



VirtuTuile : simulateur de pose de revêtement de plancher et de revêtement muraux

Livrable 4 - Génie logiciel orienté objet

Présenté à

Jonathan Gaudreault

Par

Équipe 25

| Matricule | Nom |
|-------------|------------------------|
| 111 237 498 | Laurent Aubin |
| 111 237 153 | Jean-Christophe Drouin |
| 111 156 408 | Toma Gagné |
| 111 239 483 | Vincent Lambert |

Table des matières

| | |
|---|-----------|
| Énoncé de vision | 4 |
| Version finale du projet | 6 |
| Modèle du domaine | 7 |
| Diagramme du domaine | 7 |
| Texte explicatif | 7 |
| Diagramme des cas d'utilisation | 9 |
| Textes des cas d'utilisation | 10 |
| Diagramme de classe de conception | 22 |
| Diagramme de séquence de conception | 28 |
| Détermination de la surface sélectionnée | 28 |
| Travail réalisé pour déterminer les coordonnées x, y | 28 |
| Traitement permettant de déterminer quelle surface a été sélectionnée | 29 |
| Création d'une surface rectangulaire | 32 |
| Affichage | 33 |
| Contribution | 38 |

Énoncé de vision

Peu importe l'envergure d'un projet, la planification est sans équivoque un facteur augmentant considérablement les chances de réussite. Une planification structurée permet notamment d'obtenir des estimations sur les dépenses et les délais à prévoir. Cela entraîne une meilleure gestion des ressources disponibles, une meilleure préparation face aux imprévus et le respect du budget et des échéanciers. Prenons l'exemple d'une compagnie, d'une équipe ou d'une simple personne possédant le mandat d'effectuer le revêtement de surface à l'aide de tuiles, que cela soit un mur de cuisine, de salle de bain ou même un plancher. Évidemment, il serait intéressant d'avoir une représentation visuelle de la pièce et de ses surfaces une fois recouverte, ainsi que la quantité de matériaux et les coûts à prévoir.

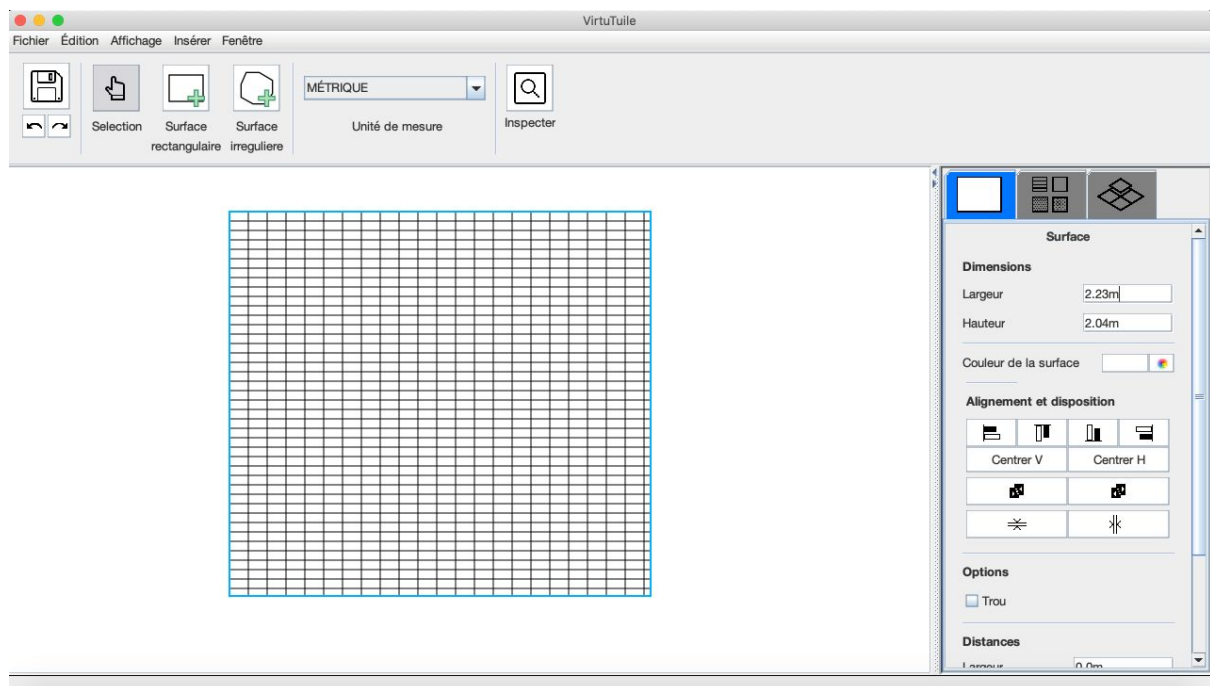
C'est ici que VirtuTuile entre en jeu. VirtuTuile est une application novatrice qui répond exactement à ce besoin. Un utilisateur de l'application peut modéliser les surfaces d'une pièce tout en respectant ses dimensions en taille réelle. À partir de ce modèle, l'utilisateur possède carte blanche sur le revêtement des surfaces qu'il a créées.

D'abord, il est invité à choisir un modèle de tuile. Une tuile est définie par son matériau, ses dimensions et sa couleur. Toutes ces caractéristiques peuvent être configurées selon les préférences de l'utilisateur. Une fois un modèle de tuile choisi, l'utilisateur peut choisir le motif, c'est-à-dire la disposition des tuiles entre elles, sur la surface ou les surfaces sélectionnées. L'application propose aussi la modification de la largeur et de la couleur des joints entre les tuiles pour obtenir une représentation réaliste des goûts et des préférences de l'utilisateur. Lors de l'application du motif sur les surfaces sélectionnées, l'utilisateur peut choisir entre un motif débutant avec une tuile pleine et un motif débutant par une tuile centrée, permettant une coupe uniforme des tuiles se trouvant aux extrémités. En tout temps, l'utilisateur peut modifier les paramètres initialement entrés dans l'application et voir l'effet des modifications en direct. S'il n'est pas satisfait du résultat de ses dernières modifications, il peut facilement retourner aux configurations précédentes à l'aide de la fonctionnalité de retour en arrière prévue par VirtuTuile.

Une fois les surfaces recouvertes et que l'utilisateur est satisfait de ce qu'il voit, celui-ci peut demander à l'application de calculer la quantité de tuiles nécessaire à la réalisation du projet et du coût à prévoir. Comme il s'agit d'une simulation, VirtuTuile s'assure que le projet peut réellement être réalisé. Pour ce faire, VirtuTuile possède un inspecteur permettant de repérer les tuiles physiquement trop petites pour une coupe dans le monde réel. Cet inspecteur peut aussi détecter des discontinuités dans la disposition des tuiles. Cet inspecteur agit alors comme un superviseur de projet, permettant une pose de revêtement optimisé et réaliste pour l'utilisateur.

