



UNIVERSITÉ
LAVAL

Faculté des sciences et de génie

GLO-2004 : VirtuBois

Livrable n°2 – Équipe #4

Étudiants :

Guillaume Genest-Lebel

Julienne Petronie Betsama Assolo

Marc-André Trahan

Maxime Prieur

Enseignant - Responsable du projet :

Jonathan Gaudreault



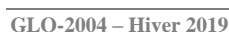
UNIVERSITÉ
LAVAL

Faculté des sciences et de génie

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

Référence du Livrable : ***GLO-2004, livrable 2, équipe 4***

1.1. Diagramme





1.2. Texte Explicatif

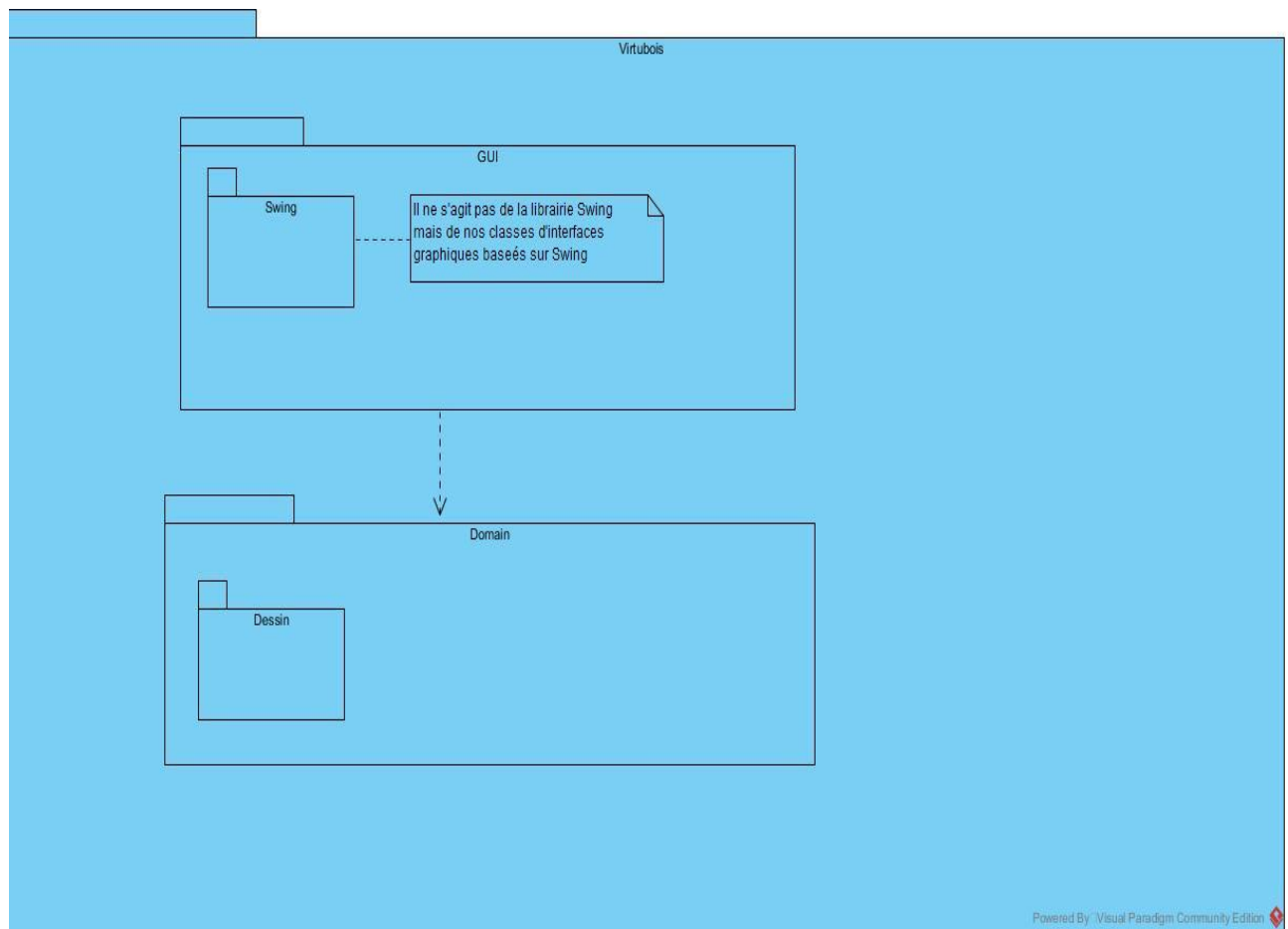
L'interface graphique de l'application est décomposée en 4 sections distinctes. Tout d'abord, nous avons la **FenetrePrincipale** qui agit comme un conteneur pour les différentes sections graphique de l'application. Elle est également en charge des différentes actions en lien avec l'interface graphique, 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 **PanneauDeDessinCoursABois**. Ensuite, nous avons le **PanneauDeDessinVuePile** qui permet de voir les paquets empilés en pile lors de la sélection d'une pile de paquet dans la cour à bois. Finalement, le **PanneauDeDessinInformationPaquet** contient les informations du paquet présentement sélectionné.

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 réalisés 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. Les deux premiers seront pris en charge par **VirtuBoisDessin** alors que la pile de paquets sera dessinée par **PileDessin**.

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.

2. 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 et d'un package Dessin. Le package dessin est notre dessinateur. Il contient toutes les classes qui nous permettent de dessiner nos objets.

3. DIAGRAMMES DE SÉQUENCE DE CONCEPTION

3.1. Déterminer l'empilement (et les paquets) à sélection lors d'un clic de souris dans la vue en plan

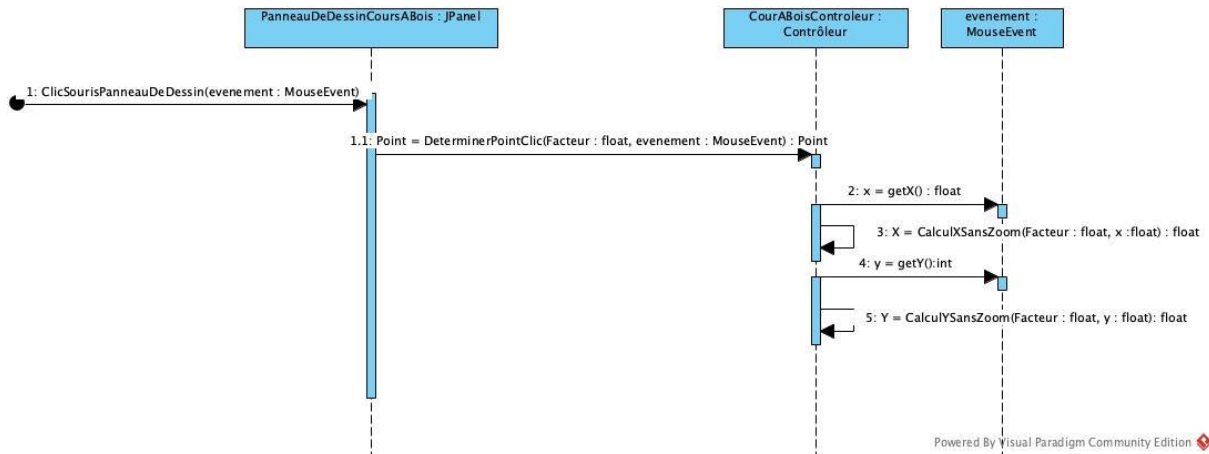


Diagramme de séquence de conception pour déterminer les coordonnées du point cliqué par l'utilisateur

L'utilisateur en cliquant sur la pile affichée dans le Panneau de dessin de la Cours à bois lance un appel au JPanel avec comme paramètre un objet MouseEvent. Le JPanel appelle alors la méthode du contrôleur DeterminerPointClic qui va renvoyer les coordonnées du point accéder par l'utilisateur et le facteur de zoom. Le contrôleur récupérer le point x obtenu par l'évènement puis le ramène à l'échelle normale (sans le zoom) et y ajoute les coordonnées de la chargeuse par la méthode CalculXSansZoom. De même pour la valeur Y.

3.2. Déterminer l'empilement (et les paquets) à sélection lors d'un clic de souris dans la vue en plan à partir du contrôleur

