



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

Livrable 3 - Génie logiciel orienté objet

Présenté à

Jonathan Gaudreault

Par

Équipe 25

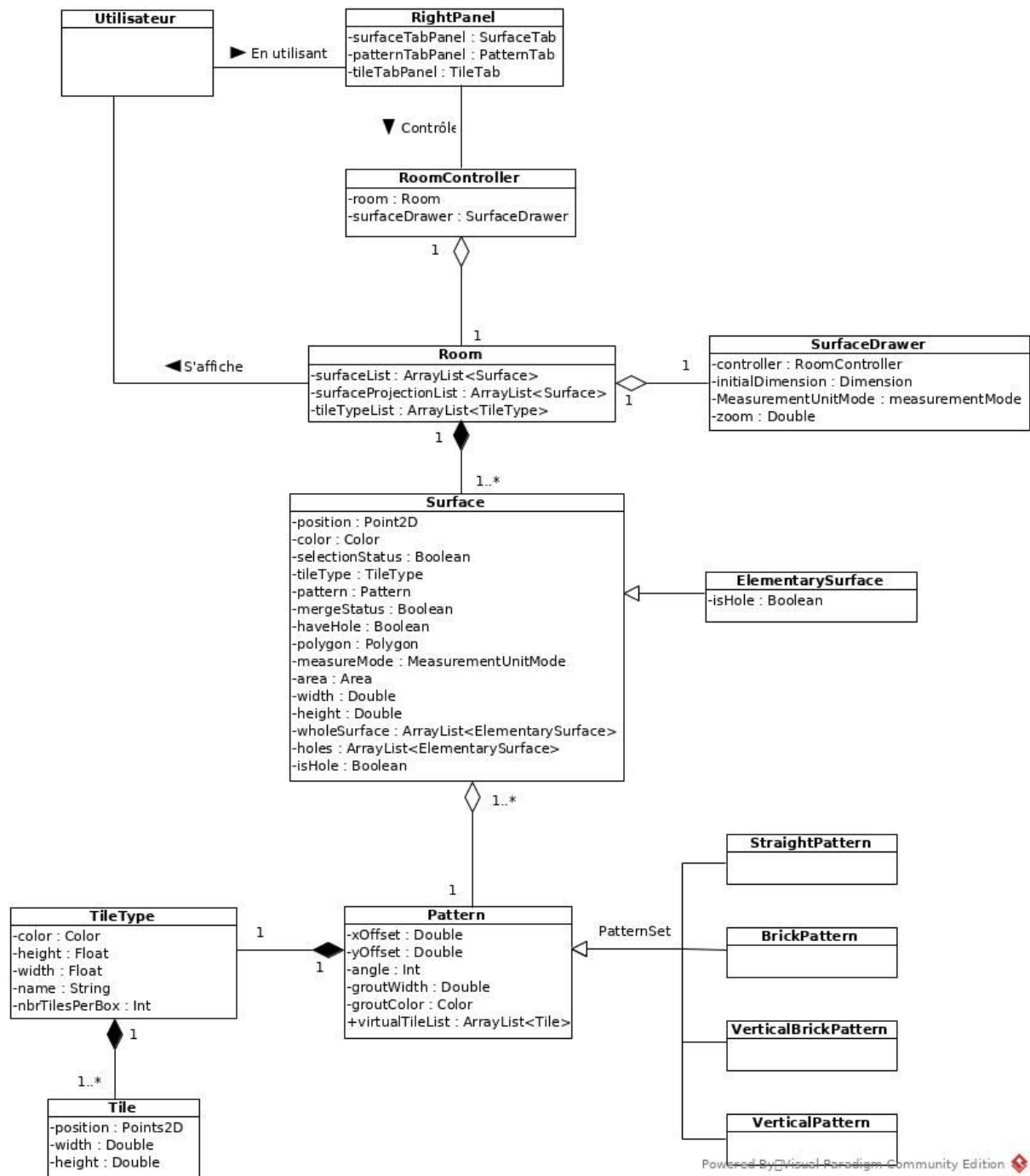
Matricule	Nom
111 237 498	Laurent Aubin
111 237 153	Jean-Christophe Drouin
111 156 408	Toma Gagné
111 239 483	Vincent Lambert

Table des matières

Modèle du domaine	3
Diagramme du domaine	3
Texte explicatif	4
Diagramme des cas d'utilisation	6
Diagramme de classe de conception	7
Architecture logique	8
Diagramme de séquence de conception	10
Détermination de la surface sélectionnée	10
Travail réalisé pour déterminer les coordonnées x, y	10
Traitement permettant de déterminer quelle surface a été sélectionnée	12
Création d'une surface rectangulaire	14
Affichage	15

Modèle du domaine

Diagramme du domaine



Texte explicatif

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

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

Contrôleur (RoomController) : les modifications d'une pièce demandées par l'utilisateur passent à travers le contrôleur qui répartit les demandes aux méthodes appropriées.

Pièce (Room) : l'ensemble des surfaces générées par l'utilisateur sont maintenues dans une pièce.

Dessin Surface (SurfaceDrawer) : les surfaces contenues dans la pièce sont dessinées à l'aide de cet outil.

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 (ElementarySurface) : une surface élémentaire est une surface simple. Cette surface peut être un trou ou une surface d'une forme rectangulaire ou d'une forme irrégulière. La forme de la surface est déterminée par l'utilisateur.

Motif (Pattern) : la disposition est choisie par l'utilisateur et c'est de cette façon que les tuiles vont être disposées sur la surface. Le motif gère également la couleur et la largeur du coulis séparant les tuiles. Les motifs «droit» et «brique» sont disponibles ainsi que leurs motifs inversés respectifs.