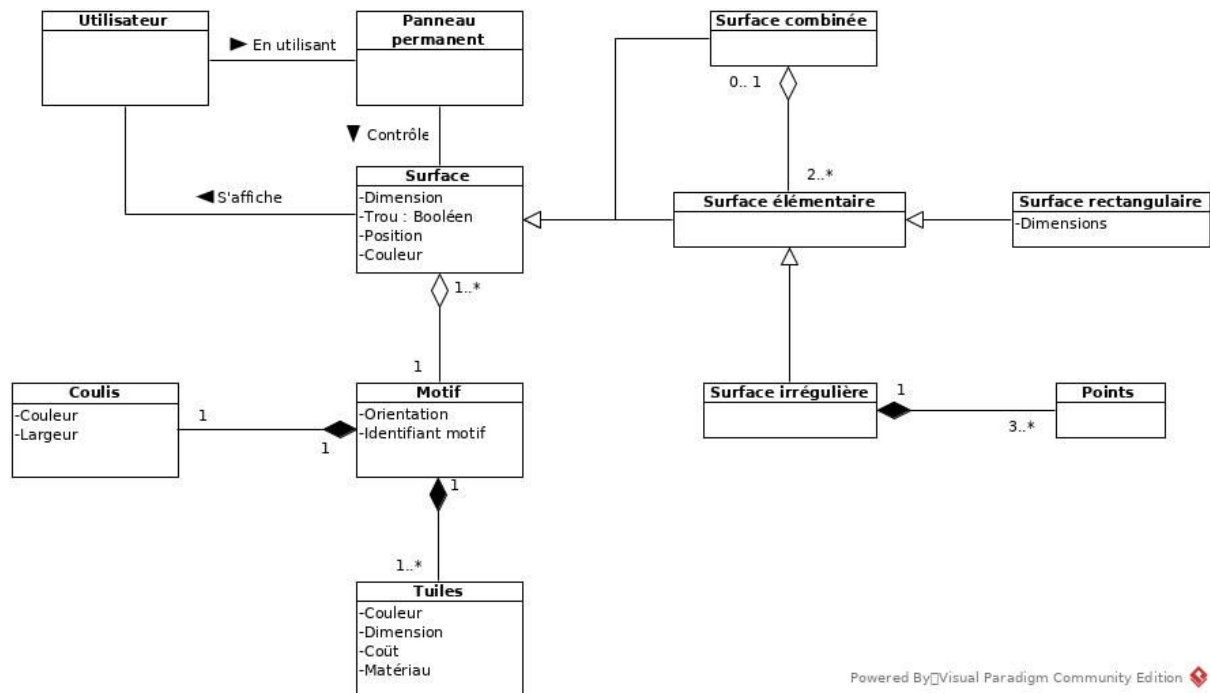
Malgré toutes ces fonctionnalités, VirtuTuile reste une application simple d'utilisation par son interface intuitive et structurée. Elle a le potentiel de devenir un outil essentiel à tous ceux désirant faire la pose de revêtement le plus efficacement possible.

Version finale du projet



Modèle du domaine

Diagramme du domaine



Powered By Visual Paradigm Community Edition

Texte explicatif

Utilisateur : utilise le panneau permanent pour choisir ses préférences pour son projet de rénovation.

Panneau permanent : permet à l'utilisateur de modifier les caractéristiques des surfaces, des motifs, des tuiles et du coulis

Surface : une surface est d'une dimension/forme et d'une couleur choisie par l'utilisateur. Cette surface possède une position précise sur l'interface utilisateur. La surface peut contenir

des trous qui représentent des endroits à ne pas recouvrir. Une surface peut être simple (élémentaire) ou combinée, selon les besoins de l'utilisateur.

Surface élémentaire: une surface élémentaire est une surface simple. Cette surface peut être d'une forme rectangulaire ou d'une forme irrégulière. La forme de la surface est déterminée par l'utilisateur.

Surface combinée: une surface combinée est la fusion de plusieurs surfaces élémentaires. Cette combinaison permet la création d'une toute nouvelle surface.

Surface rectangulaire: une surface rectangulaire est un type de surface élémentaire qui possède des dimensions définies par l'utilisateur.

Surface irrégulière: une surface irrégulière est un type de surface élémentaire ayant une forme asymétrique. Cette surface est définie par ses sommets. Ainsi, une surface irrégulière est constituée de plusieurs points.

Points: un point est un sommet d'une surface irrégulière. La combinaison de ces points permet de définir la forme de la surface.

Coulis : se retrouve sur la surface entre les tuiles. Le coulis possède une couleur et une largeur choisies par l'utilisateur.

Motif : la disposition est choisie par l'utilisateur et c'est de cette façon que les tuiles vont être disposées sur la surface.

Tuiles : les tuiles sont choisies par l'utilisateur et elles se retrouvent sur la surface avec une certaine disposition. Les tuiles ont aussi une couleur et des dimensions données par l'utilisateur, ainsi qu'un coût.