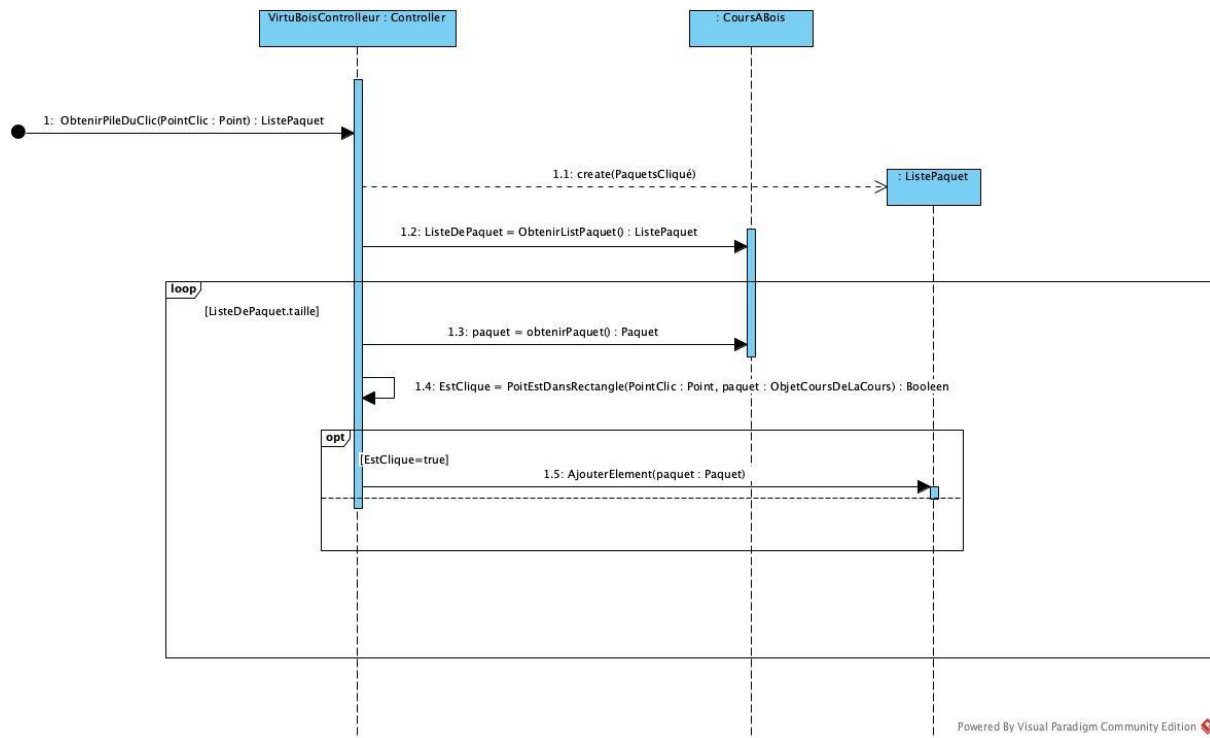


Diagramme de séquence de conception pour déterminer la pile sélectionnée par un clic de souris

Le contrôleur reçoit les coordonnées d'un point (x, y) par un appel afin de fournir les informations de la pile/sélection (ensemble de paquet contenant le point cliqué) située à l'endroit du clic de souris. On renvoie une pile vide si l'endroit cliqué par l'utilisateur est vide.

Le contrôleur crée une Liste de paquet : Paquetscliqué. Il récupère ensuite tous les paquets existants qui appartiennent à la classe CoursABois. On crée une itération sur chacun des paquets. Dans cette itération, on appelle la fonction vérifiant si le point est situé dans le paquet de l'itération. Si c'est le cas, on ajoute le paquet à Paquetscliqué qui sera retourné une fois l'itération terminée.

3.3. Création d'un nouveau paquet

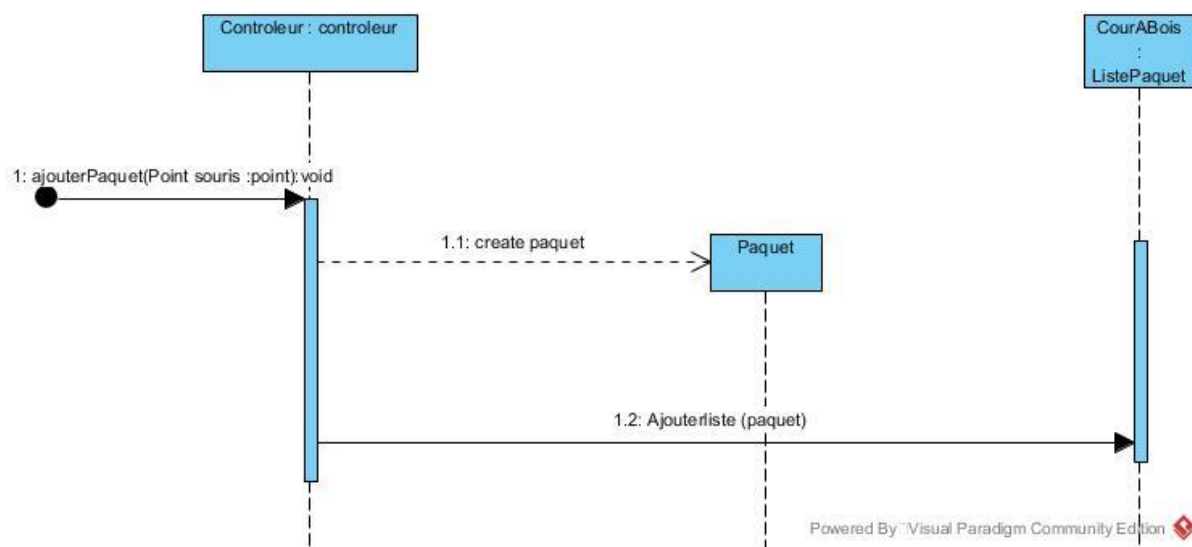


Diagramme Séquence de conception de création d'un nouveau paquet

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 demander l'ajout d'un paquet à la liste de paquets. Le paquet sera créé et sera ajouté à la liste de paquets.

3.4. Affichage de la vue en plan de la cour

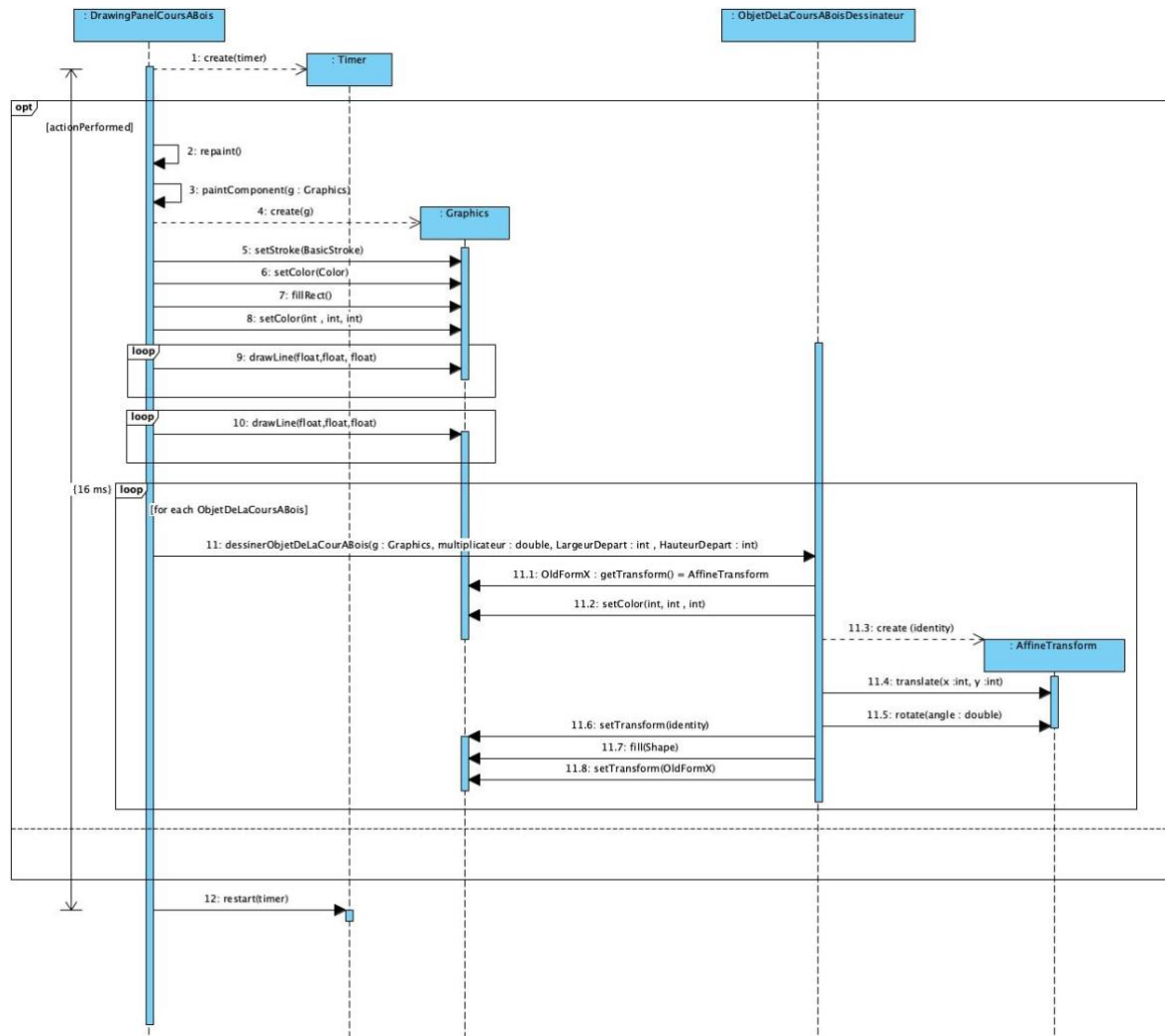


Diagramme séquence de conception Affichage de la cours à bois



Lors de la création du `DrawingPanelCoursABois` un `Timer` est créé. Ce `timer` sur une courte durée vérifie si une action a été effectuée. Si c'est le cas, le `drawingPanel` lance la fonction `repaint()`. Cette fonction permet d'effectuer à nouveau la fonction `paintComponent()`. On permet ainsi d'afficher de nouveau l'entièreté de la cour et de ses éléments, ceux-ci ayant été mis à jour lors de la prise en compte de l'action effectuée par l'utilisateur.