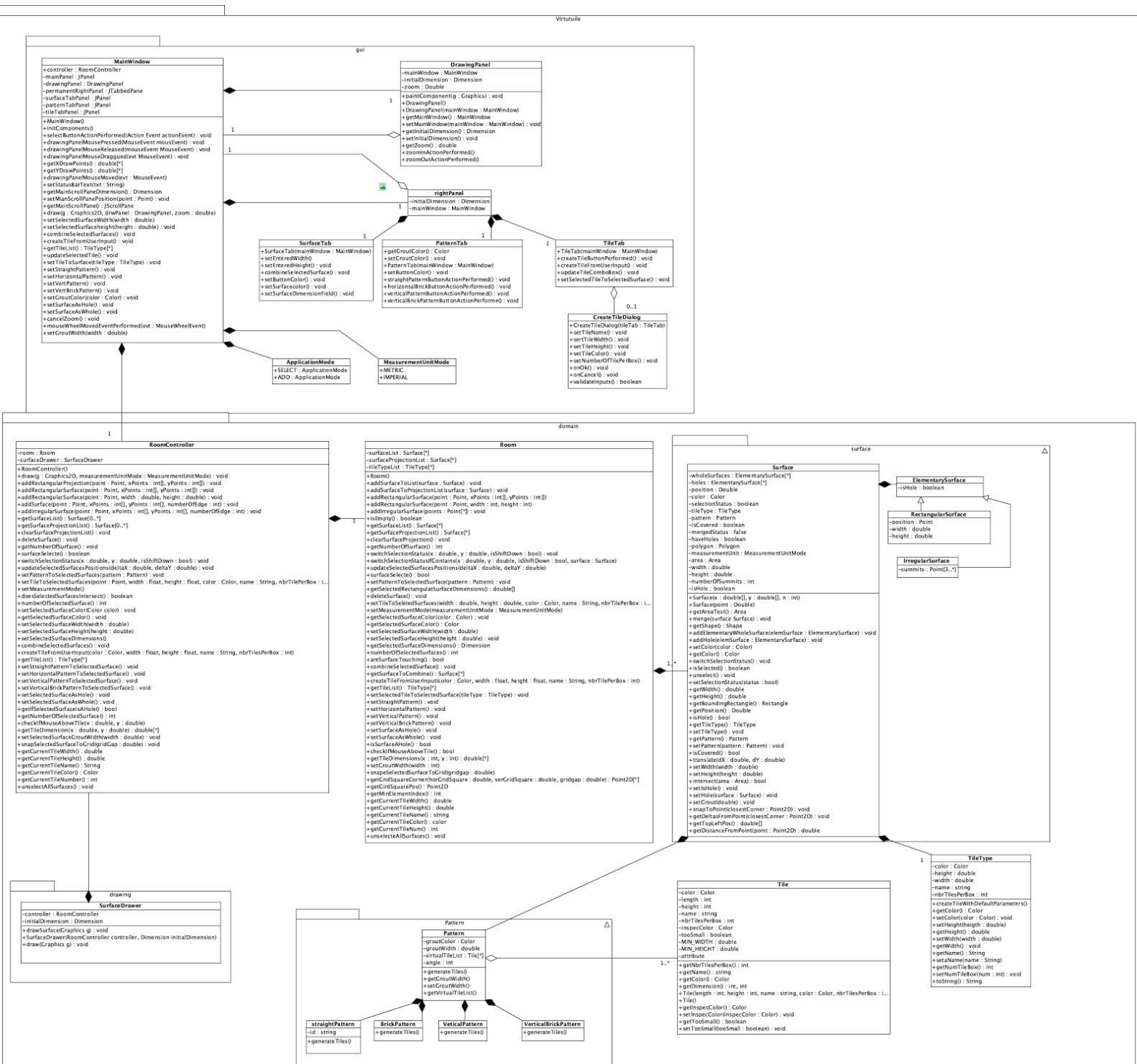
TypeTuile (TileType) : le type de tuile permet à l'utilisateur de créer un type de tuile ayant une dimension, une couleur et un nom précis. Ce type peut être réutilisé à d'autres surfaces.

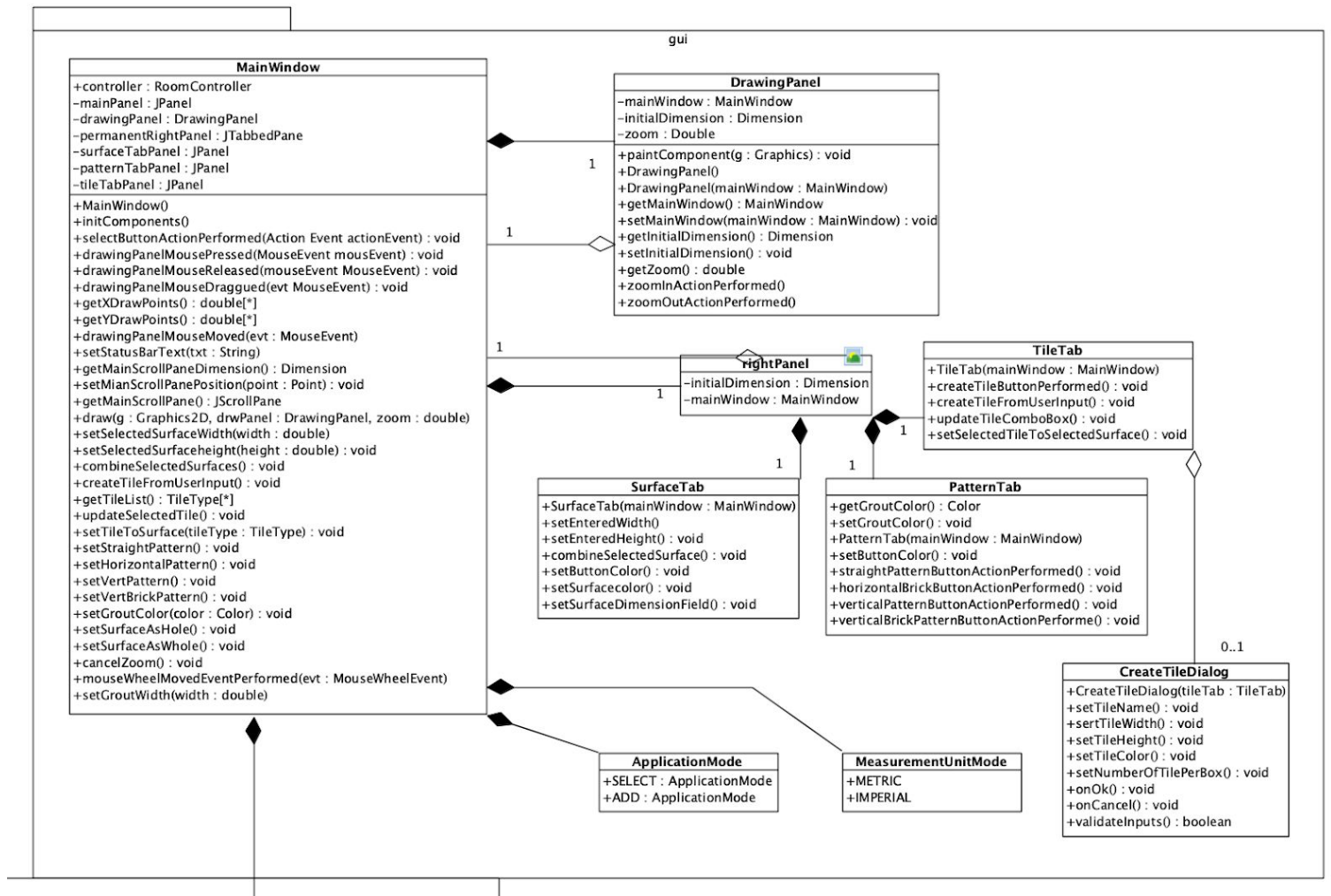
Tuiles (Tiles) : les tuiles sont choisies par l'utilisateur et elles se retrouvent sur la surface avec une certaine disposition. Les tuiles respectent des dimensions données par l'utilisateur.