Diagramme des cas d'utilisation

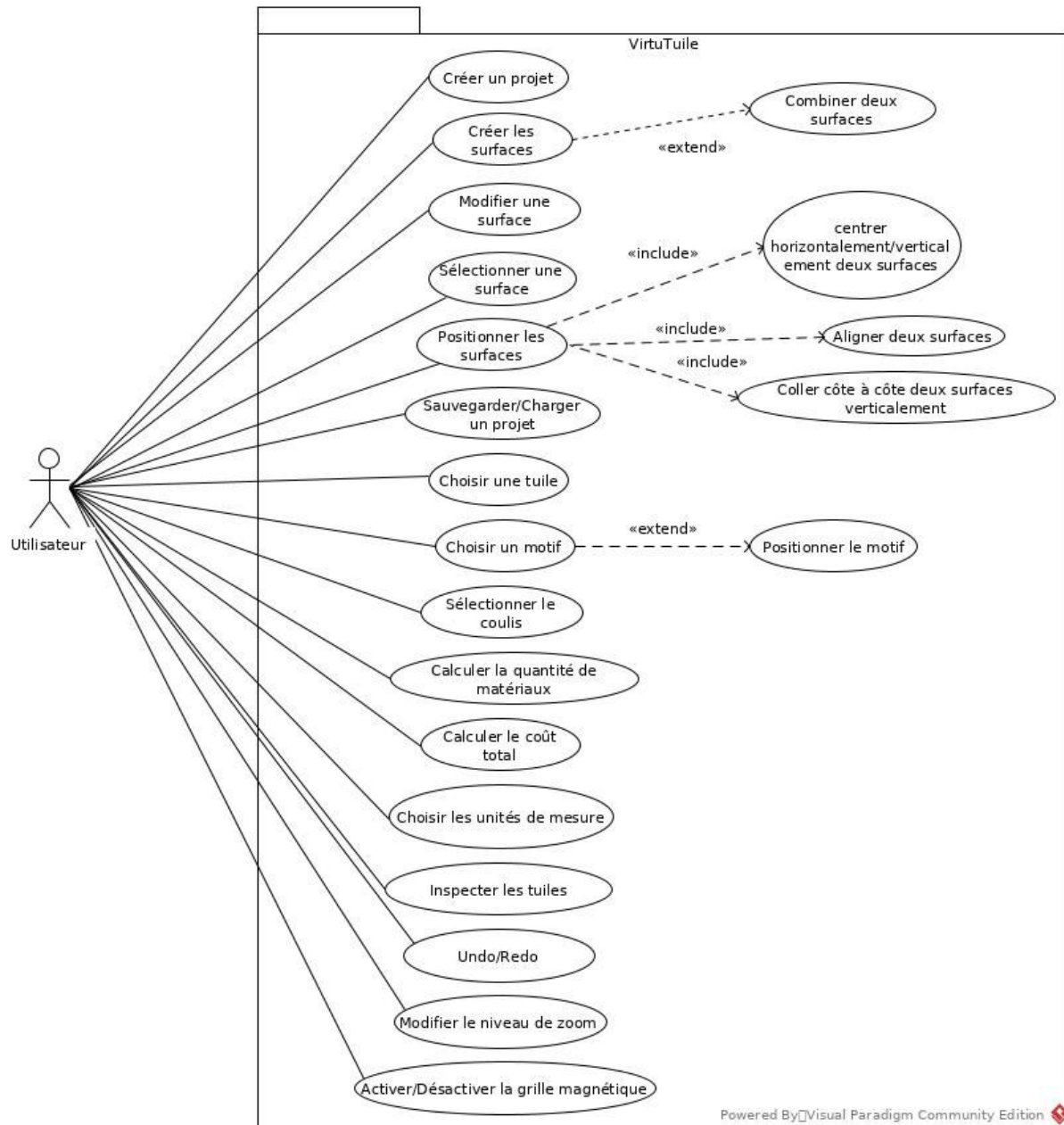
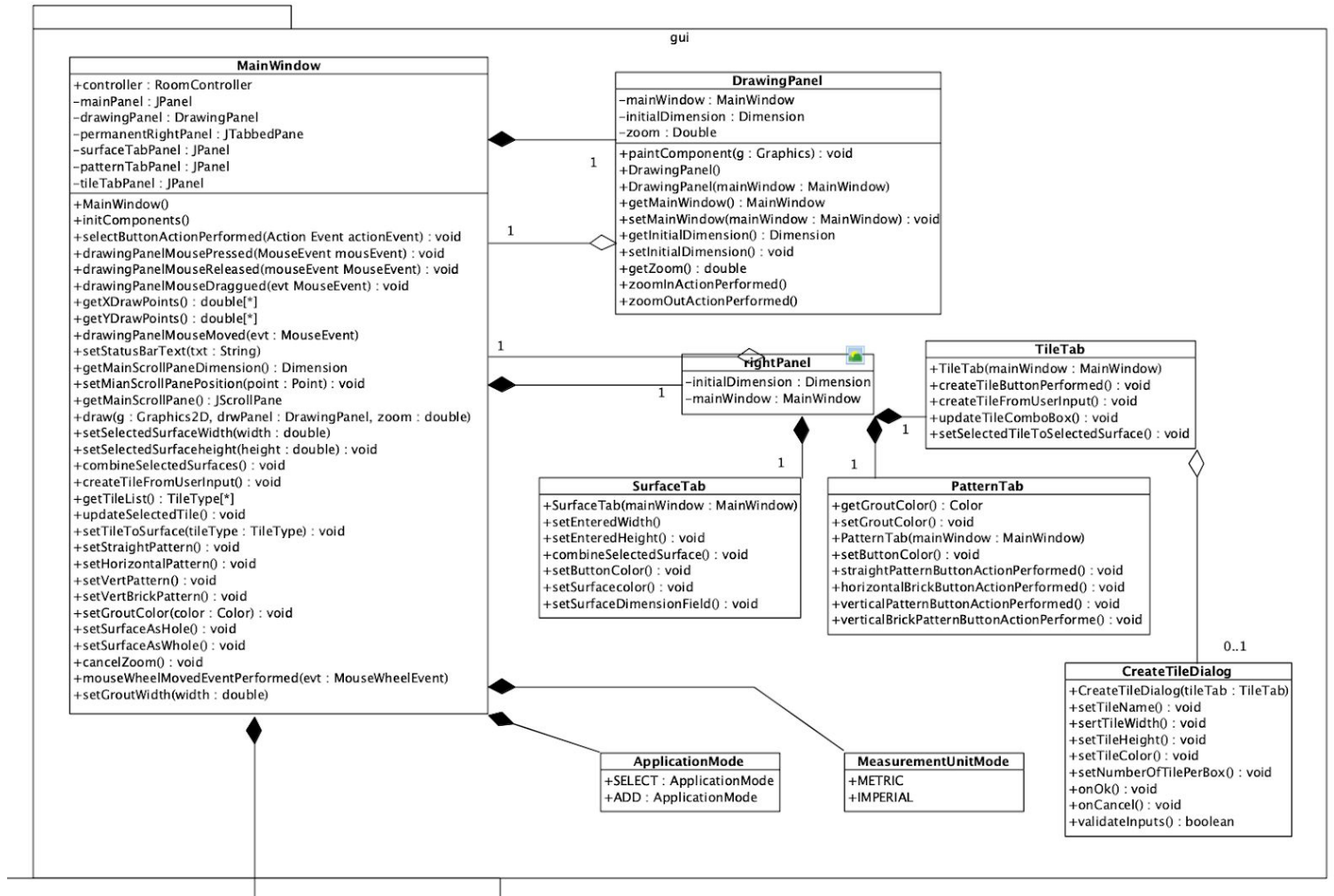
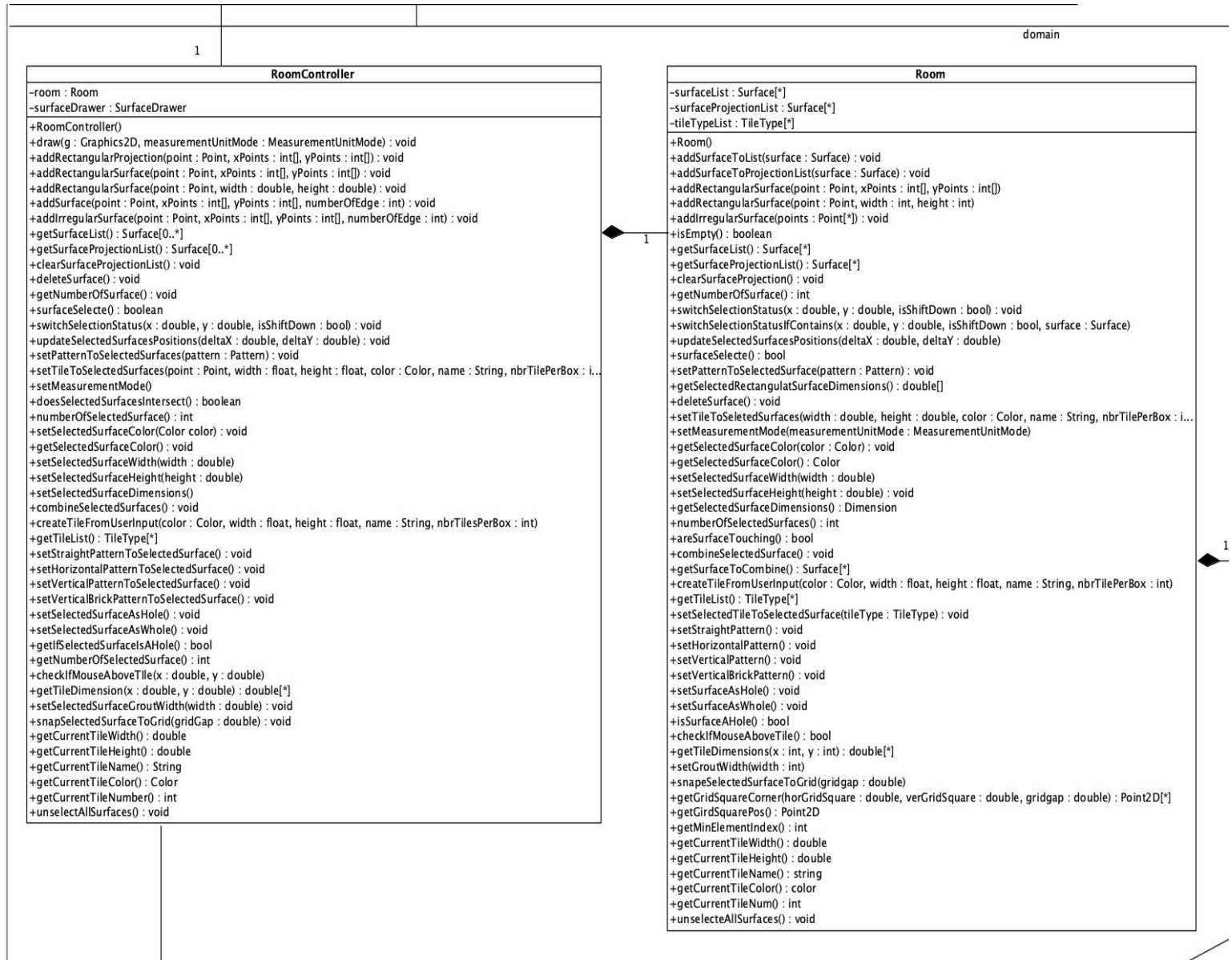