Dans cette fonction `paintComponent()`, on crée d'abord un objet `Graphics`, `g`. Cet objet va contenir l'ensemble de nos éléments à dessiner (ligne, objet avec leur forme couleur, taille...). On place une forme dans l'objet `g` avec la méthode `setStroke()`. On sélectionne ensuite une couleur de dessin avec la méthode `setColor()`. On remplit la forme précédemment construite de la couleur précédemment instanciée (on a ainsi rempli notre cours d'un arrière-fond blanc). On réappelle la méthode `setColor()` puis on effectue deux boucles pour dessiner des lignes verticales et horizontales qui constitueront notre « Grille ». Viens ensuite le tour de notre boucle de dessin principale qui appelle pour chaque objet de notre cours (aussi bien la chargeuse que les paquets puisque ce sont toutes des sous classes de la classe abstraite `ObjetDeLaCoursABois`). Dans cette boucle on appelle donc la méthode `dessinerObjetDeLaCoursABois()` appartenant à notre dessinateur `ObjetDeLaCoursABois`. Une fois dans cette méthode, on appelle la méthode de `g`, `getTransform()` qui nous retourne l'objet `AffineTransform` associé à notre `objetDeCoursABois`. Un objet `AffineTransform` est plus ou moins les coordonnées d'un objet géométrique sous forme de ligne associée à des vecteurs (Et donc très utile pour toutes nos futures translation/rotations géométriques). Au sein de notre méthode `dessinerObjetDeLaCoursABois()` on utilise des variables prenant en compte le décalage de `x` et `y` une fois le facteur de zoom appliqué. On crée l'objet `AffineTransform`, `identity`, qui va subir la rotation d'angle et translation de coordonnées de notre `ObjetDeCoursABois` par les méthodes respectives `rotate()` et `translate()`. On applique les transformations de notre objet dans l'objet `g` par `setTransformation()`. Puis on remplit finalement la forme que l'on vient de placer dans notre `ObjetGraphic`.

4. ALGORITHME POUR DETERMINER SI UN POINT APPARTIENT A UNE SURFACE

Vérifier si le point P de coordonnées (x, y) est contenu dans le rectangle A de centre (x, y) d'angle θ , de longueur l de largeur L. La méthode appartient à la classe contrôleur.

Booléen **PointEstDansRectangle (P : Point, A : ObjetDeLaCour)**

Début

```
// Centrer les coordonnées pour que A soit de centre (0,0) (image 1)
Int xTemp, yTemp;
xTemp = P.x - A.x;
yTemp = P.y - A.y;

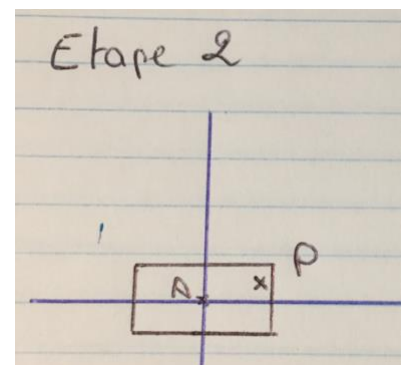
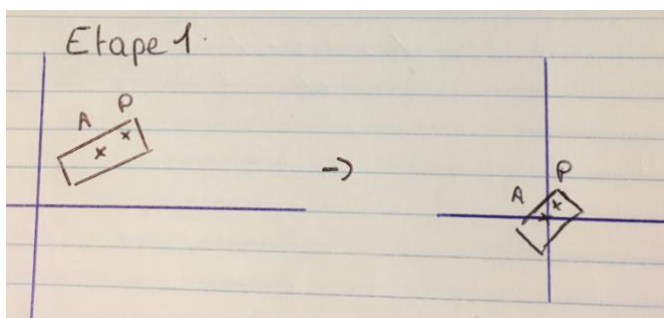
// Faire une rotation de l'angle d'inclinaison inverse du point centré pour vérifier si le
// point centré appartient au rectangle de centre l'origine et d'inclinaison nulle (image 2)
Int thetaTemp = theta * pi / 180;
Int x = xTemp;
Int y = yTemp;
xTemp = x cos(-thetaTemp) + y sin(-thetaTemp);
yTemp = -x sin(-thetaTemp) + y cos(-thetaTemp);

// Vérification finale

Booléen Appartient = Vrai;
Si (xTemp >= A.l / 2 ou xTemp <= -A.l / 2)
    Appartient = Faux;
FinSi
Si (yTemp >= A.L / 2 ou yTemp <= -A.L / 2)
    Appartient = Faux;
FinSi

Retourner Appartient;
```

Fin

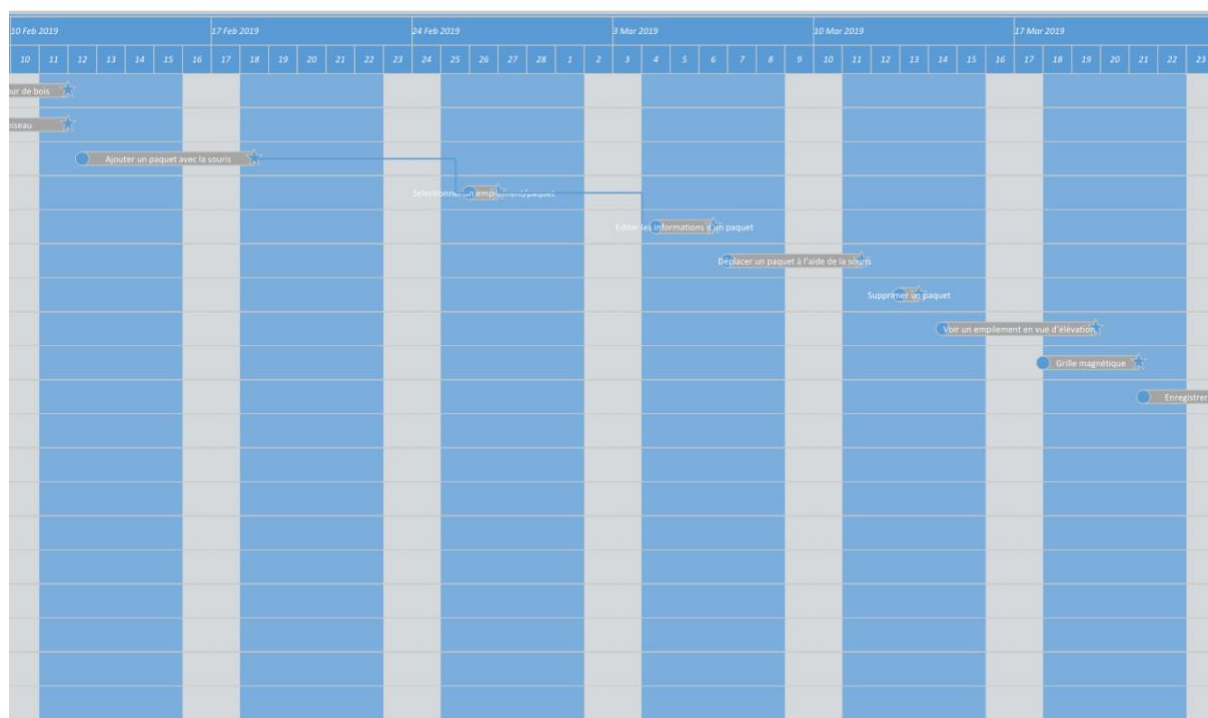




5. PLAN DE TRAVAIL MIS À JOUR TENANT COMPTE DES FONCTIONNALITÉS ATTENDUES PAR LE CLIENT AU LIVRABLE 3

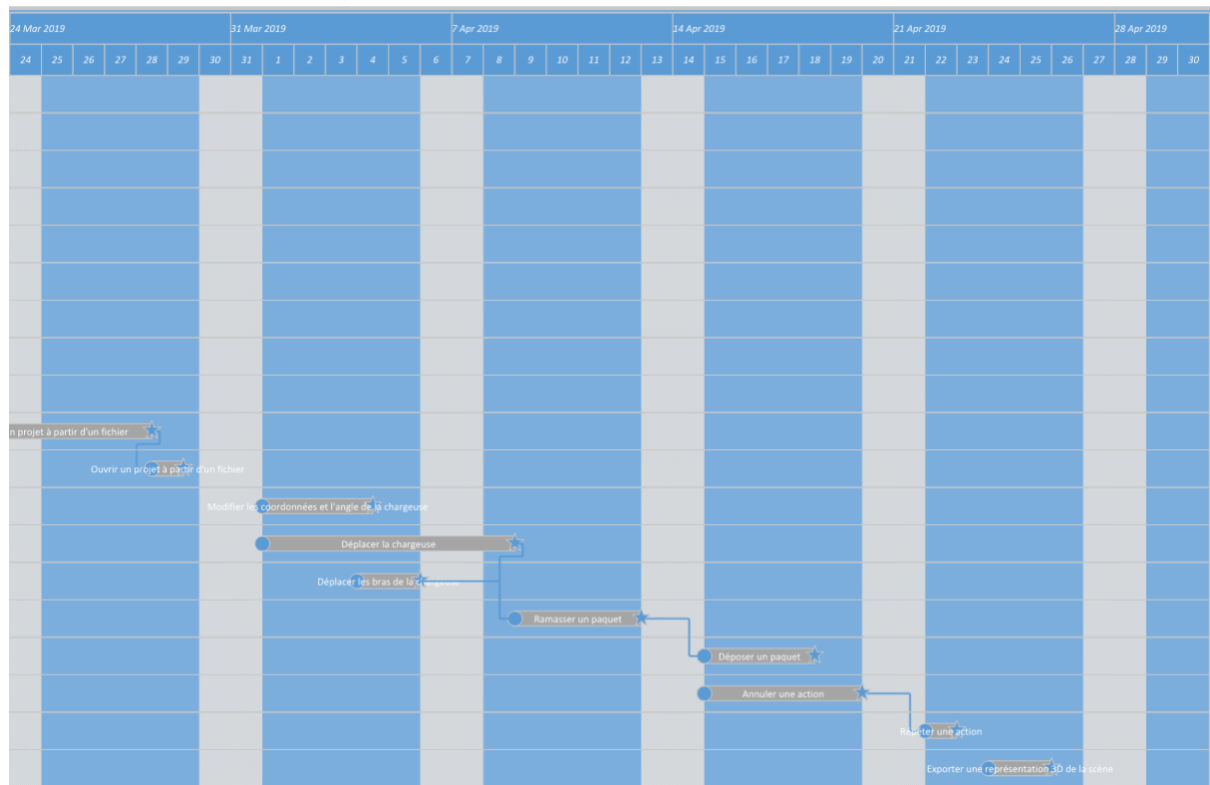
4 heures /semaine * 4 étudiants * 12
semaine = 192 heure total,
4 heures * 4 = 16 heures par semaine
16 heures / 5 jours nous donne un peu
plus de 3 heures par jour de capacité au
travers de l'équipe.

