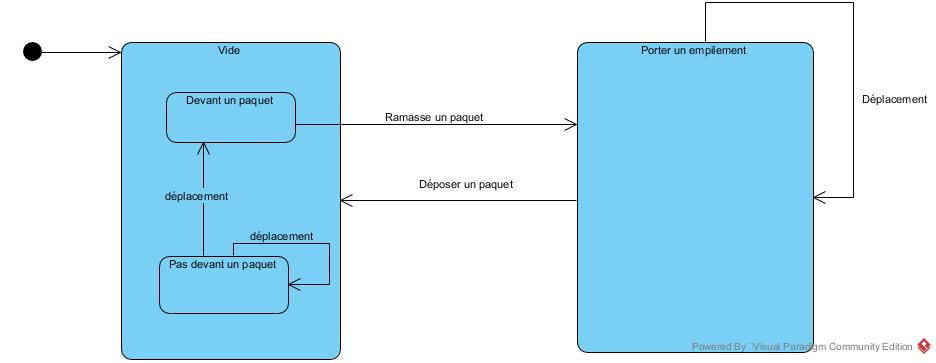


Diagramme séquence de conception Affichage de la cours à bois

A chaque fois qu’une action modifiant l’état visuel de la cour est effectuée, on appelle la méthode repaint du drawingPanel de la cour.

Dans cette fonction paintComponent()/repaint(), on instancie un objet DessinateurDeLaCours. Puis on appelle sa méthode dessinerCour() avec les paramètres nécessaires. Dans cette methode dessinerCour() on remplis d’abord la zone d’affichage par une couleur blanche. Ensuite, on appelle les méthodes dessiner de chaque dessinateur d’objet de la cour à bois (Quadrillage, Chargeuse, Bras, Paquet). (On appelle la méthode dessinerPaquet() pour tous les paquets de la cour).

# Diagramme d’état de la chargeuse



La chargeuse a deux états principaux

* L’état « vide » qui possède deux sous-états : « devant un paquet » et « pas devant un paquet ». Pour transiter entre les deux sous états, la chargeuse doit se mettre en déplacement.
* L’état « Porte un empilement ». Lorsque la chargeuse ramasse un paquet, elle passe de l’état « vide » à l’état « porte un empilement ». De même lorsqu’elle dépose un paquet, elle passe de l’état « porte un empilement » à l’état «  vide ».

# Contribution de chacun des membres de l’Équipe

|  |  |
| --- | --- |
| **Tâche** | **Nom du membre de l’équipe** |
| Mise à jour du modèle domaine |  |
| Mise à jour diagramme des cas d’utilisation | Petronie Betsama |
| Mise à jour du modèle de conception |  |
| Architecture Logique | Maxime Prieur |
| Diagramme de classe de conception | Marc – André Trahan |
| Diagrammes 1,2,3 et 4 de séquence de conception | Maxime Prieur |
| Diagramme d’état de la chargeuse | Petronie Betsama |