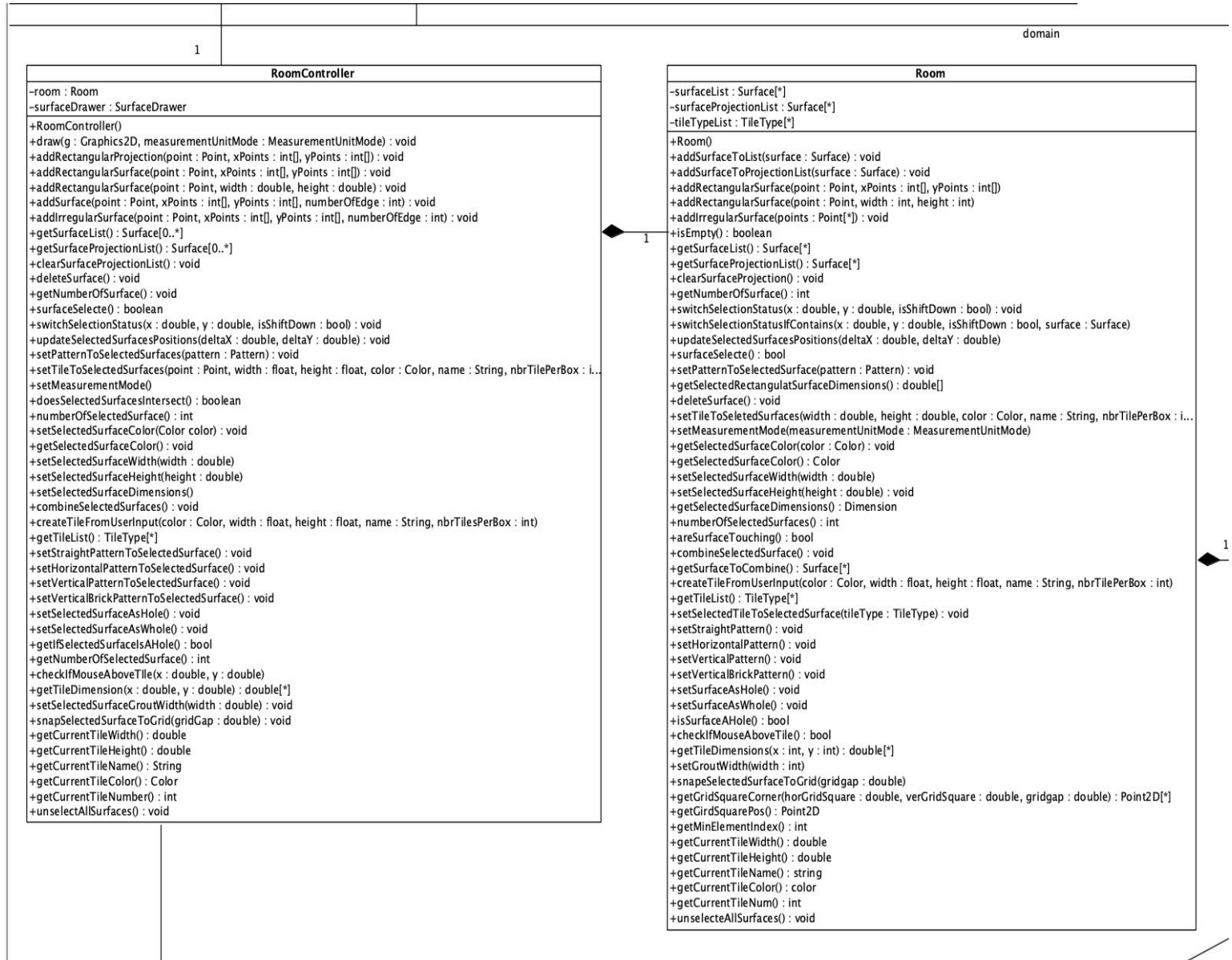
Diagramme des cas d'utilisation



Diagramme de classe de conception

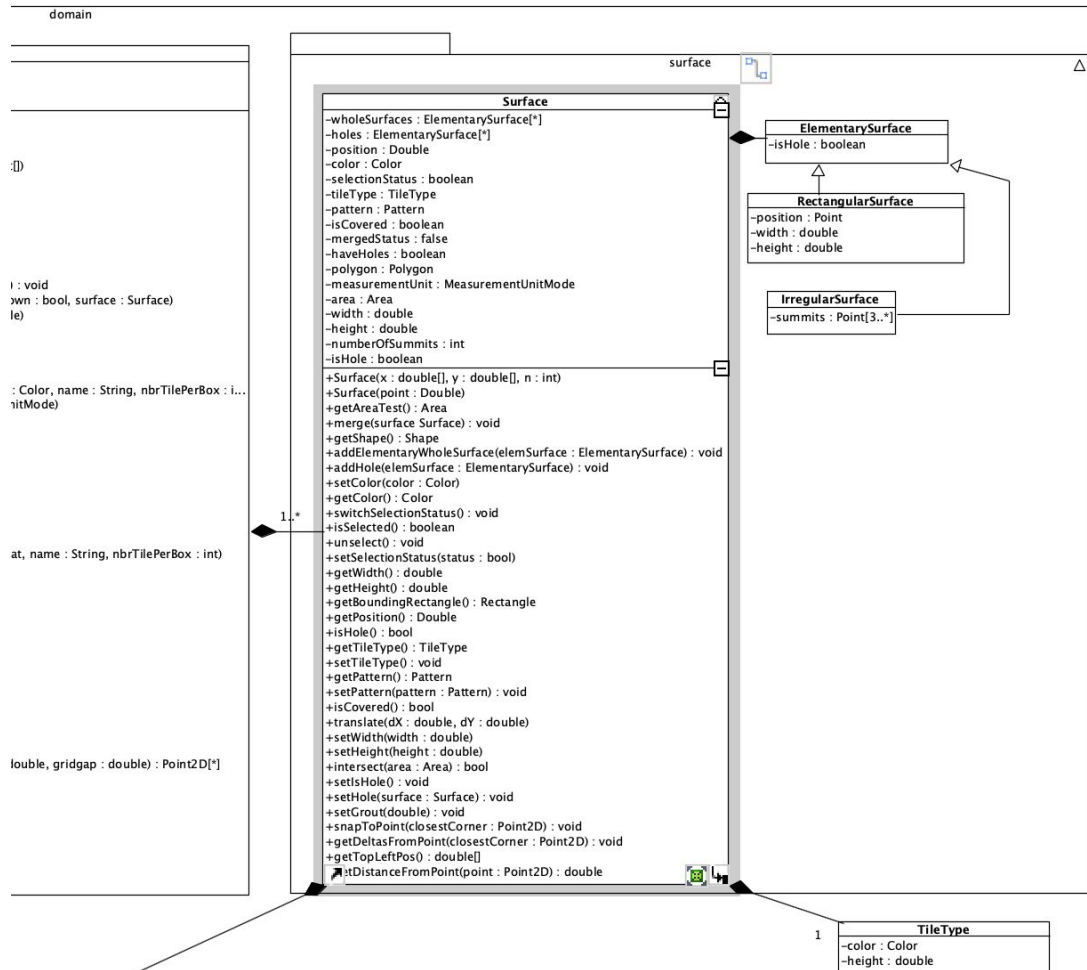






1

1



