



**Projet de session : VirtuTuile
Livrable 2**

Équipe 10 :

- Matthieu BRAULT – 111 274 530
- Zakaria LAABID – 111 276 568
- Valerian HUYLEBROECK – 111 274 586
- Antoine CASSE – 111 274 625

SOMMAIRE

Table des matières

1. Diagramme de classe de conception	3
2. Architecture logique	6
3. Diagramme de séquence de conception	7
4. Pseudo code	11
5. Diagramme de Gantt	12
6. Contribution des membres de l'équipe	13
A. Énoncé de vision	15
a. Description du projet	15
b. Les principales fonctionnalités	15
c. Contexte de développement	15
B. Modèle du domaine	17
a. Diagramme de classes conceptuelles	17
b. Texte explicatif	17
C. Modèle des cas d'utilisation	18
a. Diagramme des cas d'utilisation	18
b. Texte des cas d'utilisation	19
1. Créer une surface	19
2. Supprimer une surface	21
3. Éditer les propriétés des joints	22
4. Repositionner manuellement avec la souris le motif sur la surface	23
5. Assigner un matériau à une surface	24
6. Repositionner manuellement avec la souris une surface	25
7. Modifier manuellement les propriétés de la surface	26
8. Créer un matériau	28
9. Afficher les dimensions d'une tuile	29
10. Éditer les propriétés des matériaux	30
11. Indiquer comment le motif débute	31
12. Charger le projet depuis un fichier	32
13. Enregistrer le projet dans un fichier	33
14. Spécifier le nombre de tuiles par boîtes	34
15. Afficher le nombre de tuiles, boîtes et de surface d'enduit requises	35
16. Afficher la grille magnétique	36
17. Afficher les propriétés de la surface	36
18. Coller et décoller deux surfaces	36
19. Aligner deux surfaces	39
20. Spécifier les coordonnées d'une surface	39

- Polygon

Polygon est une classe de base servant à décrire un polygone qui pourra être affiché sur la vue, elle possède les rudiments d'un polygone tels que la liste points et une couleur, il est surtout utile pour implémenter les deux classes suivantes. Grâce à cette abstraction, il est possible de passer toutes les entités à afficher en une fois. (Nous avons seulement d'afficher besoin de polygones). Se construit avec une liste de points et une valeur de l'enum PolygonType nous permettant de différencier les descendant de cette classe.

- Surface

Décrit une zone créée par l'utilisateur pour être carrelée, celle-ci, vide à la création, contiendra motif, matériau et les tuiles qui la compose quand l'utilisateur aura décidé. Il est possible de la sélectionner/désélectionner, demander le nombre de tuiles, set un pattern et motif... Elle est créée à partir d'une liste de points donnée au constructeur, qui le passera à celui de sa classe mère Polygon avec en plus le paramètre enum pour indiquer du type surface.

- Tile

Représente une tuile dans notre vue, la classe apporte uniquement une fonction isFeasable à Polygon qui nous servira à dire si une découpe ne fait pas de sens. Les tuiles ont le même constructeur que les surfaces à l'exception qu'elles précisent le type tuile.

- Material

Symbolise les différents matériaux qui sont disponibles pour l'utilisateur, l'utilisateur peut en ajouter à volonté et les assigner à une surface, pour ce faire, les tuiles hériteront des attributs du Material choisi, à savoir les points la composant (depuis les dimensions du matériau) et sa couleur.

- Pattern

La classe Pattern est utile pour générer les tuiles dans une surface, nous avons prévu de créer 4 instances statiques de cette classe qui posséderont chacune l'algorithme utile pour créer les patterns du sujet, cette classe doit prévoir que certaines dimensions ne vont pas avec certains patterns. Sa principale fonction build sera appelée une fois le matériau et motif choisi. Elle prendra alors le matériel en question et renverra la liste des tuiles créées aux bonnes positions.

- Point

Représente un point sur la vue, 2 attributs publiques x et y.

- Color

Représente une couleur avec les int correspondant au RGB de la couleur souhaitée, et un float correspondant à l'opacité de la couleur.

- SurfaceDrawer

La classe surfacesDrawer sert à afficher la surface par-dessus la vue déjà affiché par Drawing Panel.

- MainWindow

Contient la fenêtre principale, stocke une référence au contrôleur afin de pouvoir la donner à ses deux sous classes, sert d'intermédiaire de la couche de présentation vers la couche du domaine.

