

Ahmed Yahia, Ramy (537003134)
Diamond, Félix-Antoine (536995235)
Viau, Charles (111137773)
Lavoie, Charles (537014771)
Noël, Félix-Olivier (537163132)

Génie logiciel orienté objet
GLO-2004

Livrable no 5 - Code opérationnel et démo

Travail présenté à
Marc Philippe Parent
et Anthony Deschênes

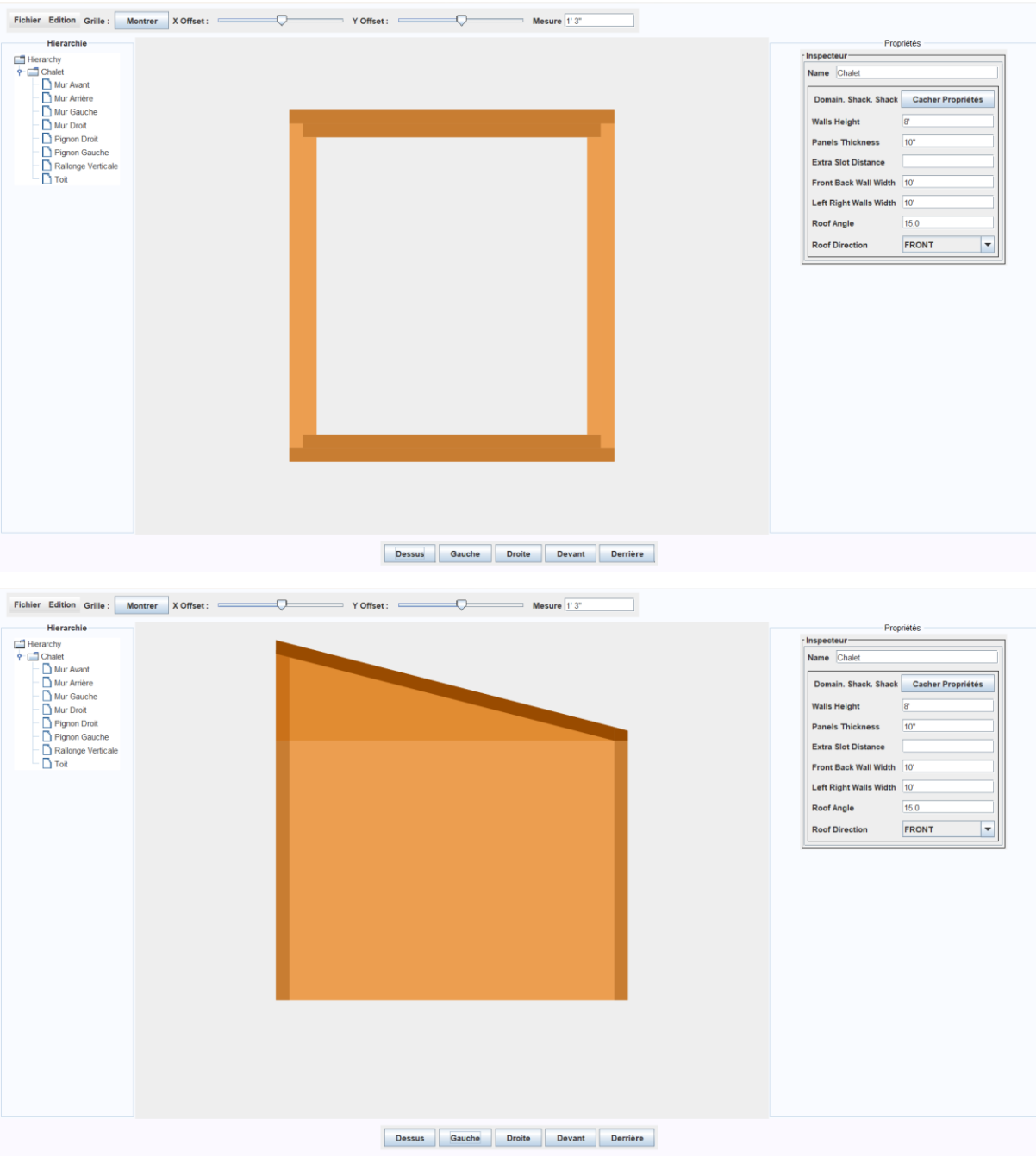
Génie logiciel et informatique
Université Laval