Diagramme de classe de conception

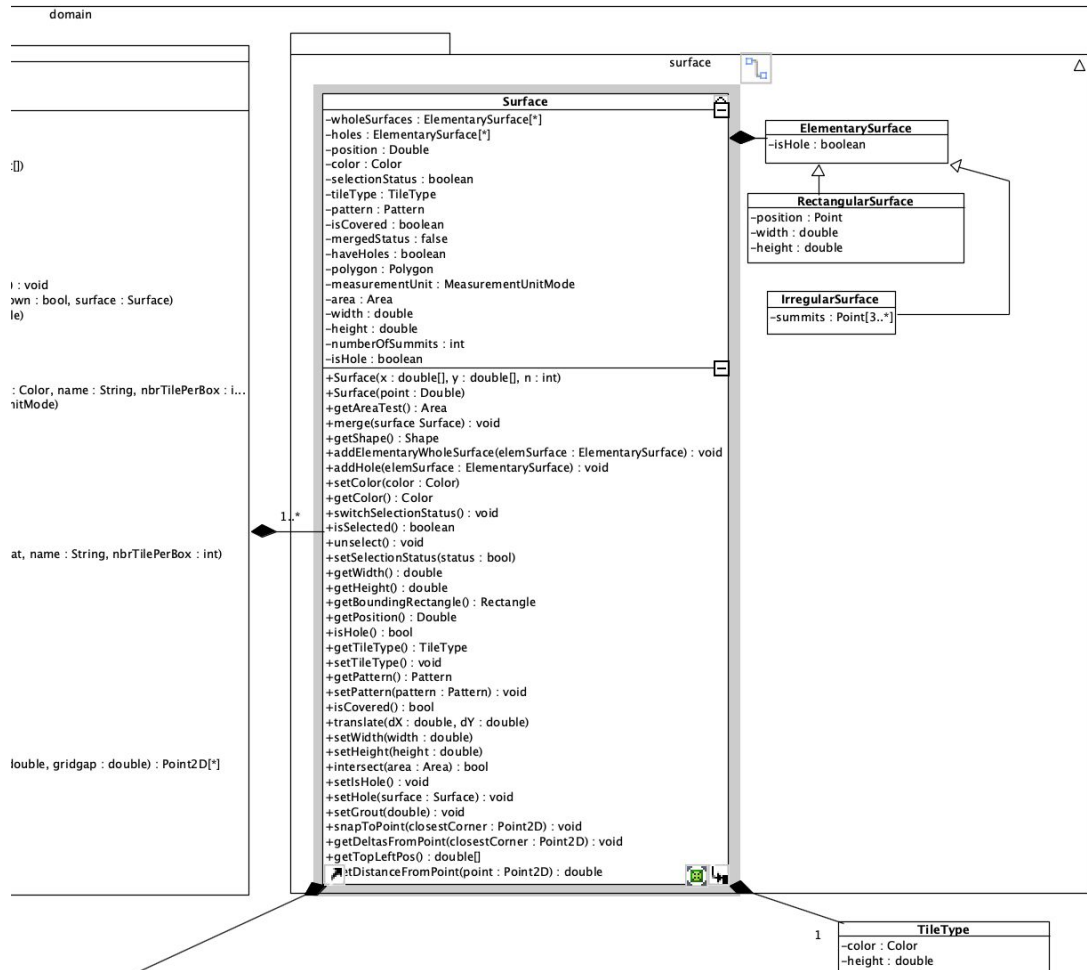






1

1



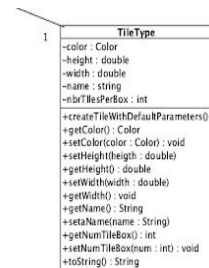
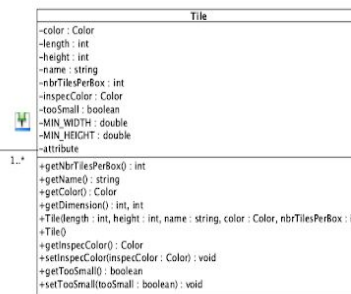
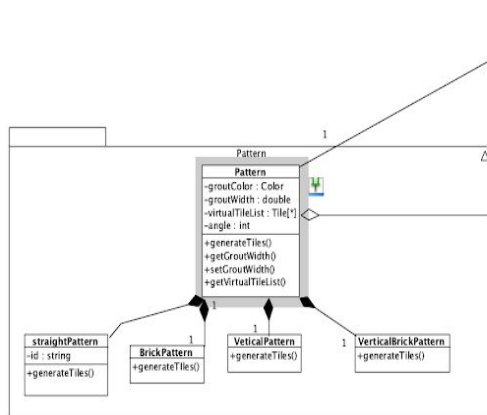
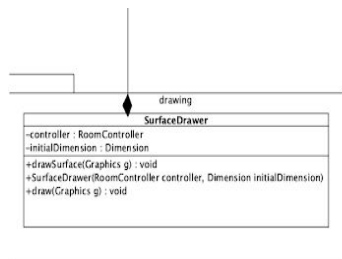
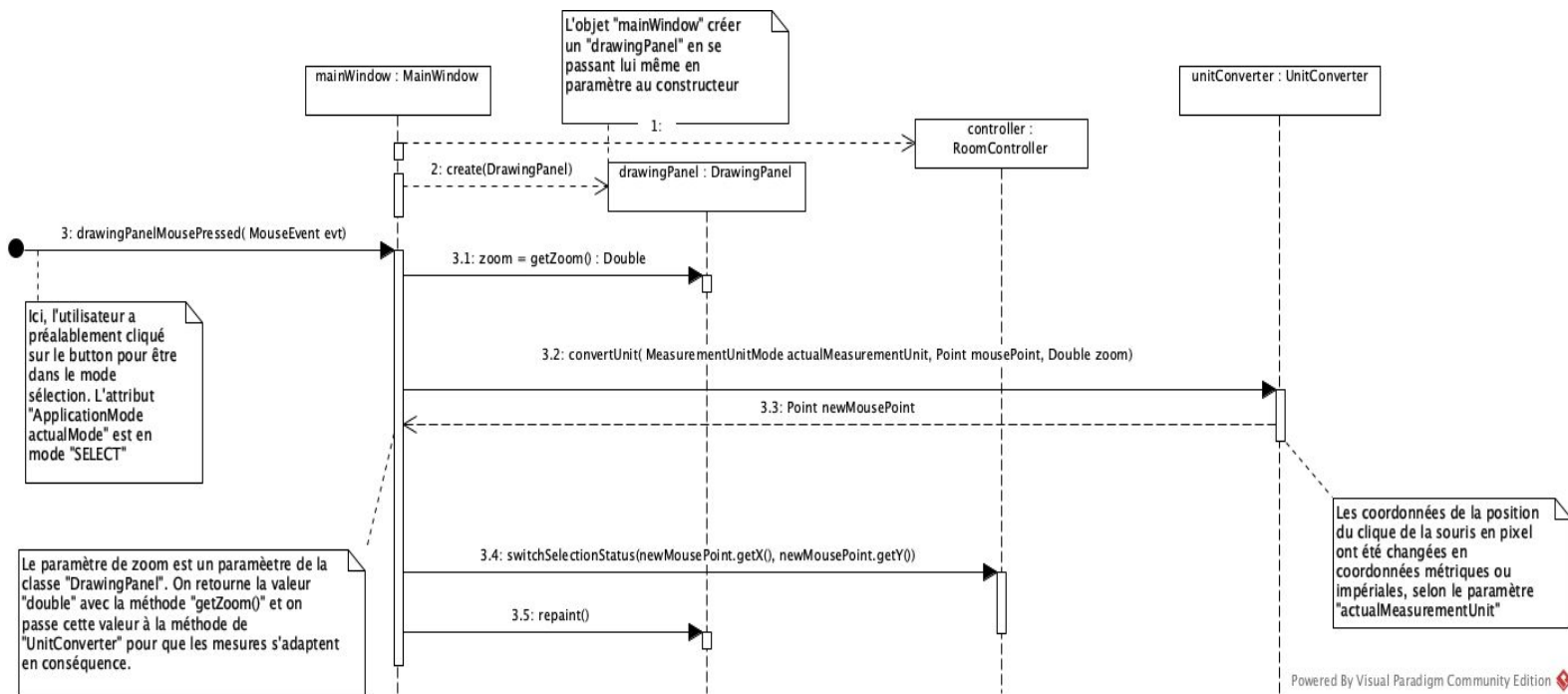


