



GLO-2004 Génie logiciel orienté objet

Livrable 4 : MicrosofTears

présenté à

M. Jonathan Gaudreault

par

Équipe 52 —

<i>matricule</i>	<i>nom</i>	<i>signature</i>
KEJOB2	536 777 791	Kevin Jobin
LUNIQ	111 267 600	Lucas Niquet
LYLEL	111 271 294	Lydia Lelièvre

Université Laval

14 décembre 2021

Historique des versions		
<i>version</i>	<i>date</i>	<i>description</i>
0.0	10 décembre 2021	création du document

Table des matières

Table des figures	ii
Liste des tableaux	iii
1 Énoncé de Vision	1
2 Modèle du domaine	3
3 Diagramme de classes de conception	4
4 Cas d'utilisation	6
5 Conclusion	8
6 Contribution des membres de l'équipe	10

Table des figures

1.1 Saisie d'écran dans Microsoftears	2
2.1 Modèle du domaine	3
3.1 Diagramme de classes de conception	5
4.1 Diagramme des cas d'utilisation	7

Liste des tableaux

Chapitre 1

Énoncé de Vision

L'objectif principal de ce projet consiste à produire une application, nommée *Microsoft-tears*, qui permettra à un utilisateur de produire un plan de micro-roulotte de style «teardrop» conforme au design de Tony H. Latham. Plus spécifiquement, le travail consiste à informatiser le processus de construction de la micro-roulotte afin de produire des plans pouvant être facilement manipulés et exportés. Ceux-ci permettront éventuellement de fabriquer les parties nécessaires de la micro-roulotte (dans le cadre de ce projet, nous nous limiterons à la conception des murs).

Pour ce faire, nous produirons une application complète de modélisation qui laissera la liberté à l'utilisateur de choisir et de paramétrer les dimensions ainsi que le positionnement des parties de sa micro-roulotte. L'application devrait faciliter la conception du plan de sa micro-roulotte, par exemple placer le mur séparateur entre l'aire principal et l'aire de cuisine, placer une porte et/ou autres ouvertures comme des fenêtres ainsi que d'ajouter divers objets au plan pour aider à son design (par exemple un lit).

Nous permettrons aussi à l'utilisateur de pouvoir importer ou sauvegarder un projet, avancer ou revenir en arrière sur ses actions, exporter dans différents formats et zoomer/dézoomer sur ses plans. D'autres fonctionnalités seront aussi ajoutés afin de faciliter la production des plans et de rendre l'expérience facile et agréable pour l'utilisateur.