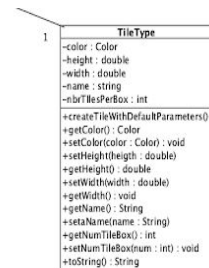
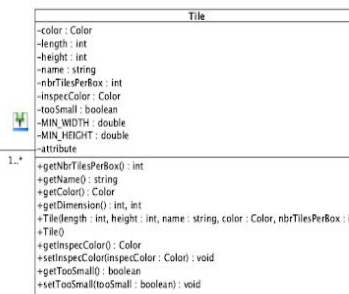
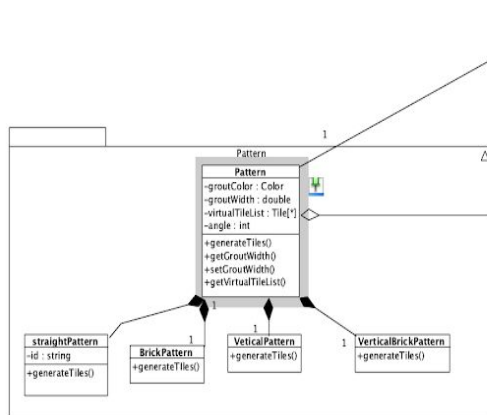
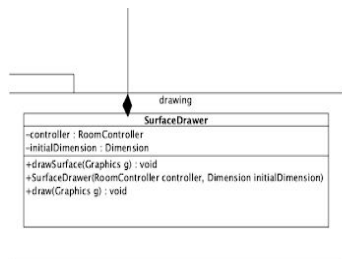
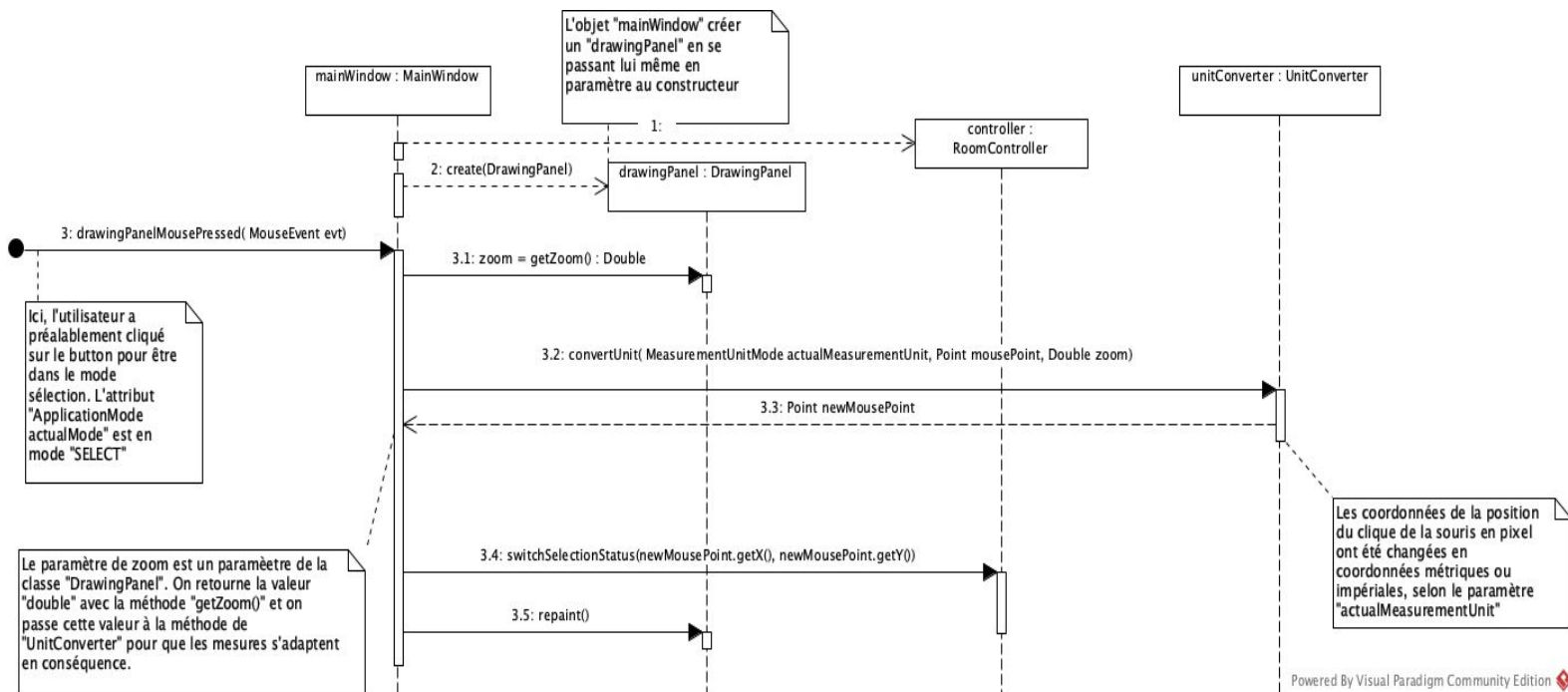