Diagramme de séquence de conception

Détermination de la surface sélectionnée

Travail réalisé pour déterminer les coordonnées x, y

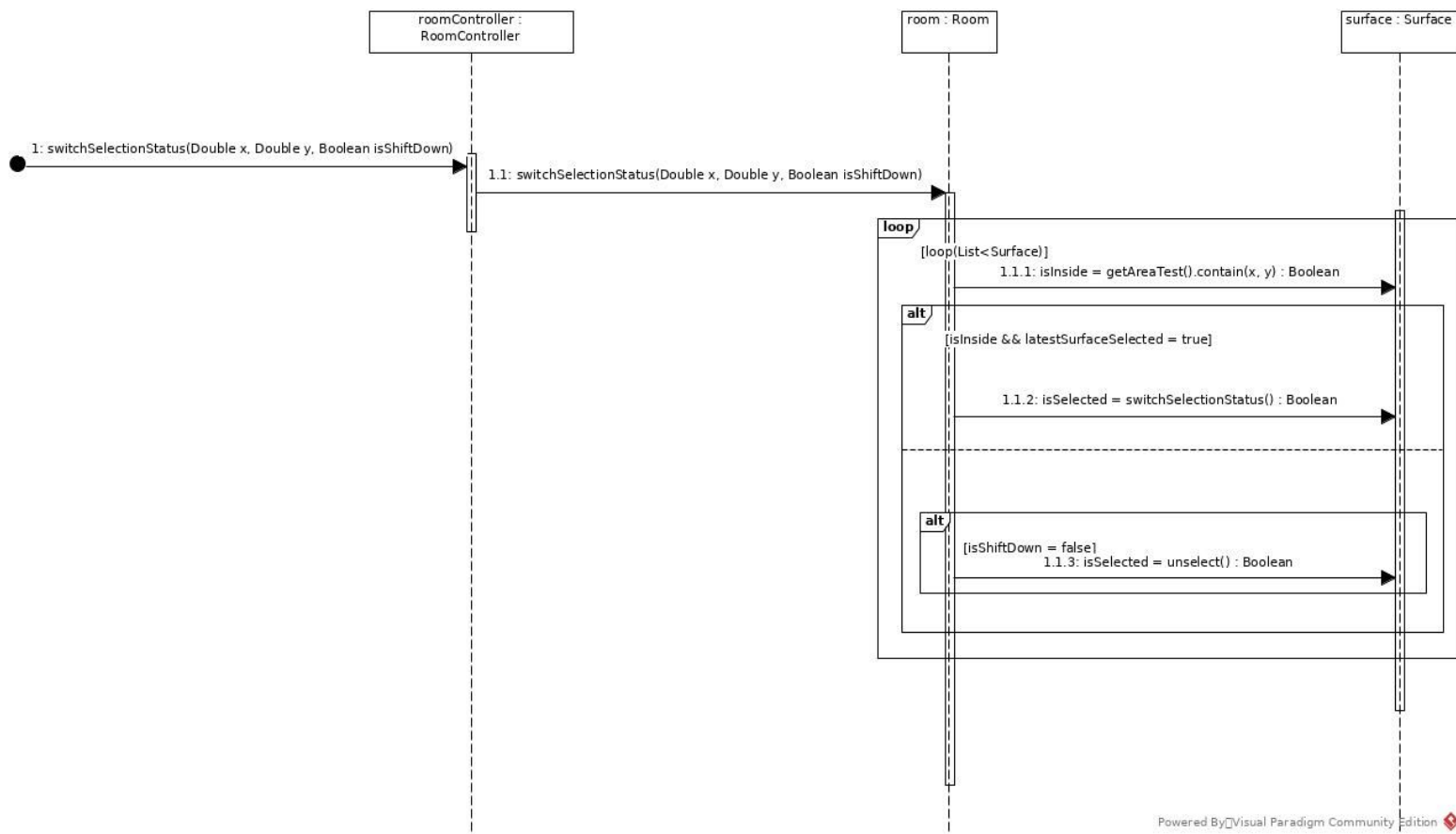


Texte explicatif

Le clic fait par l'utilisateur est fait dans le «drawingPanel» qui a été préalablement initialisé par «mainWindow» à l'exécution du programme. Pour reconnaître qu'il y a eu un clic, un «MouseListener» est ajouté au «drawingPanel» qui exécute une méthode appelée «drawingPanelMousePressed» qui est implémentée dans la classe «MainWindow».

Pour faire la conversion de la position du clic en unité de mesure, on appelle la méthode «getZoom()» qui retourne le zoom actuel du «drawingPanel». Le zoom va permettre d’avoir notre position réelle relative à notre point de référence en fonction du zoom. Une fois le zoom obtenu, nous créons un objet Point dont les coordonnées sont les coordonnées x et y du MouseEvent divisé par le zoom.

Traitement permettant de déterminer quelle surface a été sélectionnée



Texte explicatif

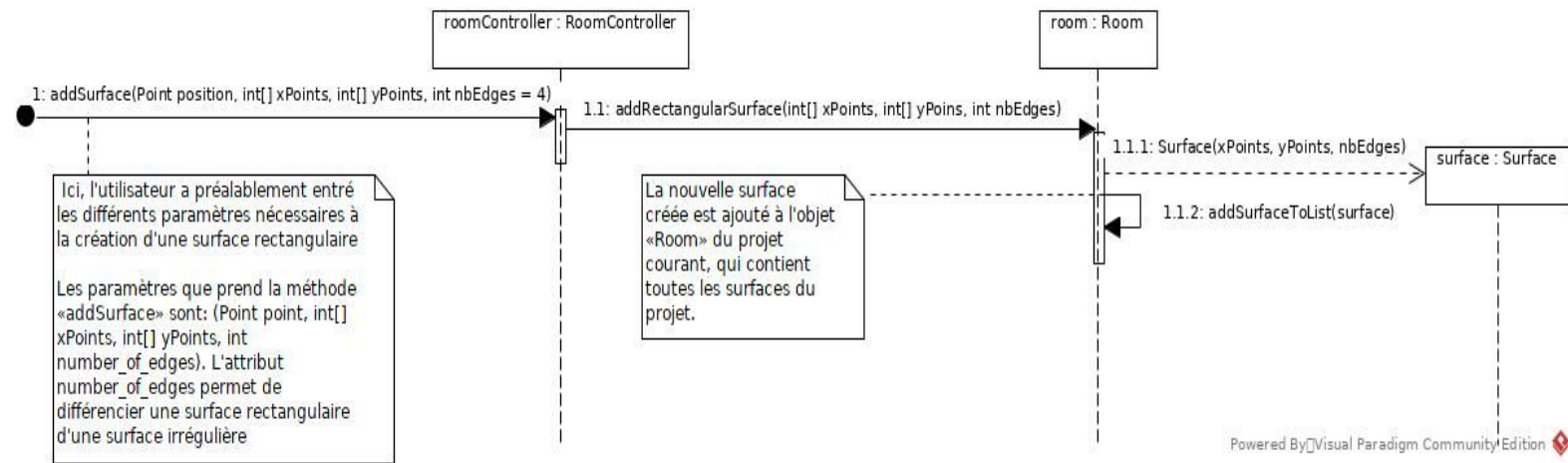
Ce diagramme est la suite du précédent. On commence ici après que l'utilisateur ait cliqué dans le panneau d'affichage «drawingPanel» et après que les coordonnées du clic de la souris ont été converties en unité de mesure x, y. La prochaine étape est l'appel de la méthode «switchSelectionStatus» avec les paramètres x, y de la classe «Room». La classe «Room» contient un argument de type «List<Surface>», qui contient tous les objets surfaces du projet actuel.

Pour chacune des surfaces dans «Room», on vérifie si les coordonnées «x, y» se trouve à l'intérieur de la surface. Pour ce faire, on fait appel à la méthode «getArea()» qui retourne un objet Area propre à la surface. Puisque la classe Area implémente l'interface Shape, il est ensuite possible d'appeler la méthode «contains()» avec les coordonnées de la souris en paramètre. La méthode «contains» retourne un booléen indiquant si les coordonnées font partie de la surface.

Si la méthode «contains» retourne «false», la méthode «unselect()» de la surface est appelée. Celle-ci permet d'attribuer le booléen «False» à l'attribut «selectionStatus», ce qui désélectionne la surface impliquée. À l'inverse, si la méthode retourne «true», alors la méthode traverse une dernière condition. Celle-ci implique la touche «Shift» du clavier. En effet, si la touche «Shift» est enfoncée, la méthode «switchSelection» de la classe Surface est appelée pour la surface en question. Son argument «selectionStatus» est alors changé pour «true» (l'argument est initialement égal à «false» lors de la création d'une surface). Dans le cas contraire, si la touche «Shift» n'est pas enfoncée, la surface est désélectionnée à l'aide de la méthode «unselect()».

Lors de l'affichage des surfaces, la classe «SurfaceDrawer» prendra en compte l'argument «selectionStatus» de chacune des surfaces et les surfaces sélectionnées seront mises en évidence.