ID	Nom de tâche	Notes de la tâche	Début	Terminer	Durée	9 Feb 2019				
						4	5	6	7	8
1	Zoomer/Dézoomer la cour de bois		2/6/2019	2/11/2019	16h					
2	Voir la cour à vol d'oiseau		2/6/2019	2/11/2019	16h					
3	Ajouter un paquet avec la souris		2/12/2019	2/18/2019	16h					
4	Sélectionner un empiement/paquet	Dépend de @3	2/26/2019	2/26/2019	4h					
5	Editer les informations d'un paquet	Dépend de @3 et @4	3/4/2019	3/6/2019	6h					
6	Déplacer un paquet à l'aide de la souris		3/7/2019	3/11/2019	8h					
7	Supprimer un paquet		3/13/2019	3/13/2019	2h					
8	Voir un empiement en vue d'élévation		3/14/2019	3/19/2019	10h					
9	Grille magnétique		3/18/2019	3/21/2019	10h					
10	Enregistrer un projet à partir d'un fichier		3/21/2019	3/28/2019	15h					
11	Ouvrir un projet à partir d'un fichier	Dépend de @10	3/28/2019	3/29/2019	3h					
12	Modifier les coordonnées et l'angle de la chargeuse		4/1/2019	4/4/2019	10.5h					
13	Déplacer la chargeuse		4/1/2019	4/8/2019	18h					
14	Déplacer les bras de la chargeuse		4/4/2019	4/5/2019	6h					
15	Ramasser un paquet	Dépend de @13 et @14	4/9/2019	4/12/2019	12h					
16	Déposer un paquet	Dépend de @14 et @15	4/15/2019	4/18/2019	10.5h					
17	Annuler une action		4/15/2019	4/19/2019	15h					
18	Répéter une action	Dépend de @17	4/22/2019	4/22/2019	3h					
19	Exporter une représentation 3D de la scène		4/24/2019	4/25/2019	8h					





Faculté des sciences et de génie



6. CONTRIBUTION DE CHACUN DES MEMBRES DE L'ÉQUIPE

Tâche	Nom du membre de l'équipe
Diagramme de classe de conception	Équipe / Marc-André Trahan
Architecture logique	Julienne Petronie Betsama Assolo
Diagramme de séquence de conception	
3.1.1 3.1.2 3.2 3.3	Maxime Prieur Maxime Prieur Julienne Petronie Betsama Assolo Guillaume Genest-Lebel
Algorithme en pseudocode	Maxime Prieur
Plan de travail mis à jour	Guillaume Genest-Lebel
Code pour interface de l'application	Marc-André Trahan – Maxime Prieur
Correction éléments livrable 1	Maxime Prieur



7. ANNEXE

7.1. Énoncé 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.



7.2. Modèle du domaine

7.2.1. Diagramme des classes conceptuelles

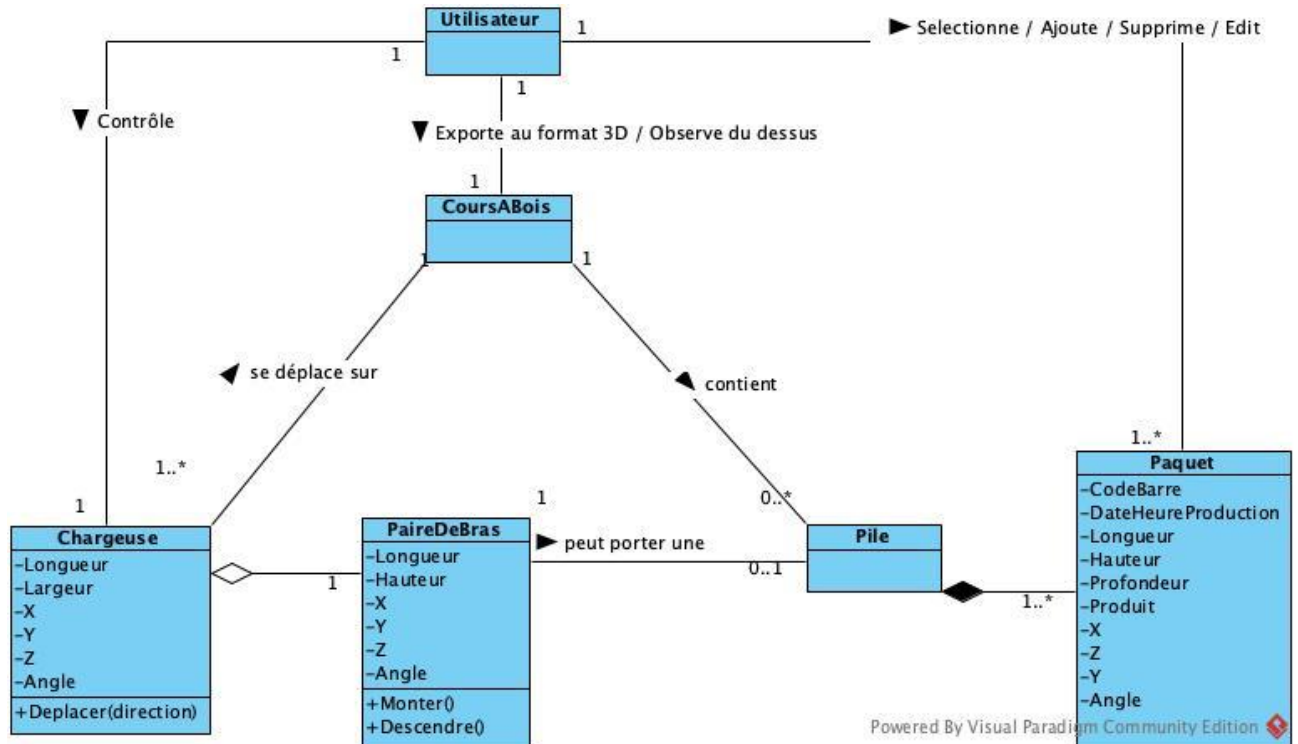


Diagramme des classes conceptuelles 1



7.2.2. *Texte explicatif des classes conceptuelles*

La **CoursABois** peut contenir une ou plusieurs **Chargeuses** et de 0 à plusieurs **Piles** de **Paquets**.

La **Chargeuse** possède une **Coordonnée**, est déplaçable via l'interface utilisateur et peut posséder une **PaireDeBras** pour le chargement et le déchargement des **Piles** de **Paquets**

La **PaireDeBras** de la **Chargeuse** possède ses propres **Coordonnées** à des fins d'alignement entre les **Piles** de **Paquets** et la **chargeuse**.

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

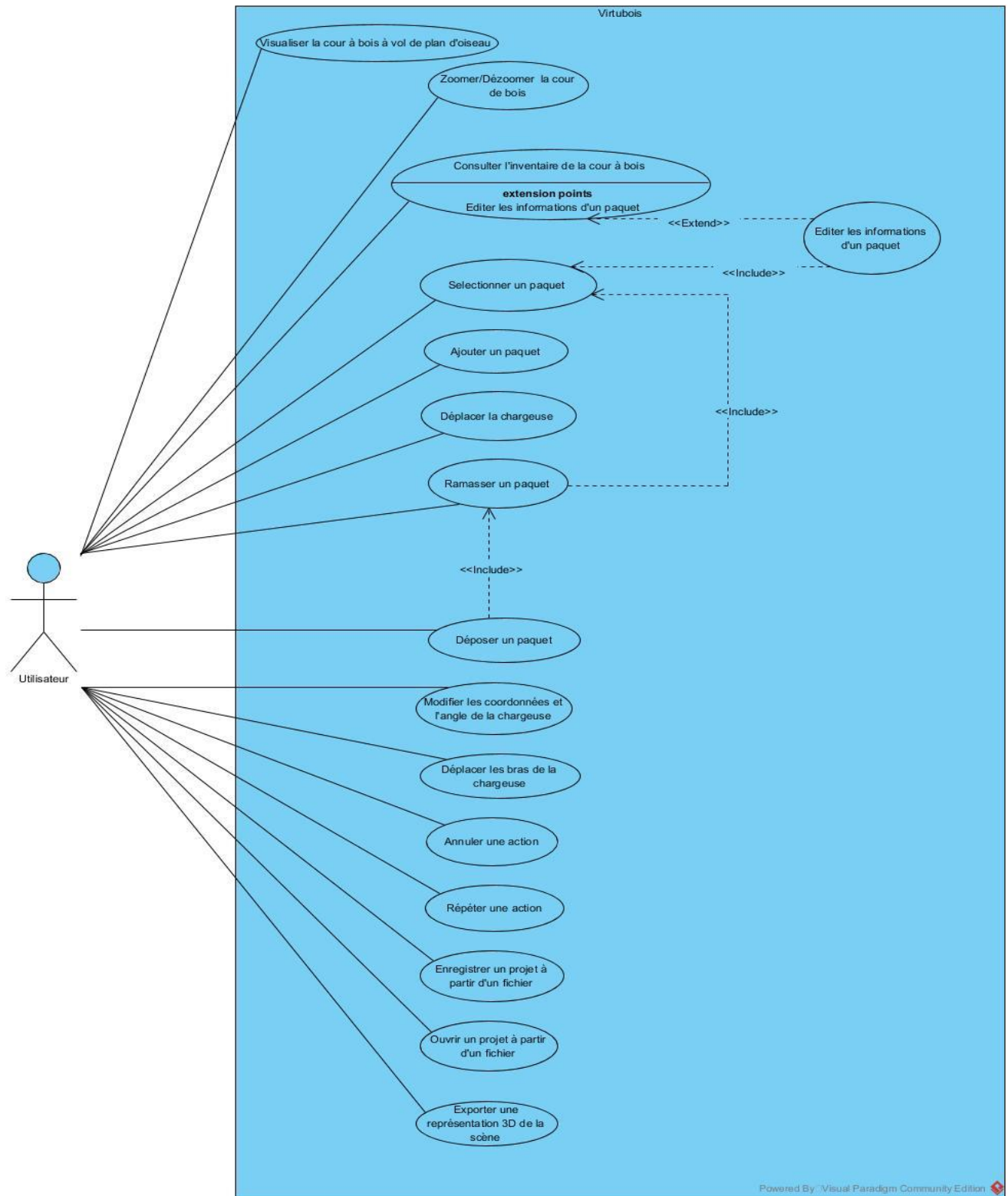
Cette pile est constituée de **paquets**.