Diagramme de séquence de conception

Détermination de la surface sélectionnée

Travail réalisé pour déterminer les coordonnées x, y

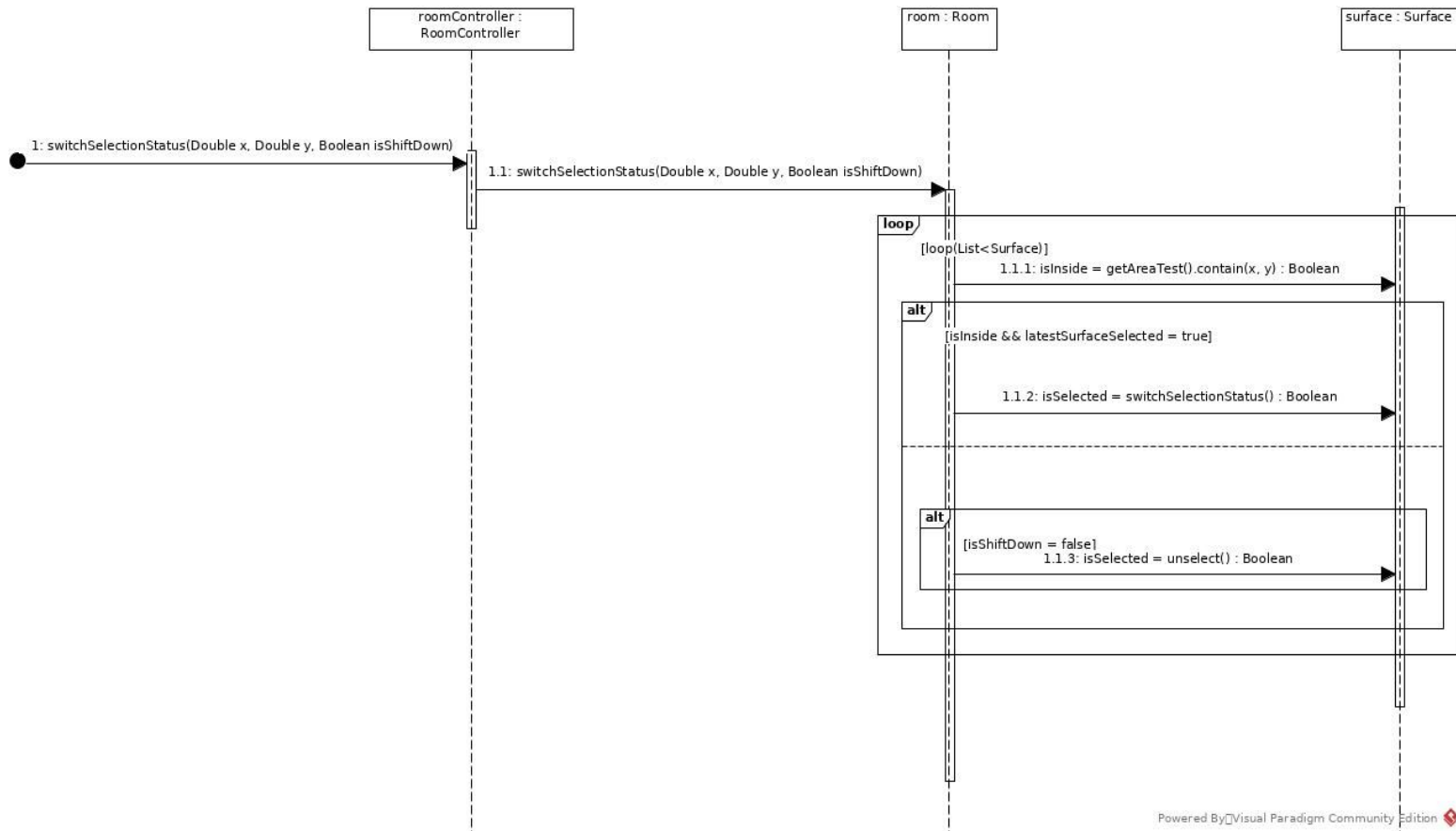


Texte explicatif

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



Powered By Visual Paradigm Community Edition

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.

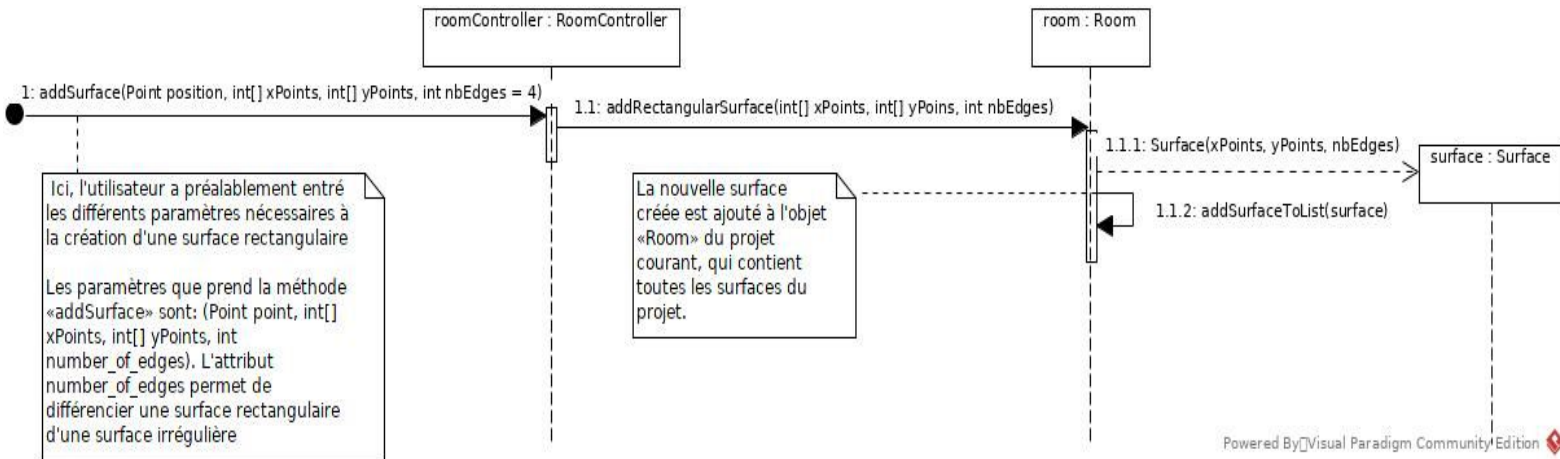
Pour chacune des surfaces dans «Room», on vérifie si les coordonnées «x, y» se trouve à l'intérieur de la surface. Pour ce faire, on fait appel à la méthode «getArea()» qui retourne un objet Area propre à la surface. Puisque la classe Area implémente l'interface

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.