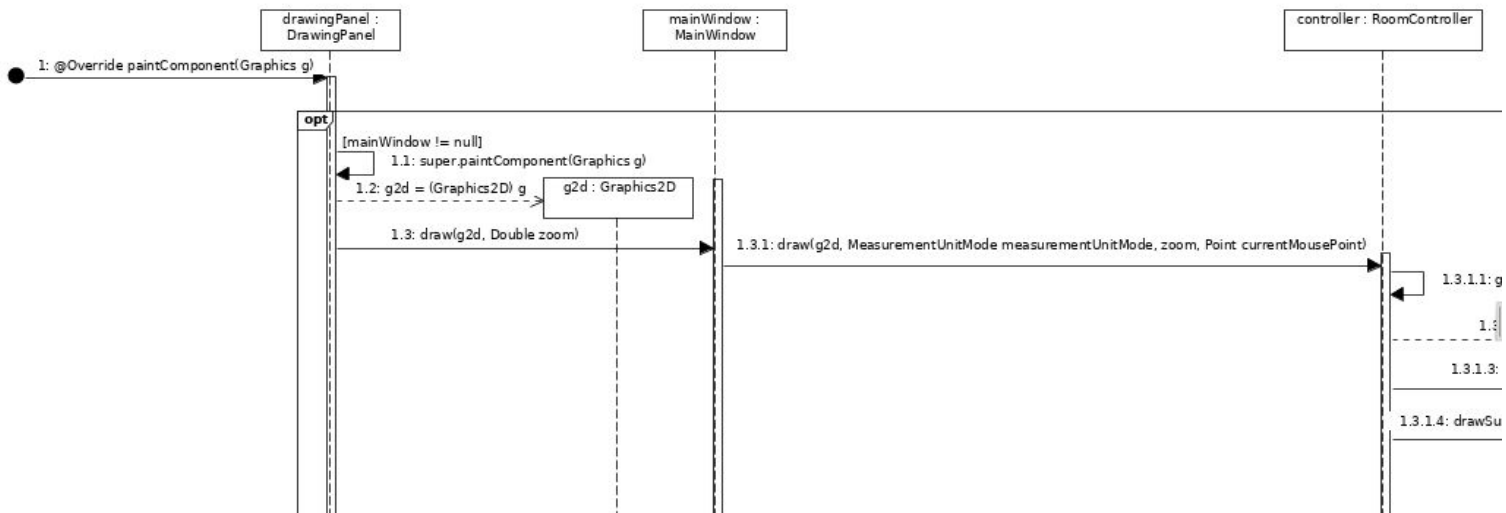
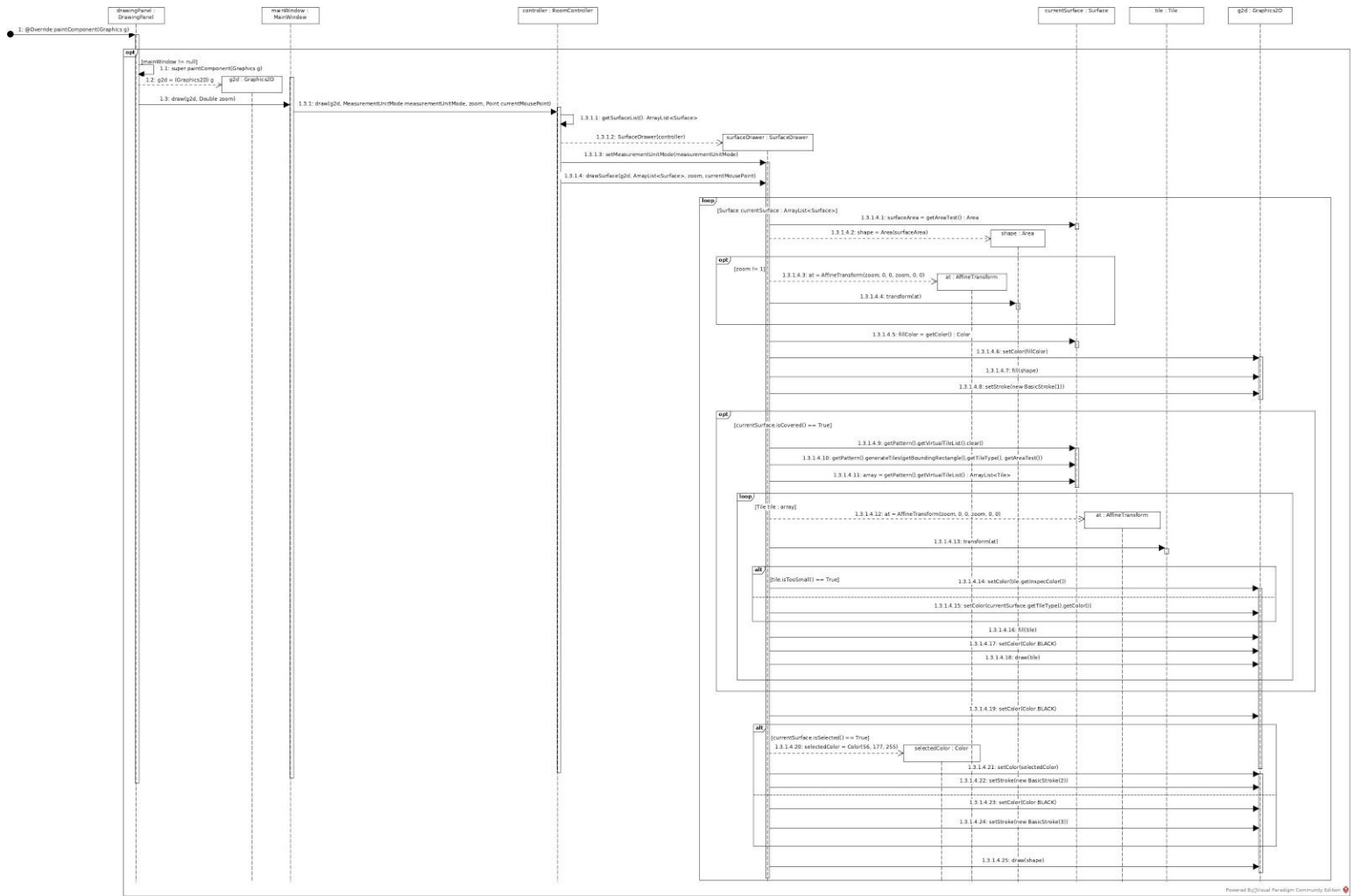
Création d'une surface rectangulaire