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

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

7.3. Modèle des cas d'utilisation

7.3.1. Diagramme des cas d'utilisation





7.3.2. *Texte des cas d'utilisation*

Zoomer/Dézoomer la cours à bois

L'utilisateur clique sur le bouton « +/- » pour zoomer / dézoomer en fonction de ses convenances et la taille de la cour s'ajuste de celles-ci.

Consulter l'inventaire de la cour à bois

L'utilisateur déplace la souris sur le plan cartésien de la cour à bois et les coordonnées, x et y (en mètres), correspondantes sont affichées sur la zone d'écran prévu à cet effet. Il sélectionne le paquet qu'il souhaite modifier (voir cas d'utilisation « Sélectionner un paquet ») et met à jour les informations de celui-ci (voir cas d'utilisation « Éditer les informations d'un paquet »).

Éditer les informations d'un paquet

L'utilisateur sélectionne la pile contenant le paquet (voir cas d'utilisation « Sélectionner un paquet »), rentre les nouvelles informations sur le paquet et fait la sauvegarde des modifications apportées.

Sélectionner un paquet

1. L'utilisateur clique sur un point de l'écran.
2. Tous les paquets qui passent aux coordonnées correspondantes sont « sélectionnés ».
3. Affichage de la pile de paquets dans la section « vue de la pile ».
4. L'utilisateur clique sur le paquet qu'il veut modifier dans la pile de paquets.
5. Les informations du paquet sont affichées dans la fenêtre attribuée à cet effet.



Ligne 4 : La pile contient uniquement un seul paquet, le paquet est automatiquement sélectionné et les informations sont affichées dans la fenêtre attribuée à cet effet.

Ajouter un paquet

1. L'utilisateur clique sur le bouton « Ajout Paquet ».
2. L'utilisateur clique à l'endroit de la cour où il veut positionner le paquet.
3. Affichage des différentes caractéristiques du nouveau paquet.
4. L'utilisateur saisit les différentes caractéristiques du paquet.
5. L'utilisateur clique sur le bouton « Valider ».
6. Le nouveau paquet est créé à l'emplacement choisi par l'utilisateur.

Ligne 6 : Les informations saisies ne sont pas valides. Affichage d'un message d'erreur et le paquet ne sera pas ajouté.

Déplacer la chargeuse

1. L'utilisateur appuie sur la touche « Flèche haut » afin de faire avancer la chargeuse.
2. La chargeuse effectue le mouvement dicté par l'utilisateur.

Ligne 1 : L'utilisateur appuie sur la touche « Flèche bas » afin de faire reculer la chargeuse.
Ligne 1 : L'utilisateur appuie sur la touche « Flèche gauche » ou « Flèche droite » afin de modifier l'angle de la chargeuse.
Ligne 1 : L'utilisateur modifie les coordonnées de la chargeuse à l'aide de la fenêtre prévue à cet effet (voir cas d'utilisation « Modifier les coordonnées et l'angle de la chargeuse »).

Modifier les coordonnées et l'angle de la chargeuse

1. L'utilisateur entre les nouvelles coordonnées et l'angle dans la fenêtre prévue à cet effet.
2. La chargeuse se retrouve à l'endroit déterminé par l'utilisateur (coordonnées x et y) et avec l'angle souhaité.

Ligne 2 : Les coordonnées ne sont pas valides, la chargeuse reste aux anciennes coordonnées.

Ramasser un/des paquet(s)

1. L'utilisateur positionne les bras de la chargeuse au niveau du paquet à soulever.



Faculté des sciences et de génie

2. L'utilisateur déplace la chargeuse devant la pile souhaitée.
3. L'utilisateur déplace les bras de la chargeuse vers le haut.
4. Le/les paquet(s) est/sont sélectionnés (voir cas d'utilisation « Sélectionner un paquet »).

Déposer un ou des paquet(s)

1. L'utilisateur ramasse le paquet (voir cas d'utilisation « Ramasser un paquet »).
2. L'utilisateur déplace la chargeuse à l'endroit de dépôt.
3. L'utilisateur abaisse les bras de la chargeuse pour qu'ils soient positionnés au sol ou sur un autre paquet.
4. L'utilisateur recule la chargeuse.
5. Le paquet est déposé à l'emplacement choisi par l'utilisateur et ces informations sont sauvegardées.

Déplacer les bras de la chargeuse

1. L'utilisateur appuie sur la combinaison des touches « Shift » et « Flèche haut ».
2. Les bras de la chargeuse montent.

Ligne 1 : L'utilisateur appuie sur la combinaison des touches « Shift » et « Flèche bas » afin de faire descendre les bras de la chargeuse.

Annuler une action

1. L'utilisateur effectue une action quelconque.
2. L'action est prise en compte par l'application.
3. L'utilisateur clique sur le menu déroulant « Fichier ».
4. Le menu déroulant s'affiche.
5. L'utilisateur clique sur le bouton « Annuler ».
6. La dernière action effectuée par l'utilisateur est annulée.



Répéter une action

1. L'utilisateur annule une action (voir cas d'utilisation « Annuler une action »).
2. L'utilisateur clique sur le menu déroulant « Fichier ».
3. Le menu déroulant s'affiche.
4. L'utilisateur clique sur le bouton « Répéter ».
5. La dernière action annulée est reprise en compte par le système.

Enregistrer un projet à partir d'un fichier

1. L'utilisateur effectue des actions modifiant l'état de la cour à bois.
2. L'utilisateur clique sur le menu déroulant « Fichier ».
3. L'utilisateur clique sur le bouton « Enregistrer ».
4. Affiche une fenêtre pour entrer le nom du fichier de sauvegarde.
5. L'utilisateur saisit le nom du fichier.
6. Sauvegarde du projet dans un fichier ayant le nom que l'utilisateur a choisi.

Ligne 4 : Si le projet a déjà été sauvegardé dans un fichier, l'application sauvegarde sans demander le nom du fichier.

Ouvrir un projet à partir d'un fichier

1. L'utilisateur clique sur le menu déroulant « Fichier ».
2. L'utilisateur clique sur le bouton « Ouvrir ».
3. Affiche une fenêtre pour sélectionner un fichier afin d'ouvrir un projet.
4. L'utilisateur sélectionne le fichier contenant le projet qu'il veut ouvrir.
5. L'application charge le fichier.

Ligne 3 : Si le projet actuel n'a pas été sauvegardé et qu'il n'est pas vide, demande à l'utilisateur s'il veut en faire la sauvegarde avant d'ouvrir un fichier.

Ligne 5 : Si le fichier n'est pas dans le format nécessaire afin de pouvoir en faire l'ouverture, affiche un message d'erreur et le projet n'est pas importé.

Exporter une représentation 3D de la scène

1. L'utilisateur clique sur le menu déroulant « Fichier ».
2. L'utilisateur clique sur le bouton « Exporter ».
3. Affiche une fenêtre pour entrer le nom du fichier d'exportation.
5. L'utilisateur saisit le nom du fichier.
6. Sauvegarde du projet dans un fichier ayant le nom du fichier d'exportation que l'utilisateur a choisi.