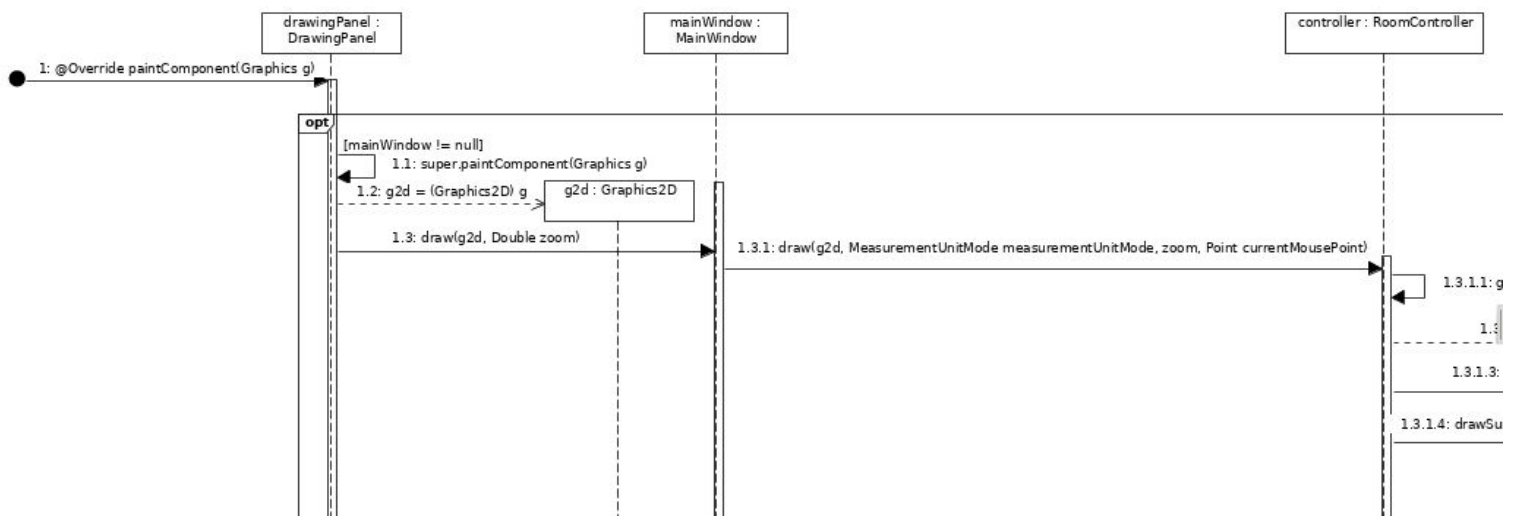
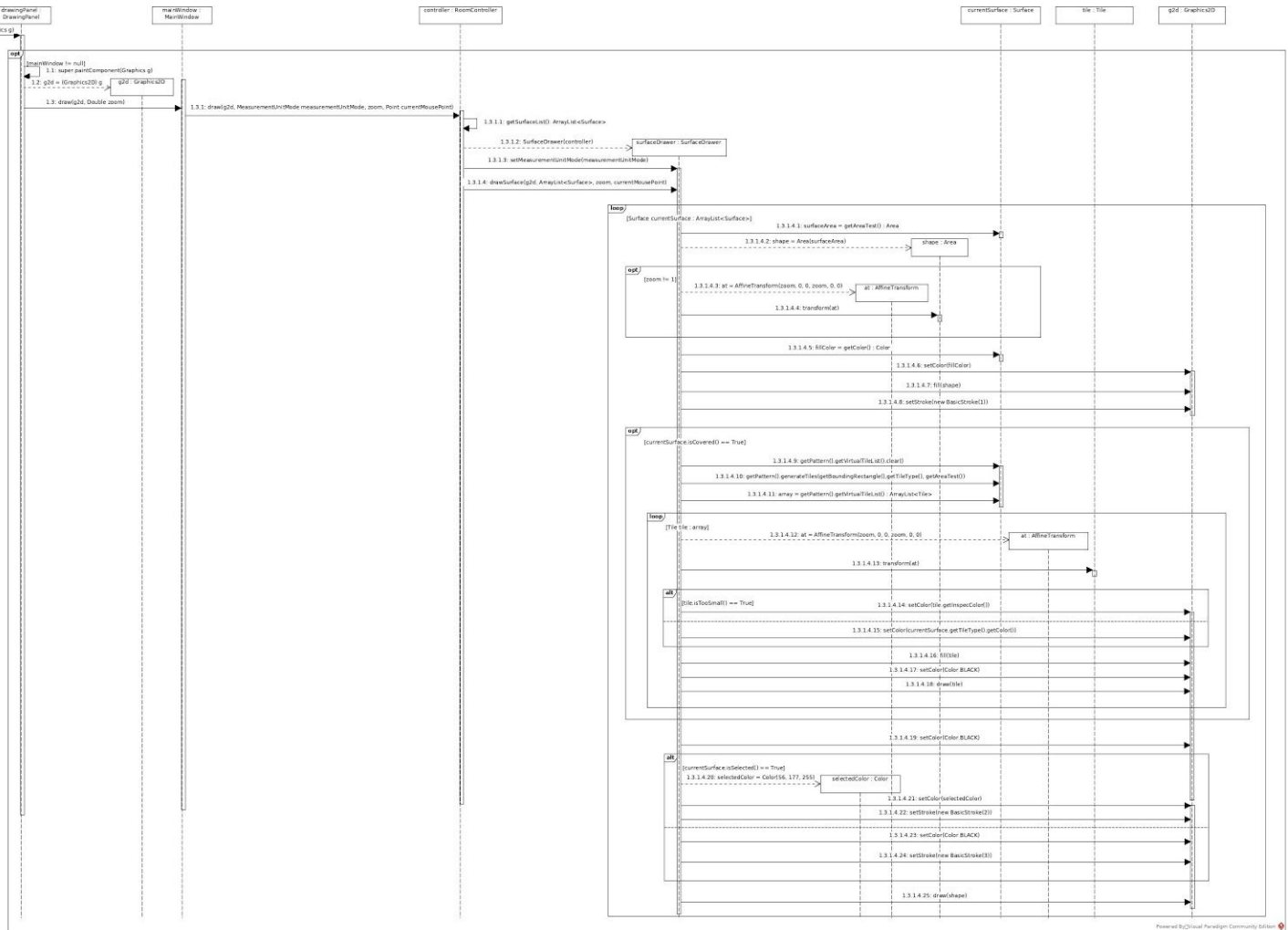
Création d'une surface rectangulaire