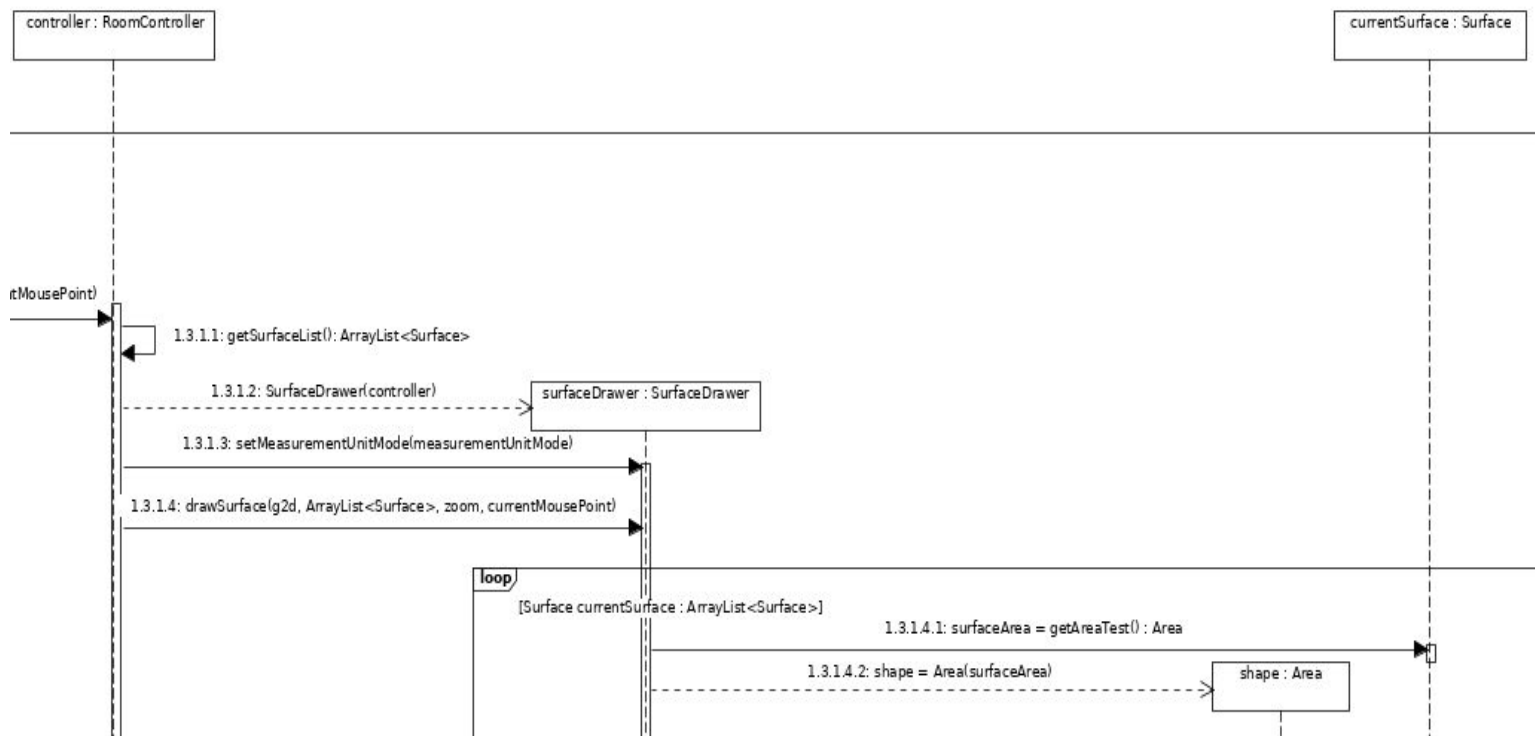


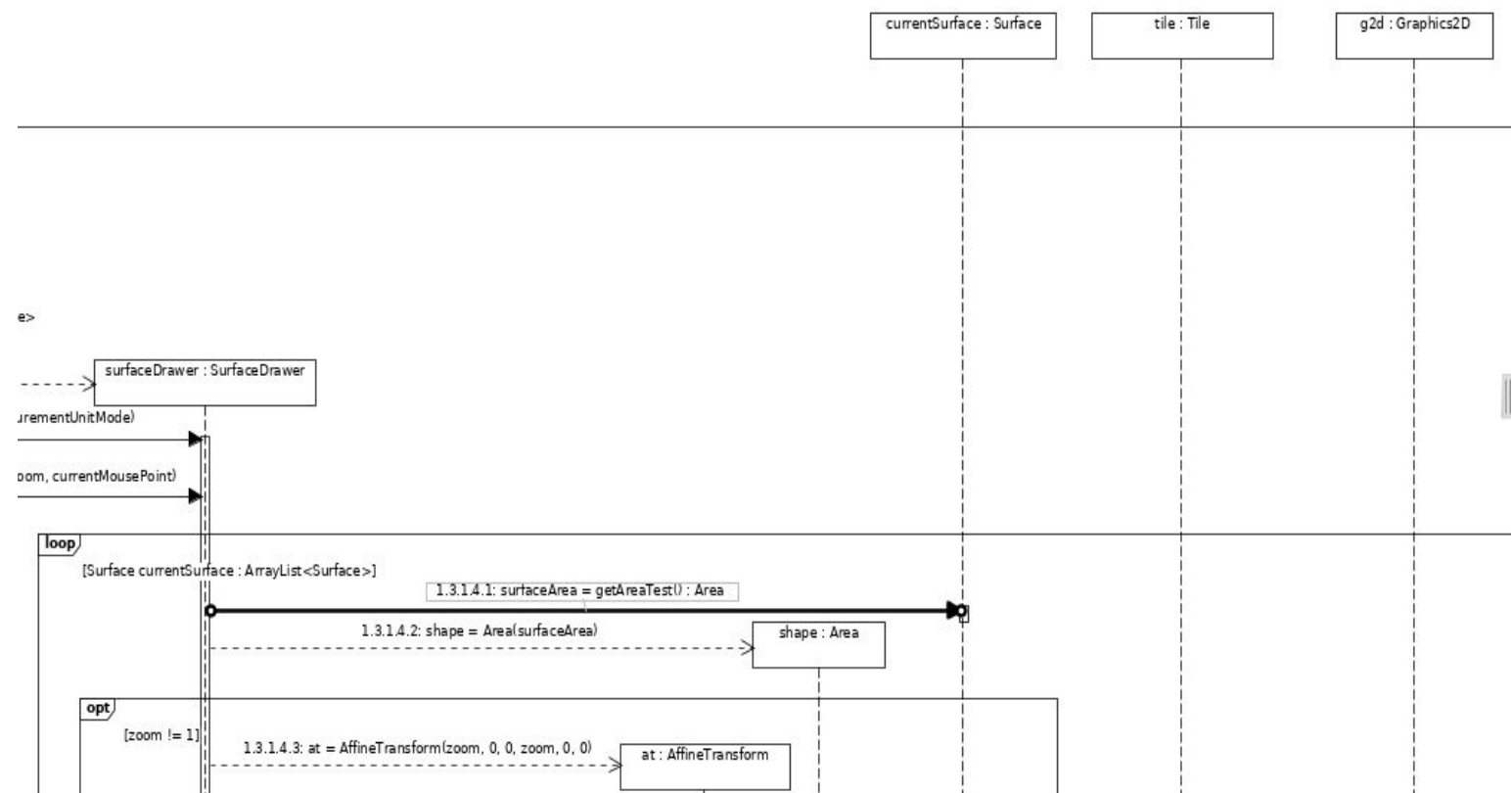
Texte explicatif

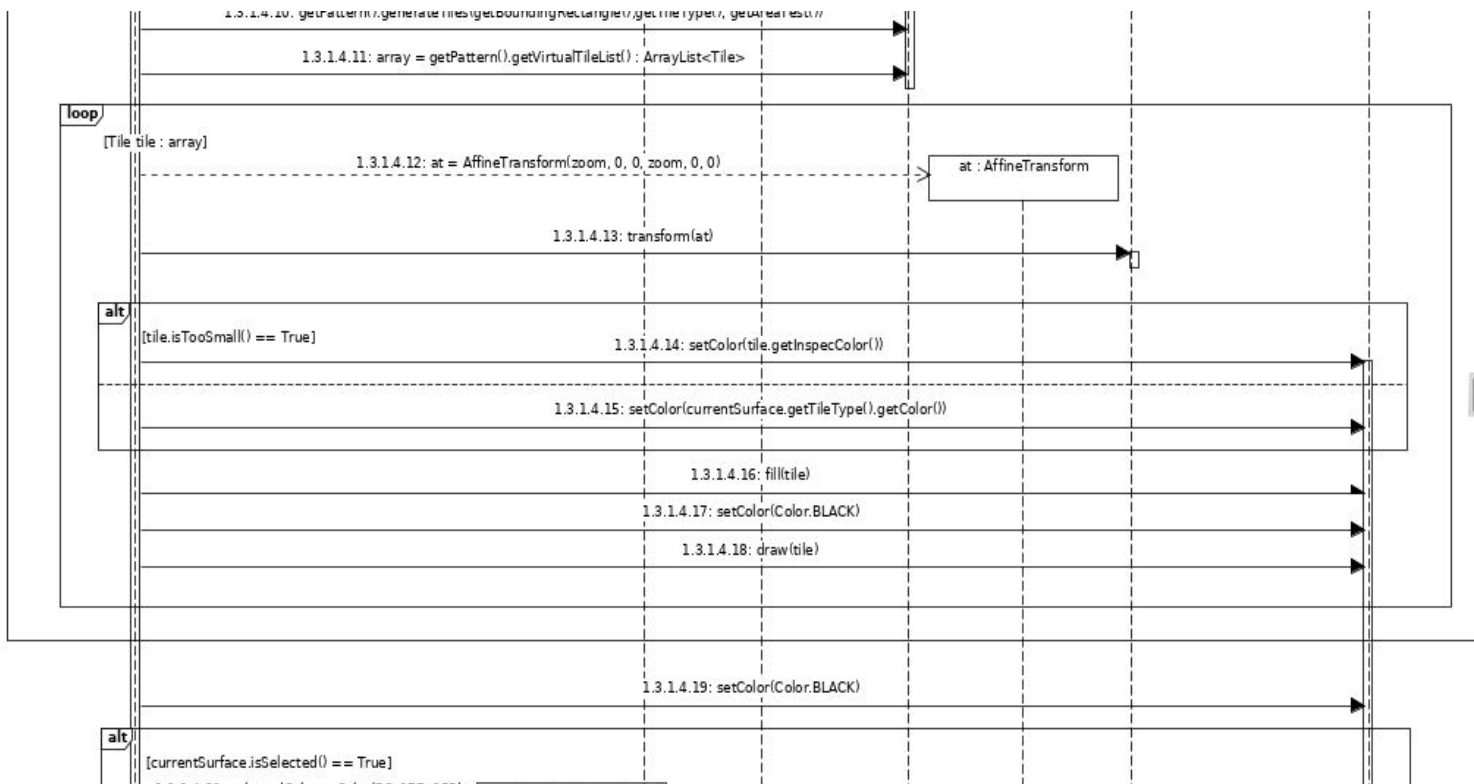
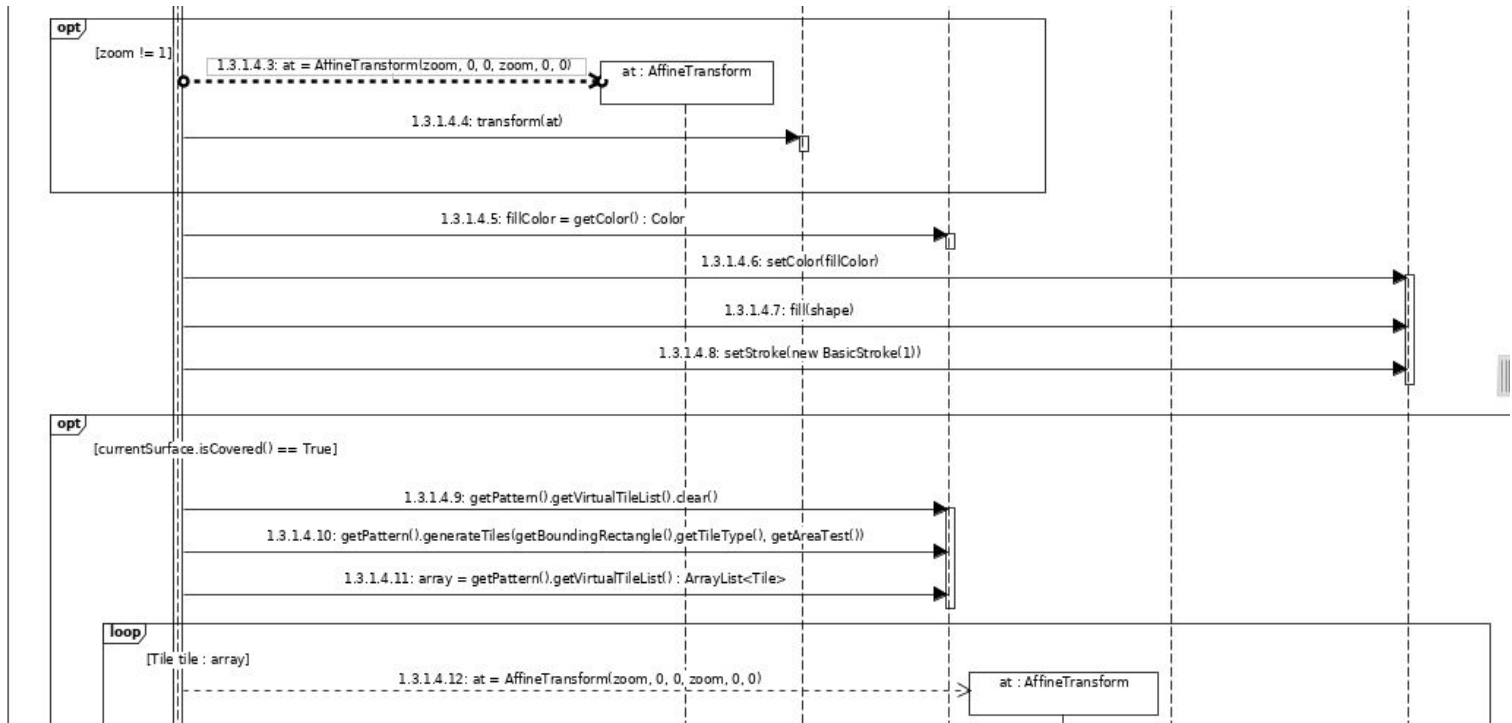
Ici, on considère que l'utilisateur a préalablement entré toutes les informations nécessaires à la création d'une nouvelle surface rectangulaire. L'appel de la méthode «addSurface» du contrôleur entraîne la création d'un objet «Surface», qui appelle le bon constructeur selon les paramètres qui ont été entrés. Ces paramètres représentent les coordonnées des sommets ainsi que le nombre de sommets de la surface. Le contrôleur ajoute ensuite la nouvelle surface à l'objet «Room» du projet actuel à l'aide la méthode «addSurfaceToList» de «Room». Cette dernière méthode ne fait qu'ajouter la nouvelle surface à l'argument «surfaceList» de «Room», qui est une liste qui contient toutes les surfaces du projet actuel.

Affichage











Texte explicatif

La méthode «`paintComponent(Graphics g)`» de la classe «`DrawingPanel`» permet d'abord d'appeler la méthode parent «`paintComponent(Graphics g)`». Puis, un objet de la classe «`Graphics2D`» est créé et passé en paramètre lors de l'appel à la méthode «`draw(g2d, Double zoom)`» de `mainWindow`. Cette méthode permet d'appeler la fonction «`draw(g2d, MeasurementUnitMode measurementUnitMode, zoom)`» du contrôleur. De cette manière, il sera possible de procéder à l'affichage sous l'unité de mesure désirée. Le contrôleur accède à la liste de «`Surface`» de l'objet «`Room`» à l'aide de la méthode «`getSurfaceList()`». Ensuite, une instance de la classe «`SurfaceDrawer`» est créée avec le contrôleur en paramètre. On y applique l'unité de mesure désirée via la fonction «`setMeasurementUnitMode`» et on appelle la méthode «`draw(g2d, ArrayList<Surface>, zoom)`». La fonction «`drawSurface(g2d, ArrayList<Surface>, zoom)`» de «`SurfaceDrawer`» permet de dessiner dans le «`DrawingPanel`» l'ensemble des surfaces comprises dans la liste de l'objet «`Room`». Les éléments d'une