1. ÉNONCÉ DE VISION

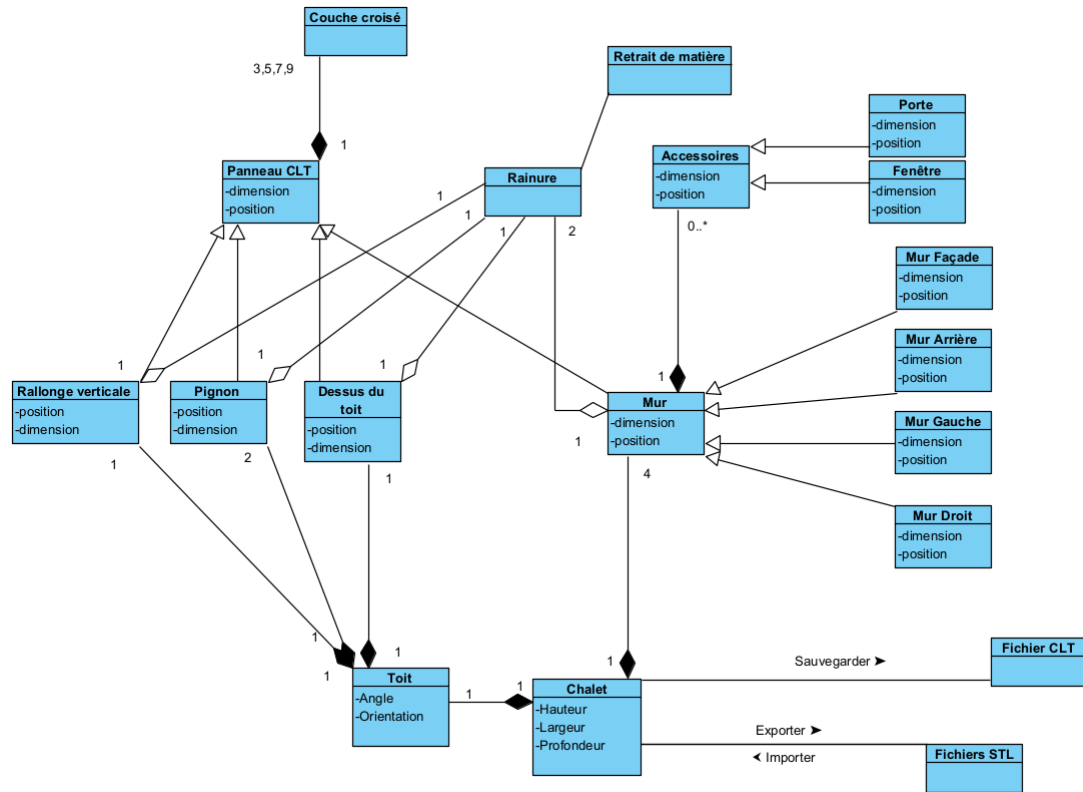
L'équipe a pour mandat de développer un logiciel de design afin de faciliter le processus de conception de chalets. La méthode actuelle consiste à produire le modèle 3D dans Revit et d'ensuite transférer l'objet dans AutoCAD afin de générer les plans de découpage des panneaux. Le nouveau logiciel, ChalCLT, combinera les fonctionnalités des deux autres logiciels afin d'accélérer le design du chalet. Plusieurs caractéristiques seront présentes afin de faciliter la navigation et l'utilisation de ChalCLT. Il y aura cinq vues du chalet, une vue pour chaque côté du chalet, et une vue en plan qui permettra de visualiser les connexions entre les murs. L'utilisateur aura la possibilité de «*zoomer*» sur chacune des cinq vues et de faire afficher une grille d'aide au positionnement afin d'assurer des mesures optimales. L'interface du logiciel sera divisée en quatre morceaux principaux, la vue présente du chalet, un arbre de hiérarchie contenant tous les objets modifiables (accessoires et panneaux), une fenêtre d'inspection où les propriétés de l'objet sélectionné sont affichées et modifiables, ainsi qu'une fenêtre de sélection de la vue.