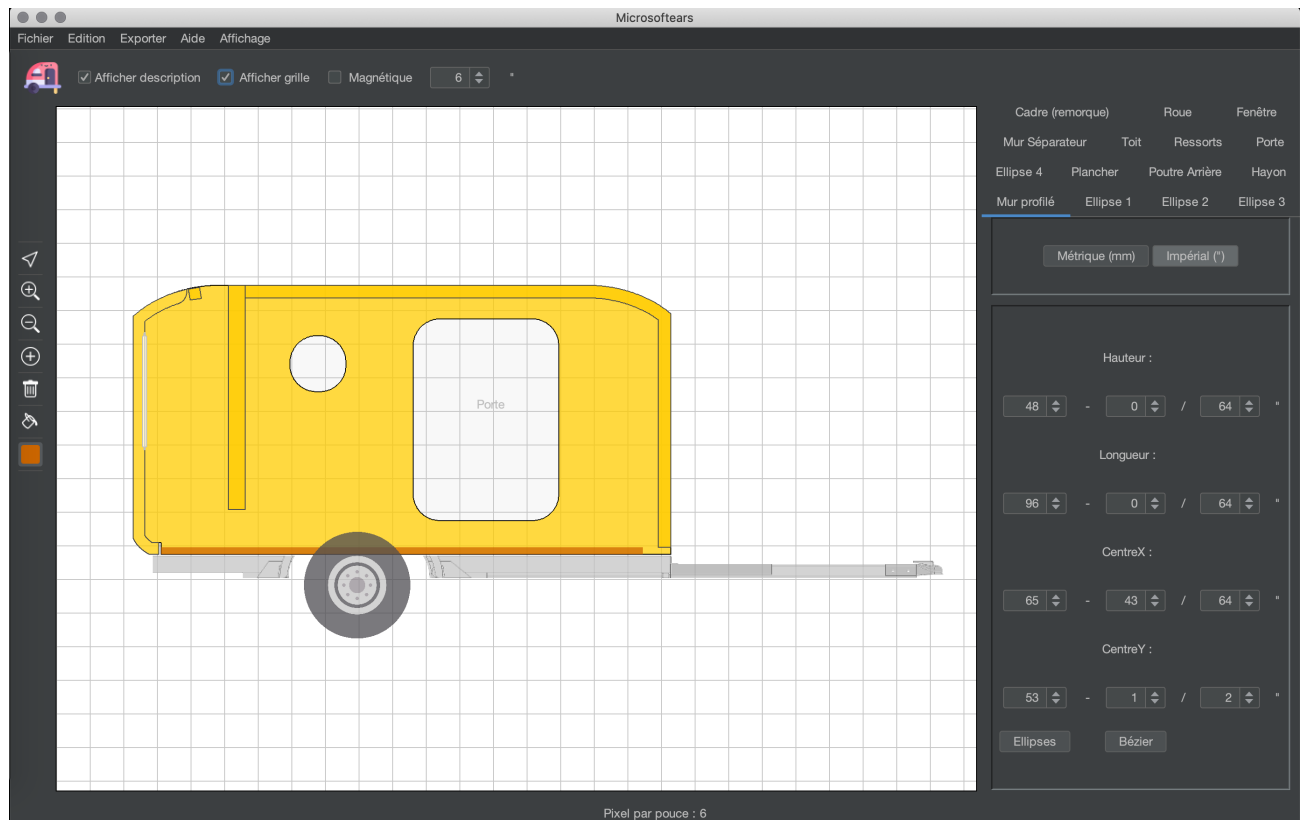


FIGURE 1.1 – Saisie d'écran dans Microsoftears

Chapitre 2