surface, soit le motif, les tuiles ou le coulis seront dessinés en respectant le niveau de zoom de l'utilisateur.

Points forts et points faibles

Points forts

Le point fort principale de l'application est l'expérience utilisateur qu'elle offre. L'utilisation est intuitive et l'interface démontre bien qu'il y a eu de l'effort pour la présentation.

Autre que ce point, nous considérons que tous les autres éléments de l'application pourraient être retravaillés pour que le tout soit parfait

Points faibles

L'application fait l'arrondissement des mesures entrées. Selon la valeur entrée, on peut perdre jusqu'à 0.08 pouces de précision, ce qui n'est pas optimal pour une application où les mesures sont importantes.

De plus, notre application ne supporte pas l'affichage de mesures très petites. Par exemple, il n'est pas possible de mettre un largeur de coulis de 1mm et de le voir à l'affichage.

Les algorithmes pour la génération des tuiles de certains motifs sont aussi assez gourmands. L'application génère tous les motifs à chaque "tick". Si de très petites tuiles sont dessinées ou si on affiche de très grande surface, il est possible d'apercevoir des baisses de performance de l'application.

Contribution

Les membres de l'équipe ont travaillé sur l'ensemble du livrable. Nous avons tous travaillé sur le développement du projet, ainsi que sur le rapport. De plus, nous nous sommes rencontrés à plusieurs occasions où nous avons bien travaillé sur l'avancement du projet.

Ainsi, tous les membres de l'équipe ont contribué au succès du livrable 4.