Un chalet par défaut sera présent à l'ouverture du logiciel, l'utilisateur sera en mesure de modifier plusieurs propriétés. Par contre, certaines seront cachées et gérées par le logiciel et modifier certaines propriétés appliquera de la logique d'affaire spécifique déterminée par le client sur d'autres mesures. Le toit du chalet aura un seul versant et il sera possible de modifier son angle. L'utilisateur pourra ajouter des portes et des fenêtres aux murs du chalet, la dimension et la position de ceux-ci seront modifiables. ChalCLT va notifier l'utilisateur en cas d'erreur. Par exemple, si deux accessoires se superposent. Lorsque le modèle sera complété, il sera possible d'exporter les panneaux individuels en format STL.

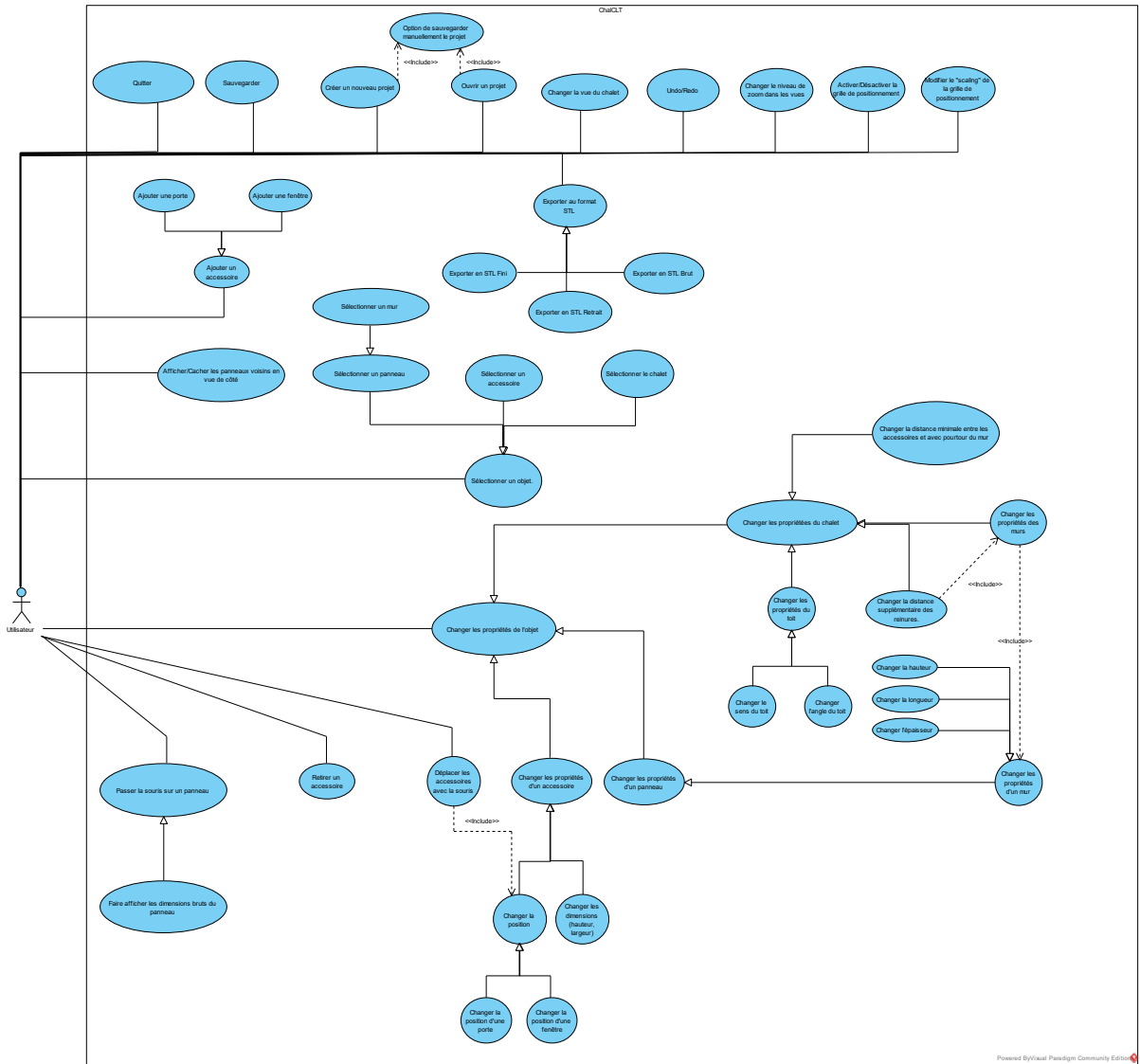
2. SAISIES D'ÉCRAN

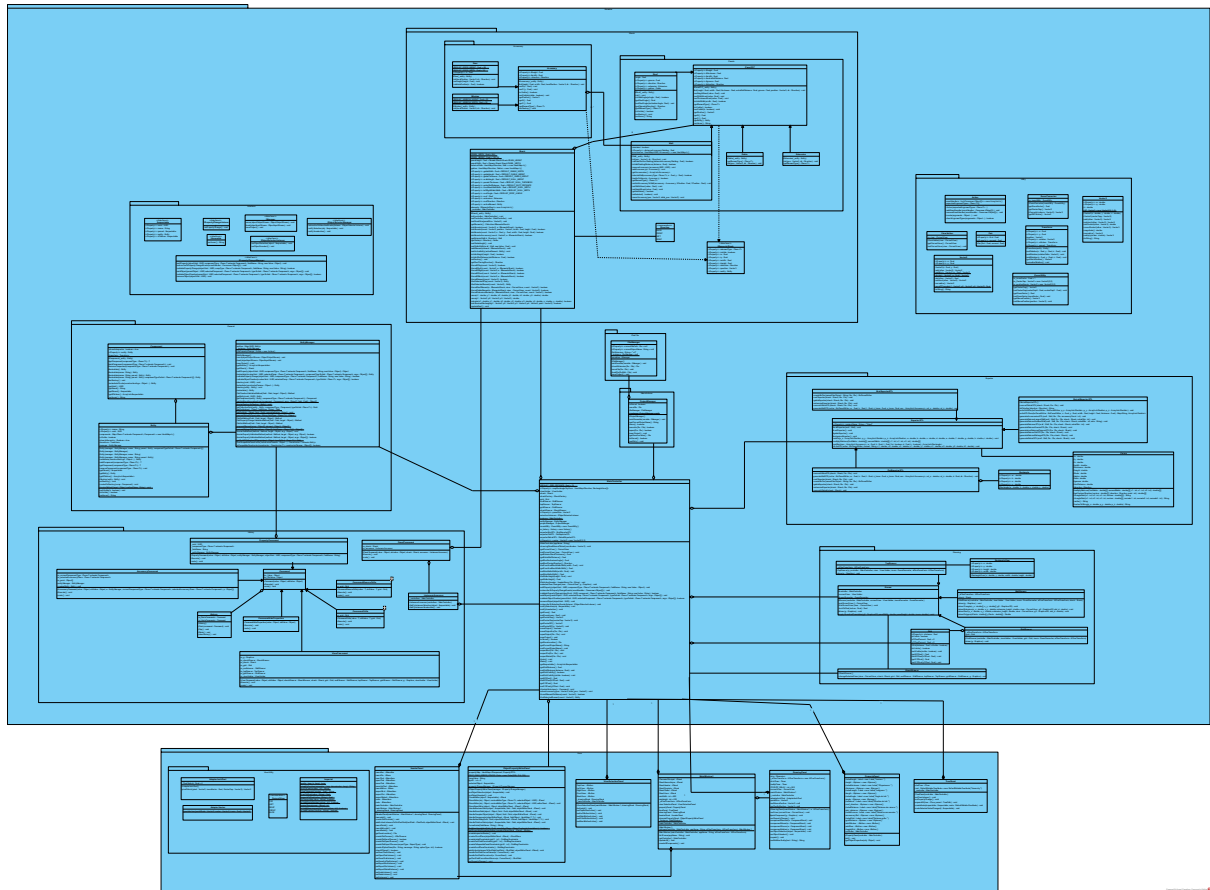


3. MODÈLE DU DOMAINE



4. DIAGRAMME DES CAS D'UTILISATION





5. DIAGRAMME DES CLASSES DE CONCEPTION

6. CONCLUSION

Le développement de notre application a été une entreprise intéressante et enrichissante, mettant en lumière des aspects de la conception logiciel qui nous étaient alors inconnus et révélant des opportunités d'amélioration sur nos connaissances actuelles. Cette conclusion vise à fournir une synthèse des points forts et des points faibles de notre projet.

Points Forts :

Architecture logique:

L'un des principaux points forts de notre application réside dans le design architectural du panneau d'inspection de propriété des objets. Ce dernier a été conçu avec une approche très générale, utilisant la réflexion Java pour permettre l'ajout éventuelle de différents types d'objets sans avoir à toujours faire un nouveau panneau (visuel) pour chaque nouveau type d'objet ou avoir besoin de créer de nouvelles façons de modifier les données dans le *back-end* puisqu'il est créé dynamiquement et utilise toujours la même méthode pour modifier tout type de donnée. Ceci étant dit, Un système connexe permet aussi de créer des panneaux custom pour remplacer les panneaux d'inspection de base, ce qui rend cette fonctionnalité très modulaire et facile d'entretien dans le temps. D'ailleurs, le système de changement de propriété mis en place permet de facilement lancer un évènement custom pour n'importe quel changement.

Dans un contexte d'utilisation entreprise, cet atout nous permettrait de facilement corriger le logiciel selon les besoins d'un client spécifique. De plus, on serait capable d'ajouter de nouvelles fonctionnalités, tel qu'un nouveau type d'accessoire ou une nouvelle façon de faire le toit.

Le UI/UX:

La hiérarchie visuelle (JTree), la disposition des éléments et la facilité de navigation contribuent à une expérience utilisateur optimale, améliorant ainsi l'efficacité globale de l'application. L'ajout des raccourcis standard comme *ctrl+y*, *ctrl-z*, *ctrl+s*. et les petites fonctionnalités comme le déplacement de la grille de positionnement rendent aussi agréable son utilisation.

Points Faibles :

Système d'affichage:

Cependant, le point faible majeur identifié au cours du développement réside dans le système d'affichage. Bien que l'application offre des fonctionnalités robustes du côté du design architectural, le rendu visuel des objets pourrait bénéficier d'améliorations significatives et avoir un design architectural de meilleure qualité permettant d'afficher tout type d'objet au lieu de *hardcoder* le dessin des objets. Il serait impératif d'explorer

des solutions pour optimiser les performances et qui sont plus générales sur un projet à long terme.

Perspectives Futures :

En dépit des points faibles identifiés, le projet global a atteint ses objectifs initiaux et offre une base solide pour des développements futurs. L'énoncé de vision initial est en accordance avec le livrable produit. En priorisant les améliorations du système d'affichage, nous pouvons consolider notre logiciel et offrir une expérience utilisateur plus riche et plus fluide.

7. CONTRIBUTION DES MEMBRES

Pour la remise:

Mise à jour du diagramme de classe : Ramy Ahmed Yahia

Mise à jour du modèle du domaine : Charles Lavoie

Mise à jour du diagramme des cas : Félix-Antoine Diamond

QA : Félix-Olivier Noël

Rédaction du rapport : Charles Viau

Par système programmé :

Charles Viau : Changement de propriété, Property panel, System Entity/Component, Debugging et aide sur multiples systèmes.

Charles Lavoie : Exportation STL, Drawing.

Ramy Ahmed Yahia : Drawing, Sélection, Exportation STL, Back-end du chalet.

Félix-Antoine Diamond : Système de sauvegarde, UI, Conversion Impérial, Tooltip, Hiérarchie d'objet (JTree)

Félix-Olivier Noël : Undo/Redo (Command Pattern), Hiérarchie d'objet (JTree), UI, Zoom, Sélection.