Modèle du domaine

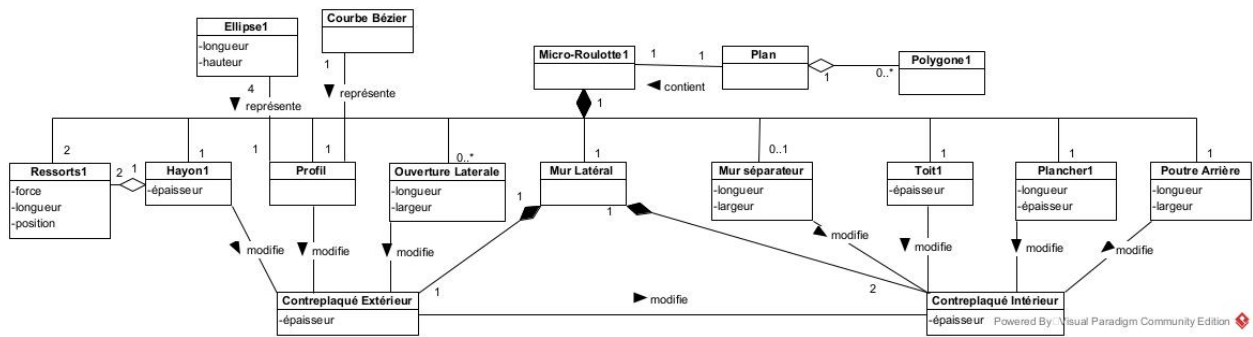
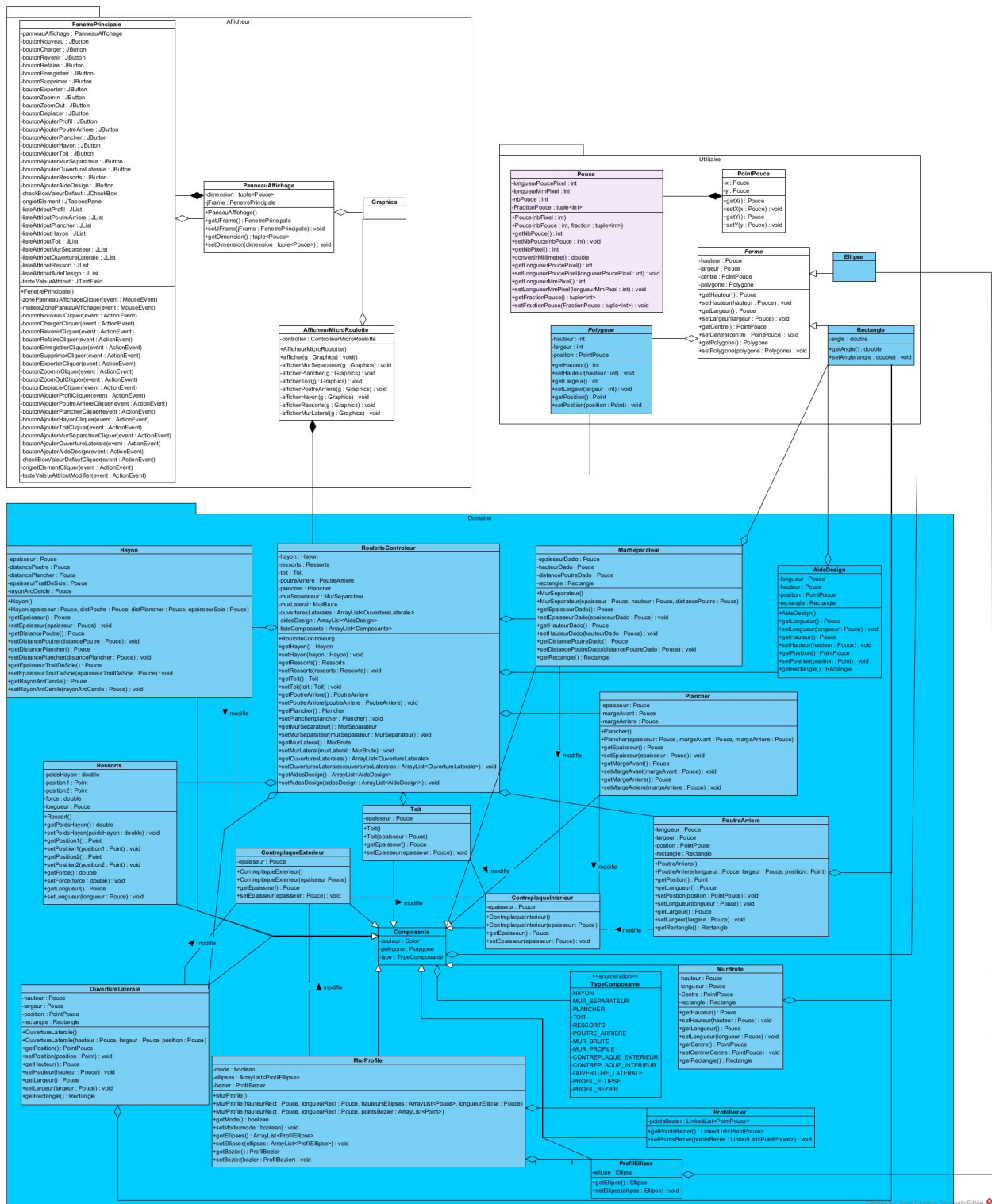