7.3.3. Diagramme de séquence système

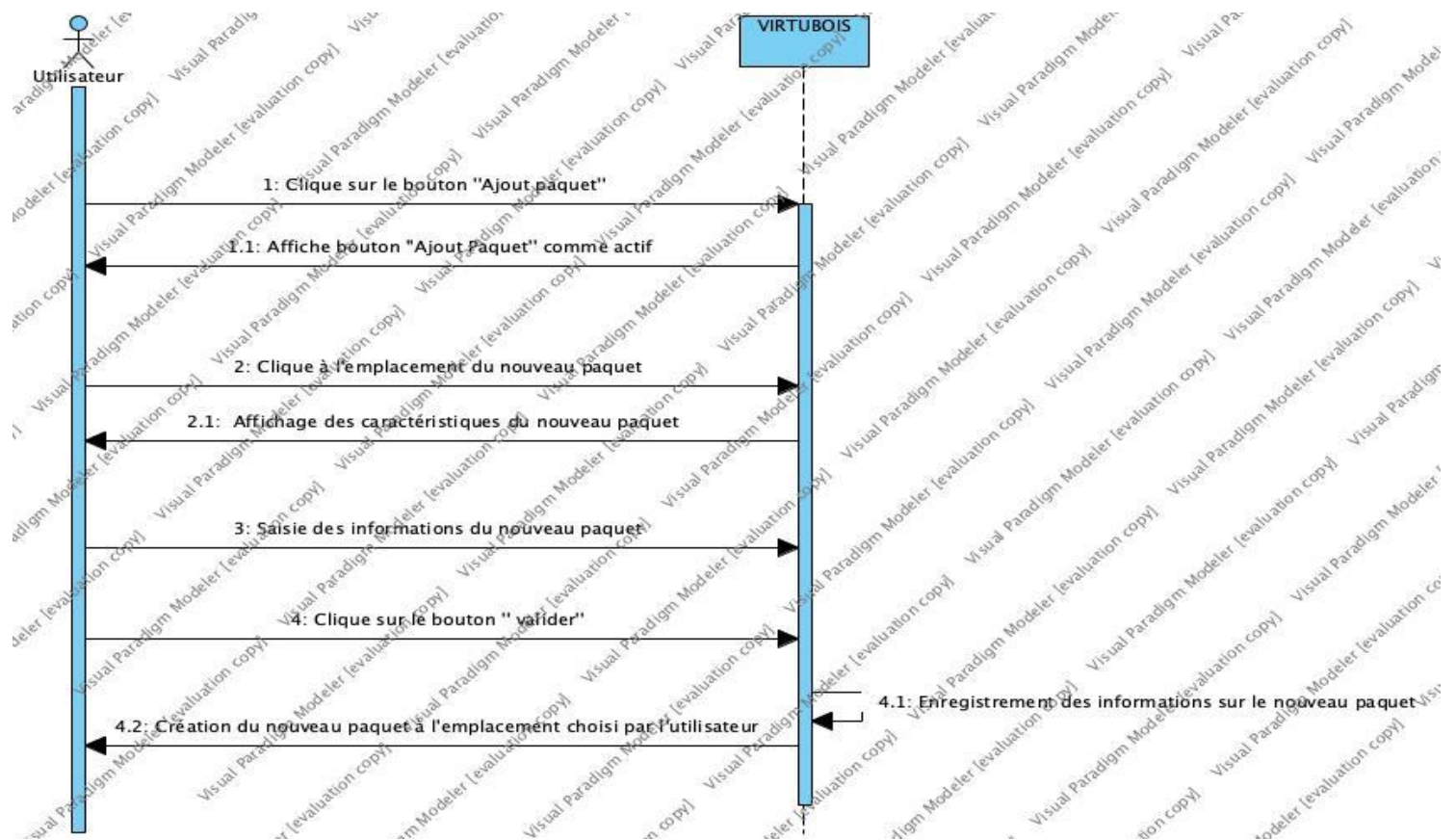
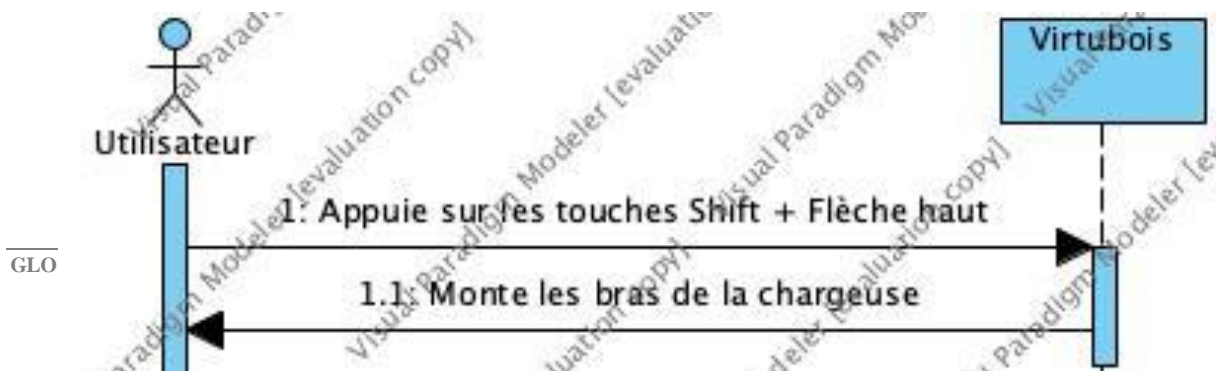