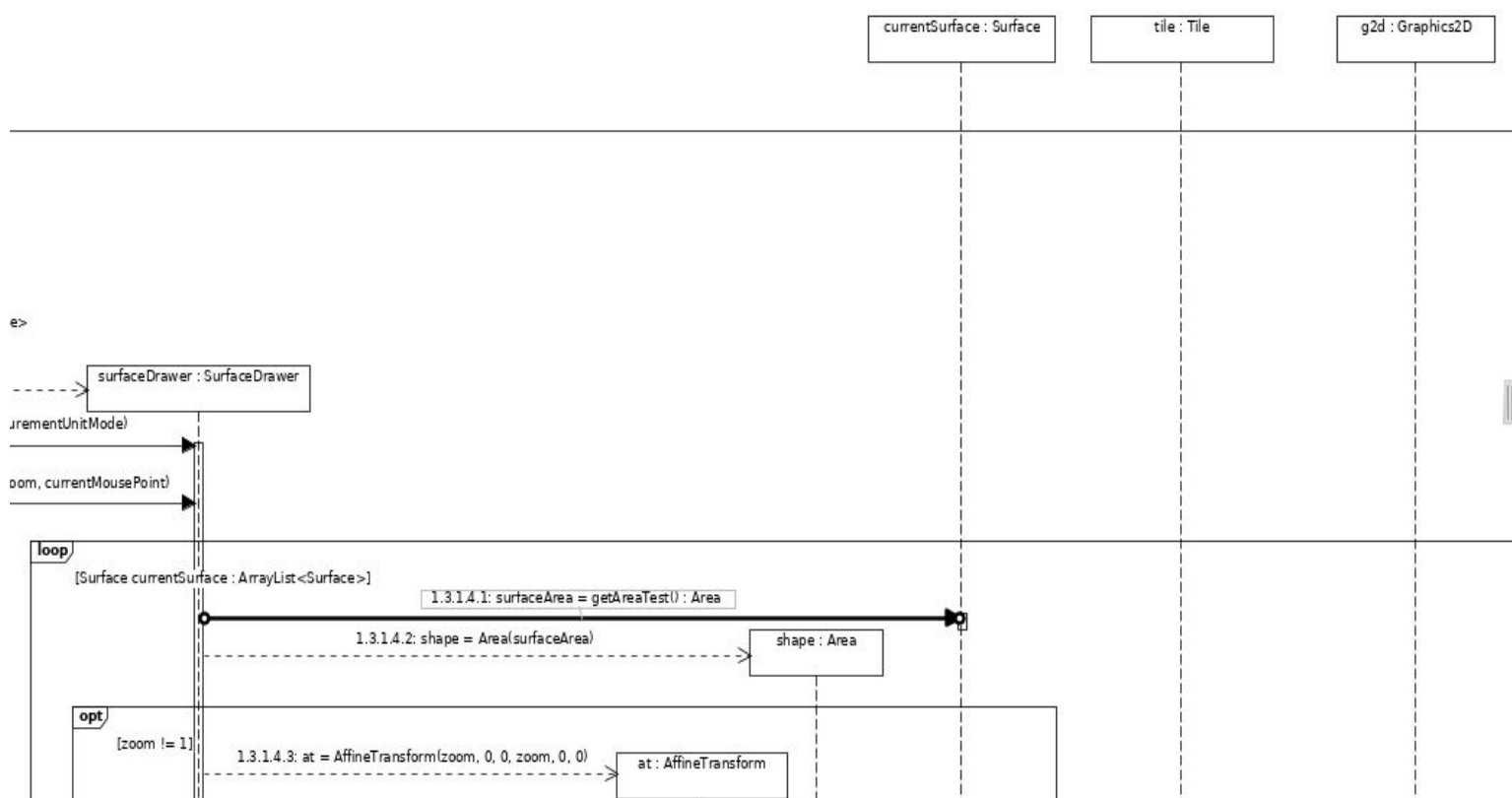
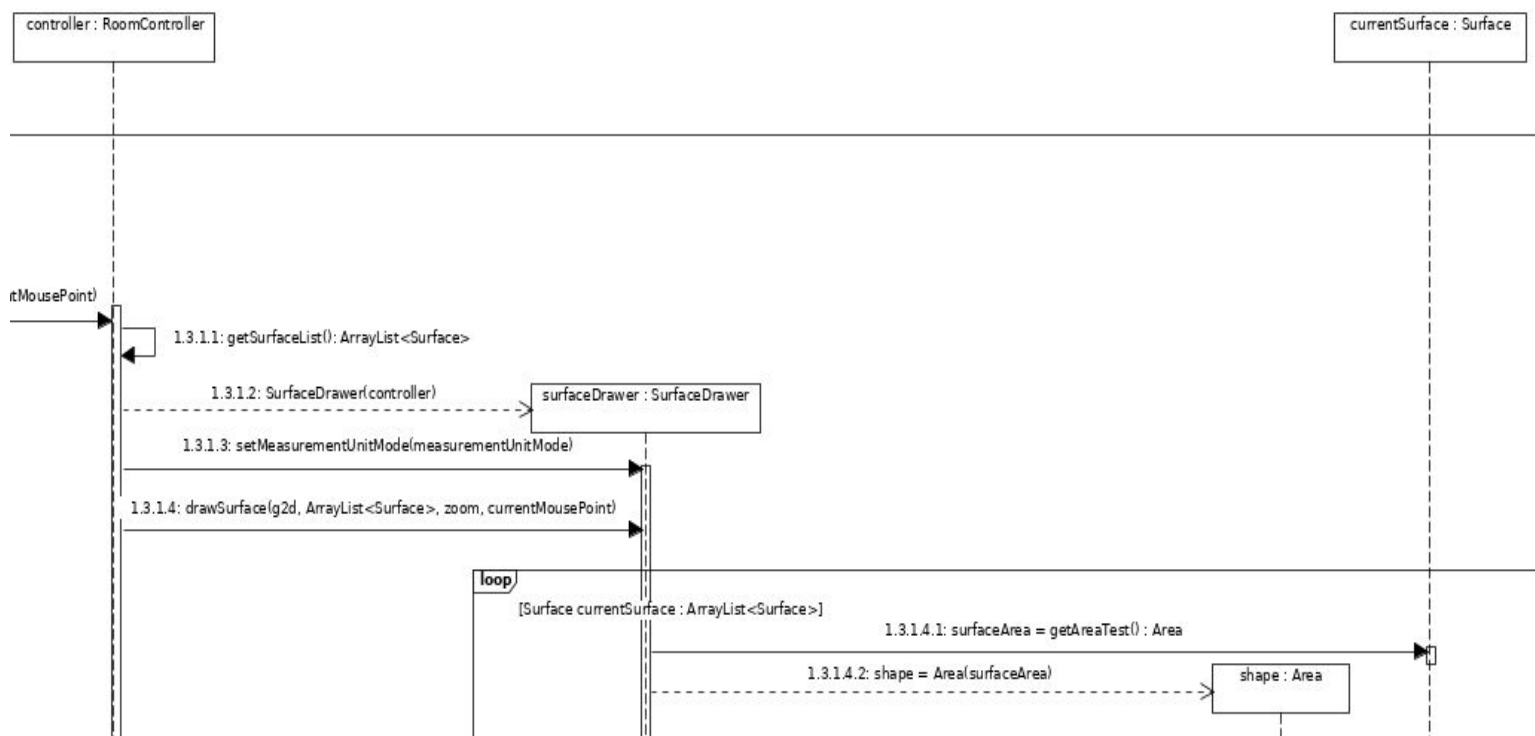