FIGURE 2.1 – Modèle du domaine

Chapitre 3

Diagramme de classes de conception



Chapitre 4

Cas d'utilisation

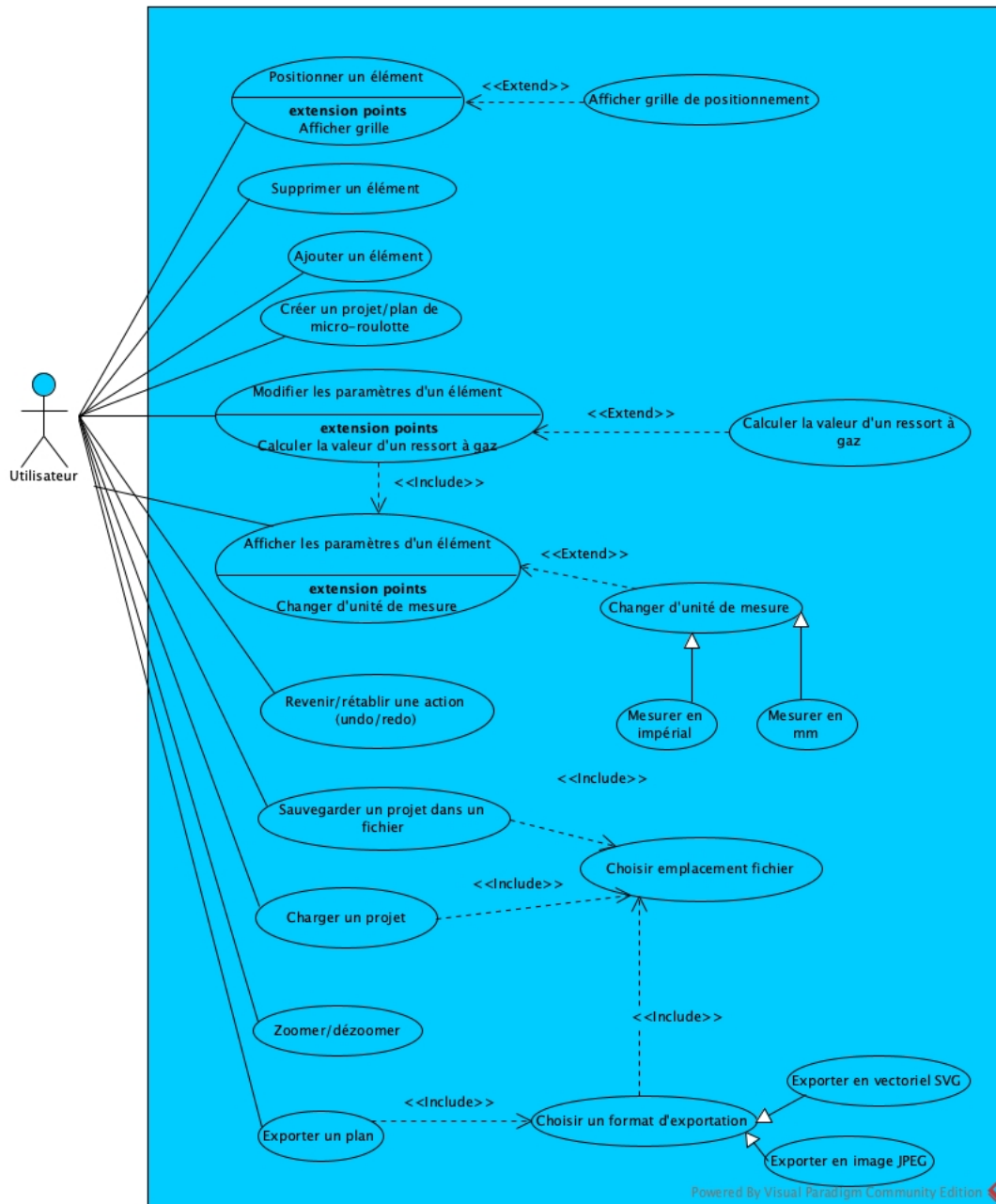


FIGURE 4.1 – Diagramme des cas d'utilisation

Chapitre 5

Conclusion

Les points forts : Au niveau du design de notre application, les classes du domaine était bien pensée. Le code était bien fondé et facile à réutiliser ce qui a permis une croissance aisée de l'application à travers les différentes itérations du projet. L'utilisation des classes abstraites a facilité l'ajout de différentes composantes. Notre affichage et l'apparence général de l'application est agréable et facile pour l'utilisateur à comprendre. L'ajout de nombreux détails visuels et la communication en temps réel entre la vue et le domaine permettent à l'utilisateur d'être en plein contrôle, limité seulement par son imagination. Le fait d'avoir créé nos classes pour les figures en 2D nous a permis une conversion d'impérial vers métrique sans trop de perte de précision et aussi de profiter des classes de la librairie standard de Java. Notre application permet l'entrée de donnée précise au 1/64 de pouce. Il aurait été souhaitable de supporter aussi les données négatives dans nos calculs mathématiques et de pousser plus loin la précision des décimales. Pour conclure, nous sommes très fiers du travail rendu.