Figure 1 : Ajout Paquet



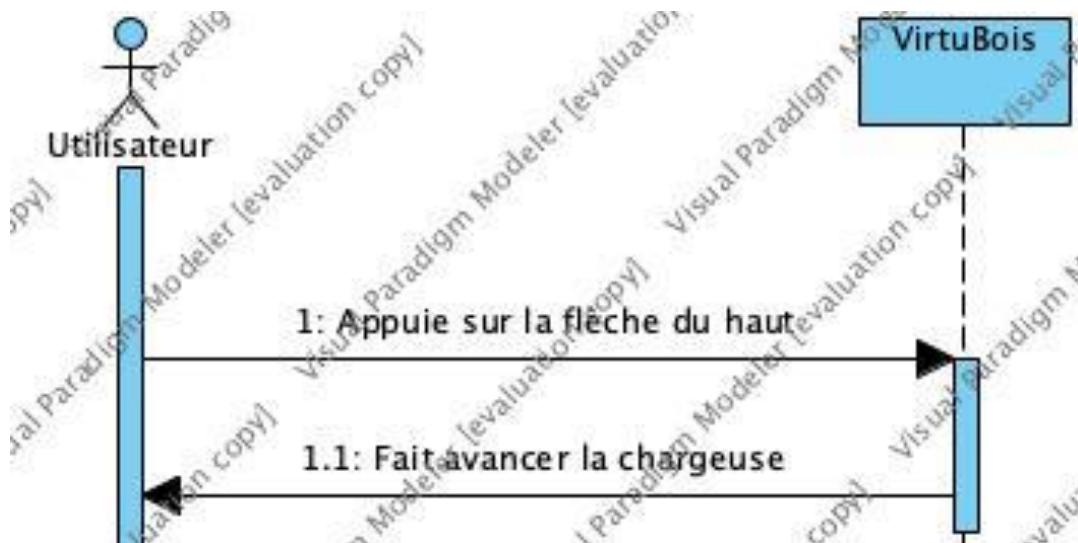


Figure 3 : Déplacer la chargeuse

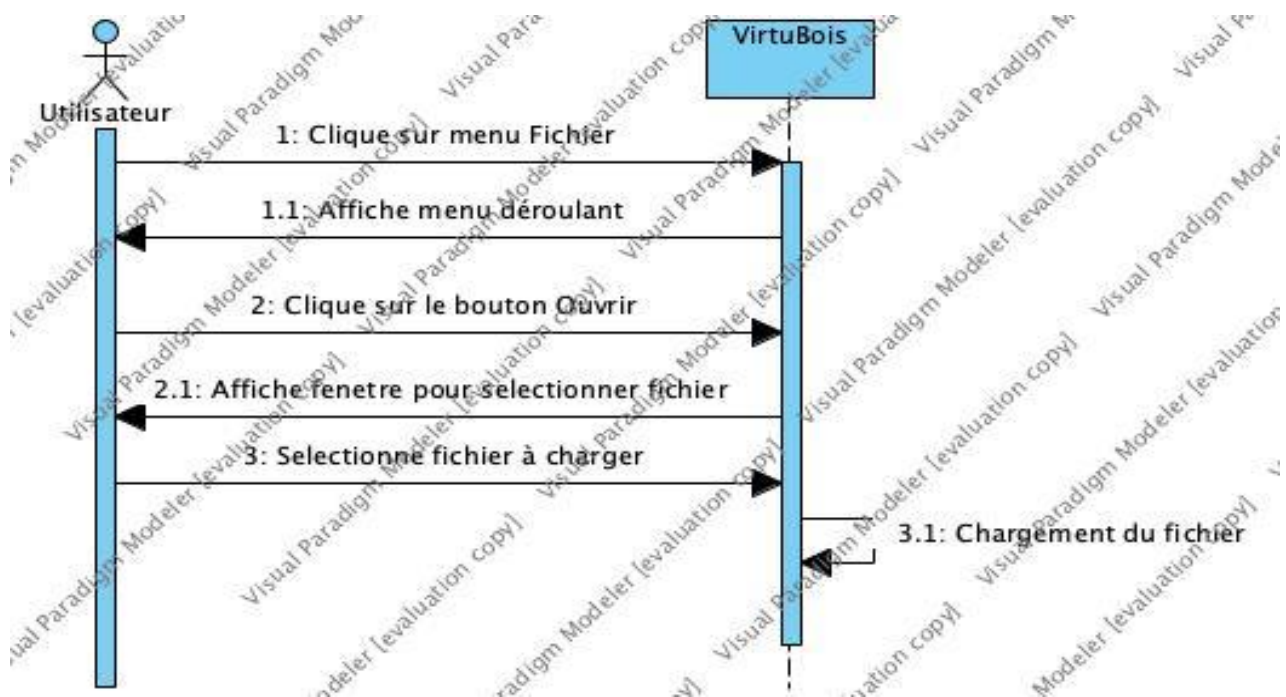


Figure 4 : Charger un fichier

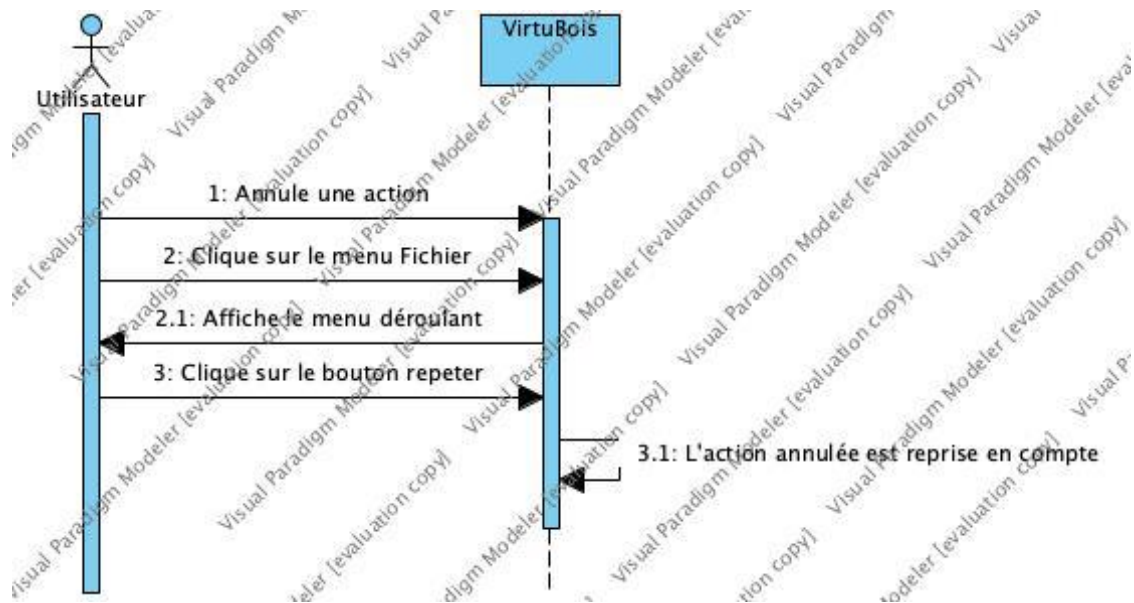


Figure 5 : Répéter une action

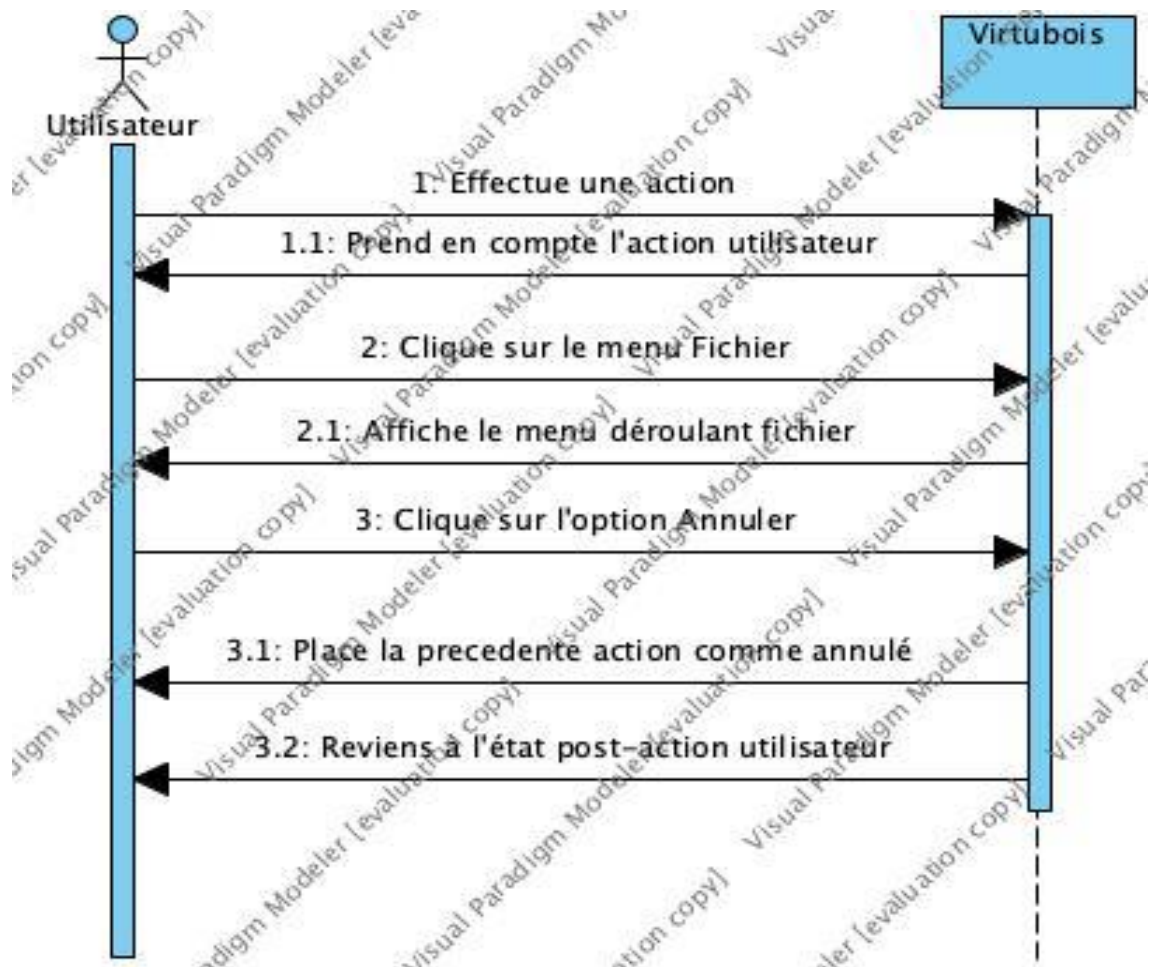


Figure 6 : Annuler une action

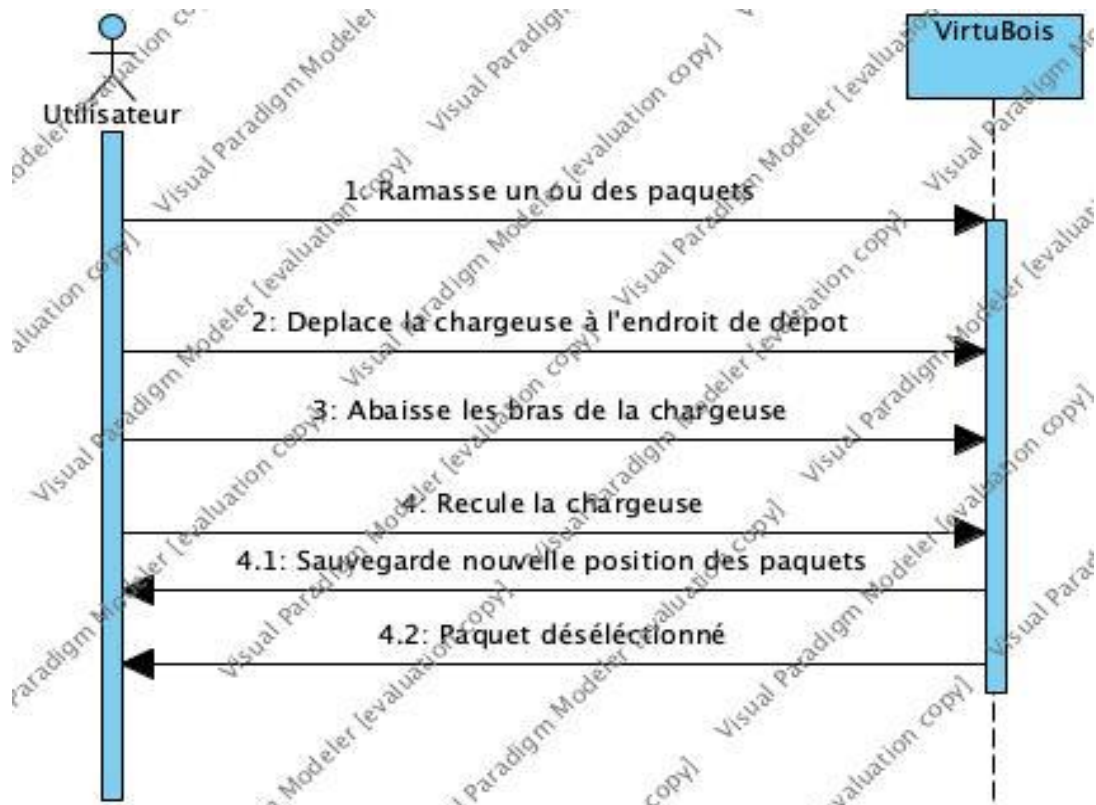


Figure 7 : Déposer un paquet