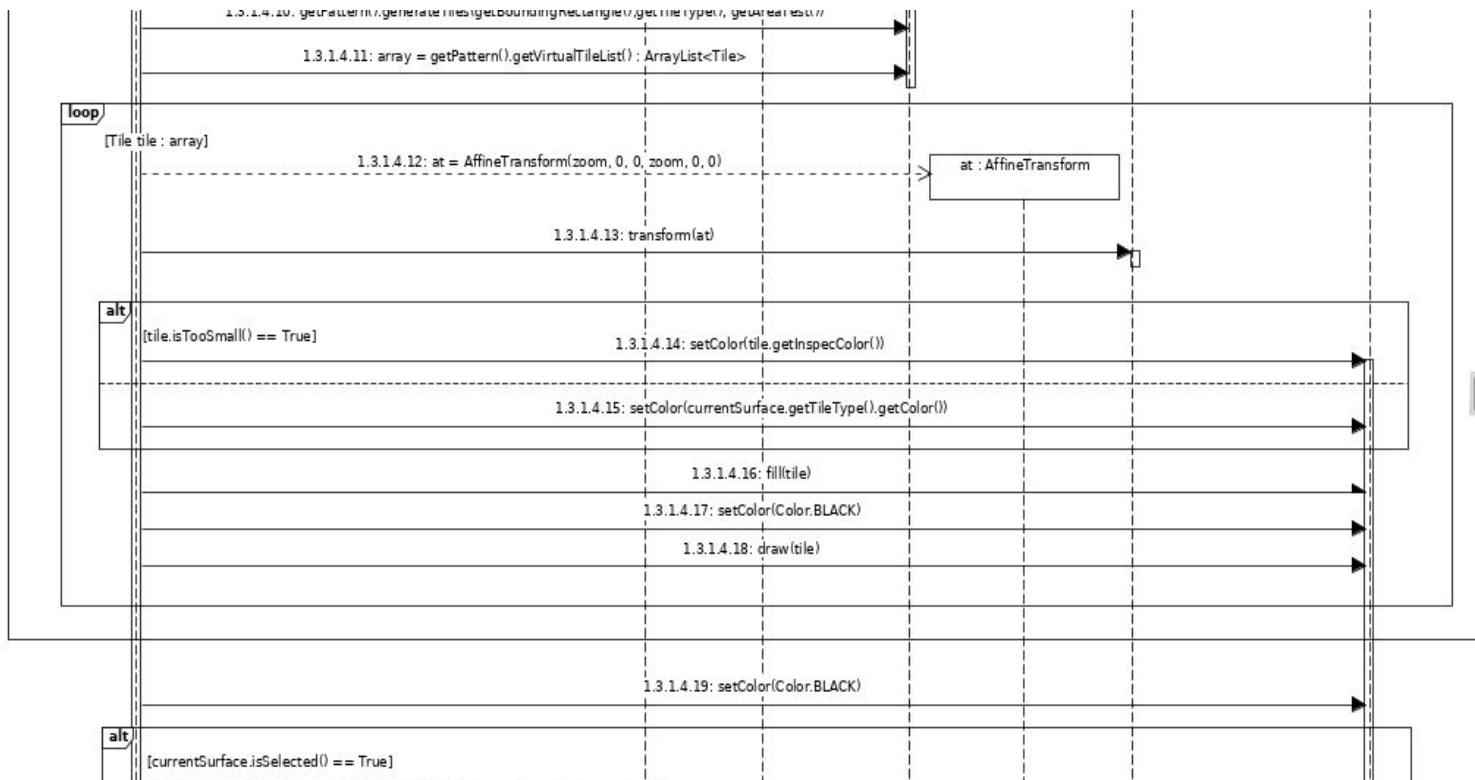
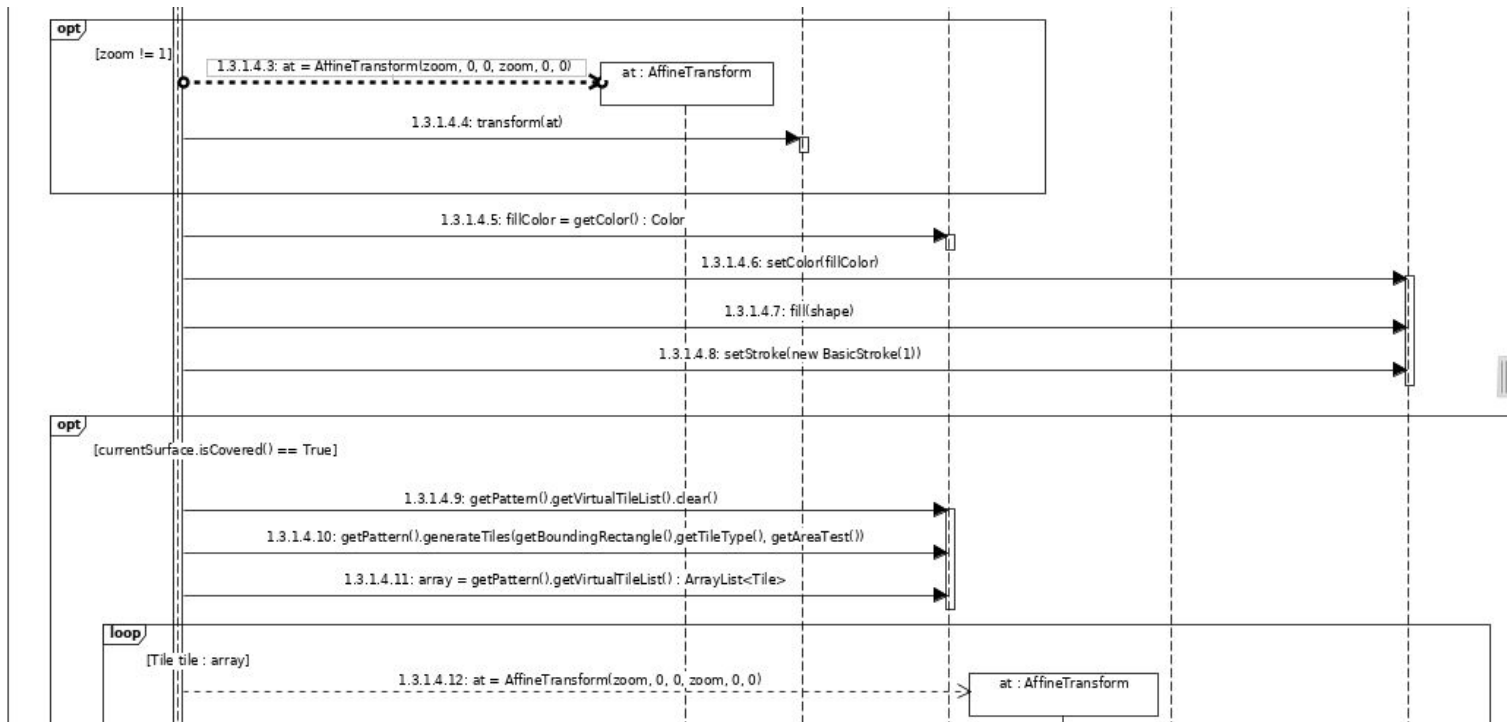
Texte explicatif

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.

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 3.