Les points faibles et améliorations souhaités : Dans un contexte d'application utilisé dans l'industrie, il faudrait implémenter des moyens de garantir les services rendus à l'aide du concept de contrat. Cela permettrait d'éviter les utilisations qui pourraient faire planter le programme, car dans notre application peu de vérifications sont faites (on assume que l'utilisateur connaît et respecte la logique, ex : pas de mesure impossible). De plus, la majorité de nos fonctions ne sont pas optimisées, le temps de calcul est gourmand et les performances en sont affectées. En outre, l'utilisation de la sérialisation pour les fonctions *revenir* et *rétablir* (*undo* et *redo*) n'ont pas pu être implémentées pour l'ensemble des actions de l'interface en raison de la lenteur d'un tel processus. En contexte industriel, il serait intéressant d'utiliser les patrons du *Gang of four* appelés *Memento* et *Command* pour ces fonctions. En ce qui attrait à un des aspects majeurs de l'application, soit l'exportation de plans, le rendu final est plutôt imprécis et ne réponds pas aux attentes dans un contexte industriel. L'utilisation d'une librairie externe pour les fichiers de découpe (.svg) serait extrêmement utile, on peut penser à la librairie *Batik* de *Apache* ou encore *JFreeSVG*. Finalement, le design de l'interface, bien qu'il soit propre et agréable à regarder, n'est pas tout à fait optimal si l'on souhaiterait en faire une application à échelle industrielle. L'utilisation d'un éditeur graphique serait de mise, car dans ce projet chacun des *layout* a dû être codé à la main ce qui a ses avantages et ses inconvénients (coûteux en temps et parfois difficile d'imaginer le rendu versus un éditeur WYSIWYG (*What you see is what you get*)).

Chapitre 6

Contribution des membres de l'équipe

Lydia : Pour ce qui est du code, je me suis chargé de l'ajout des aides au design ainsi que les ouvertures latérales. je me suis aussi occupé de leur suppression. Pour ce qui est du livrable, je me suis chargée d'incorporer l'énoncé de vision et de rédiger la conclusion.

Lucas : Pour le code, j'ai travaillé sur les points du ressort et le calcul de la force, la liste de points pour le toit ainsi que la possibilité de modifier son épaisseur et la liste de points pour le mur séparateur ainsi que la possibilité de modifier ses paramètres. J'ai également travaillé sur les fonctionnalités undo/redo, le chargement de projet et l'enregistrement du projet. Pour le rapport, j'ai travaillé sur le diagramme de classes conceptuelles et le diagramme de classes de conception.

Kevin : Pour ce qui est du code, j'ai travaillé sur l'affichage du profil en mode courbe de Bézier, j'ai intégré une grille adaptable et magnétisable à l'interface. J'ai aussi travaillé sur les différentes composantes du domaine ainsi que sur la liaison entre la vue et les classes du domaine. J'ai peaufiné l'ajout d'aide au design ludique et d'ouvertures au design. J'ai travaillé sur les fonctions d'affichage en mode bézier. J'ai participé au codage des classes utilitaires et j'ai codé l'exportation sous forme d'image et sous forme svg. J'ai implémenté le zoom ainsi que le drag des composantes dans l'afficheur. J'ai aussi maintenu tout au long du projet la possibilité de changer de mesure impérial vers métrique. Pour ce qui du design, j'ai intégré une palette de choix de couleurs et j'ai participé au design de l'interface. Pour ce qui est du rapport, j'ai mis à jour les cas d'utilisation (qui n'ont pas changé) et j'ai aussi ajouté la capture d'écran ainsi qu'écrit la conclusion.