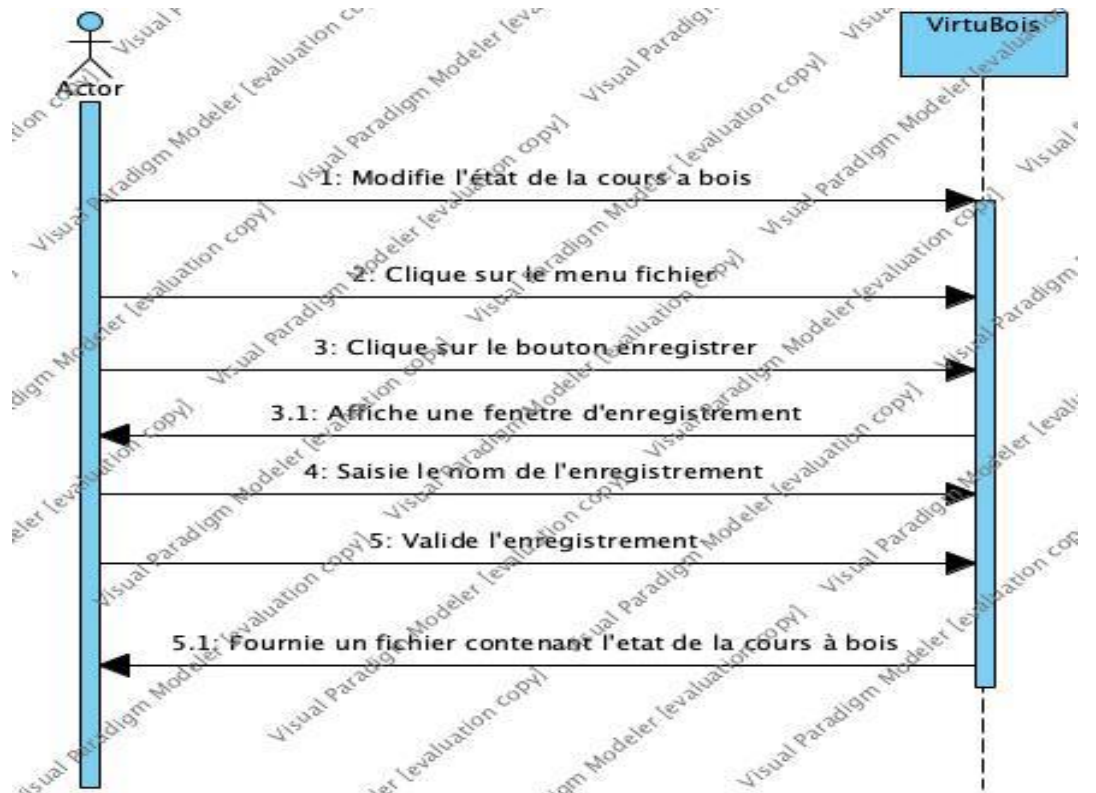


Figure 8 : Enregistrer dans un fichier

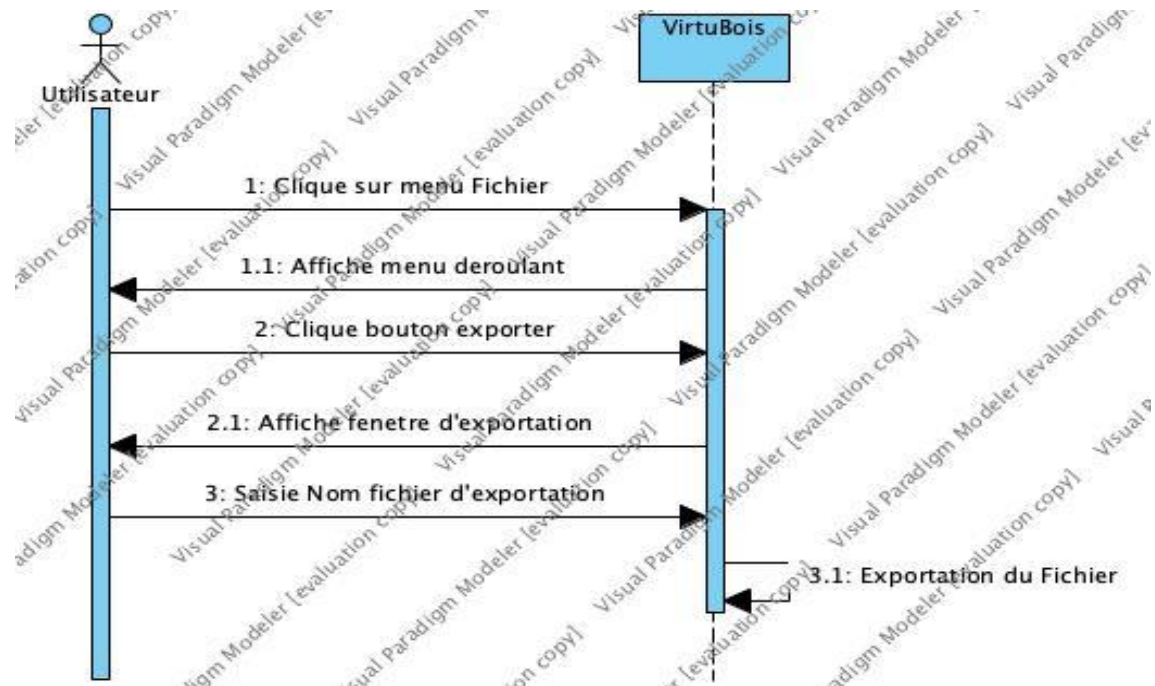


Figure 9 : Exporter au format 3D

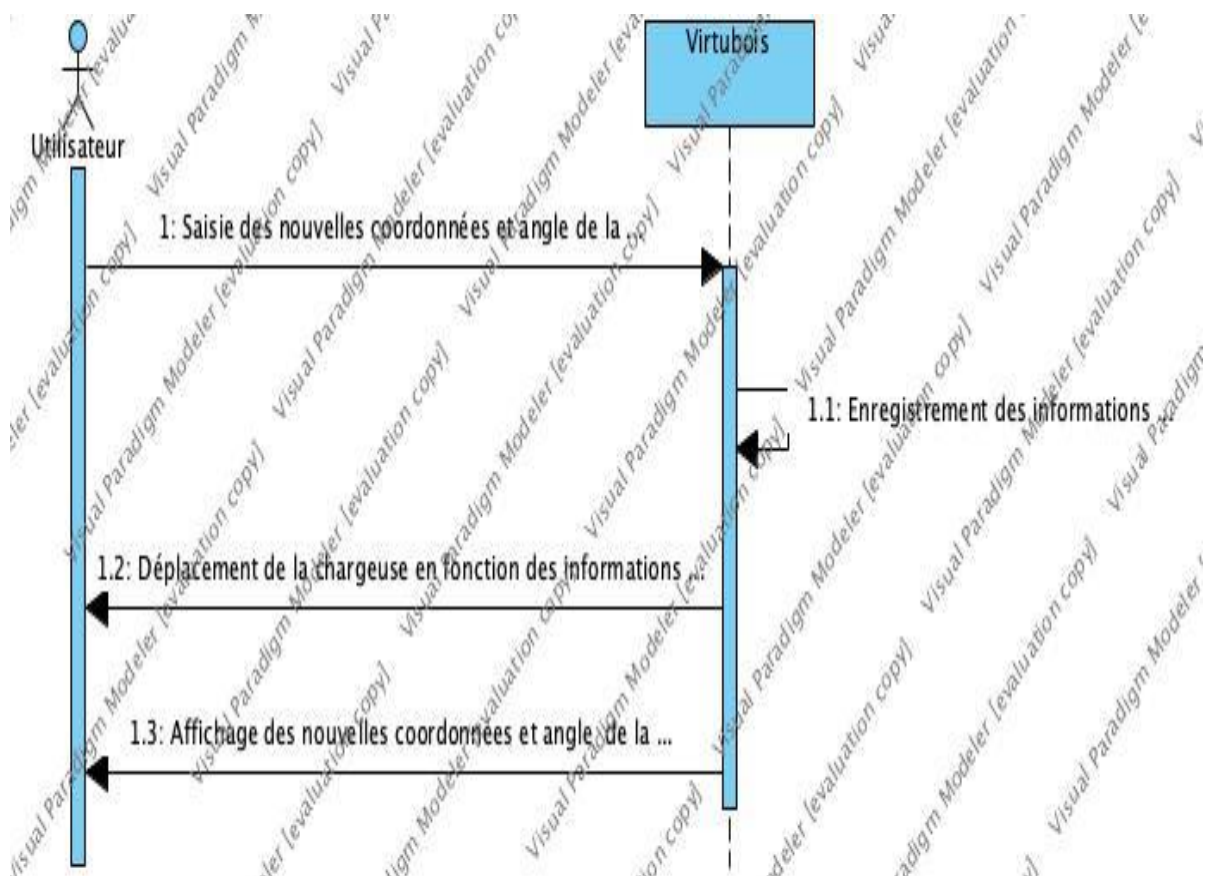


Figure 10 : Modifier manuellement les coordonnées de la chargeuse

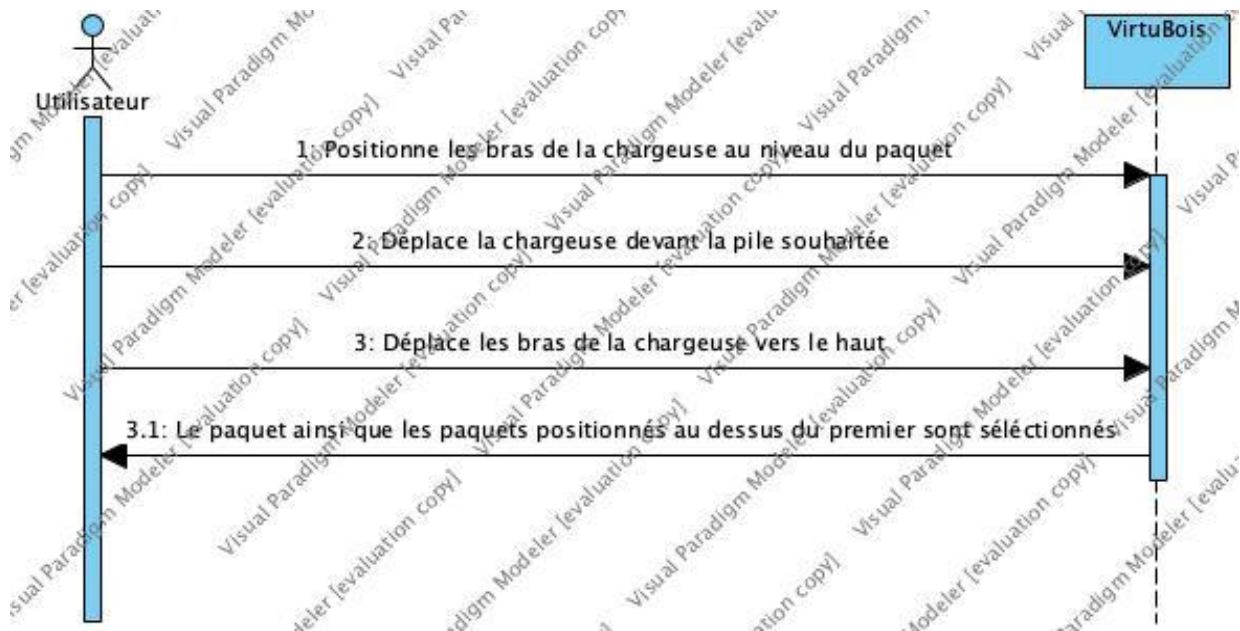


Figure 11 : Soulever un paquet



Figure 12 : Sélectionner un paquet



UNIVERSITÉ
LAVAL

Faculté des sciences et de génie