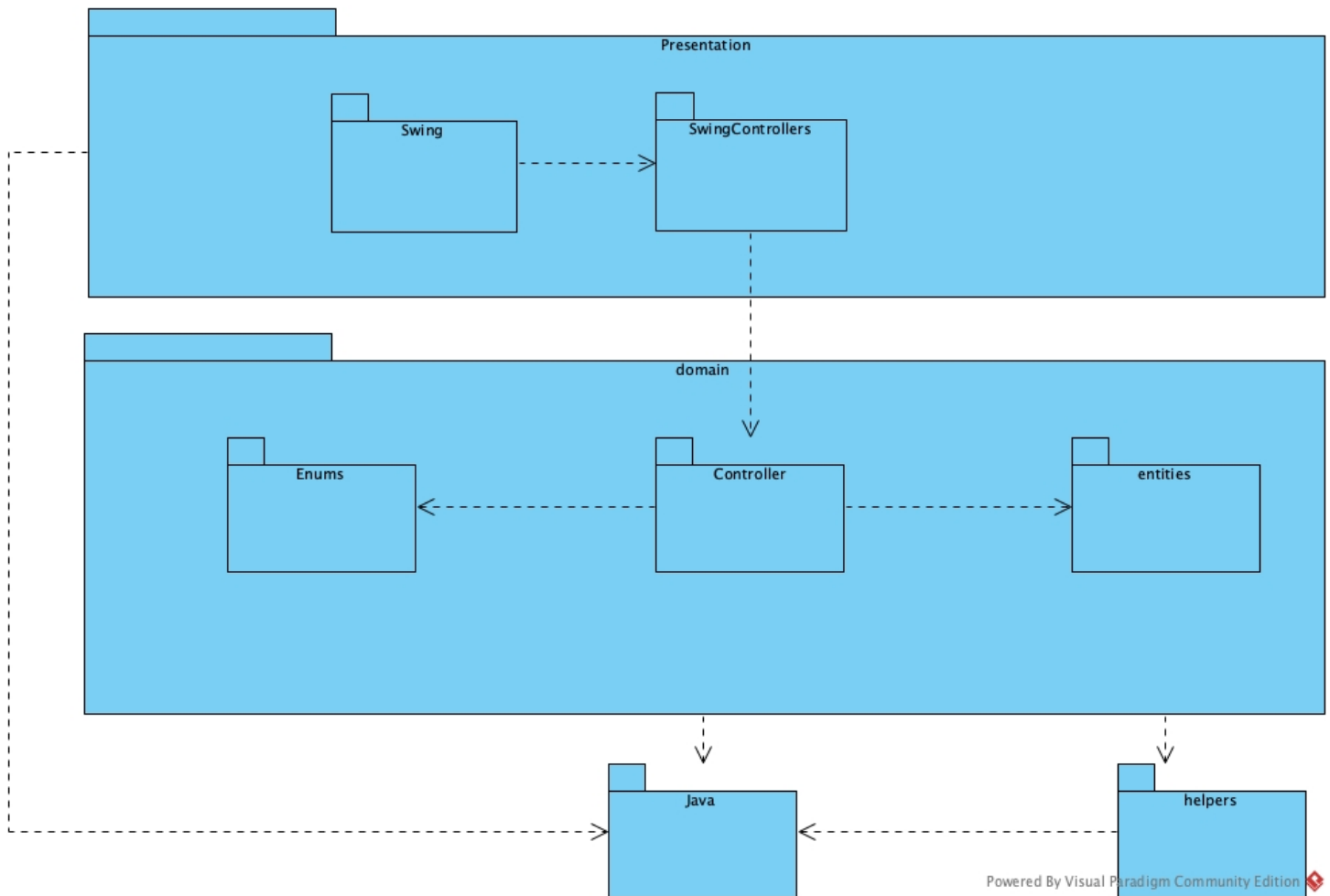
- DrawingPanel

Représente la vue sur laquelle l'utilisateur manipulera ses surfaces et tuiles, divers events dus aux entrées de souris appellent respectivement les actions correspondantes dans le contrôleur. Elles obtiennent la MainWindow à la construction et peuvent donc demander la référence au contrôleur.

- ToolBarPanel

Cette classe représente la barre d'outils de notre application, elle contient divers boutons correspondant aux différents outils. Une fois cliqué, la fonction correspondante est appelée et va passer l'information jusqu'au contrôleur de la même façon que DrawingPanel, le Controller saura alors quel outil a été sélectionné et donc comment traiter un clic sur le canvas.

2. Architecture logique



L'architecture de notre application comportera deux couches. La couche Présentation et la couche Domaine qui utiliseront les librairies Java et les helpers qui sont des fonctions utilitaires.

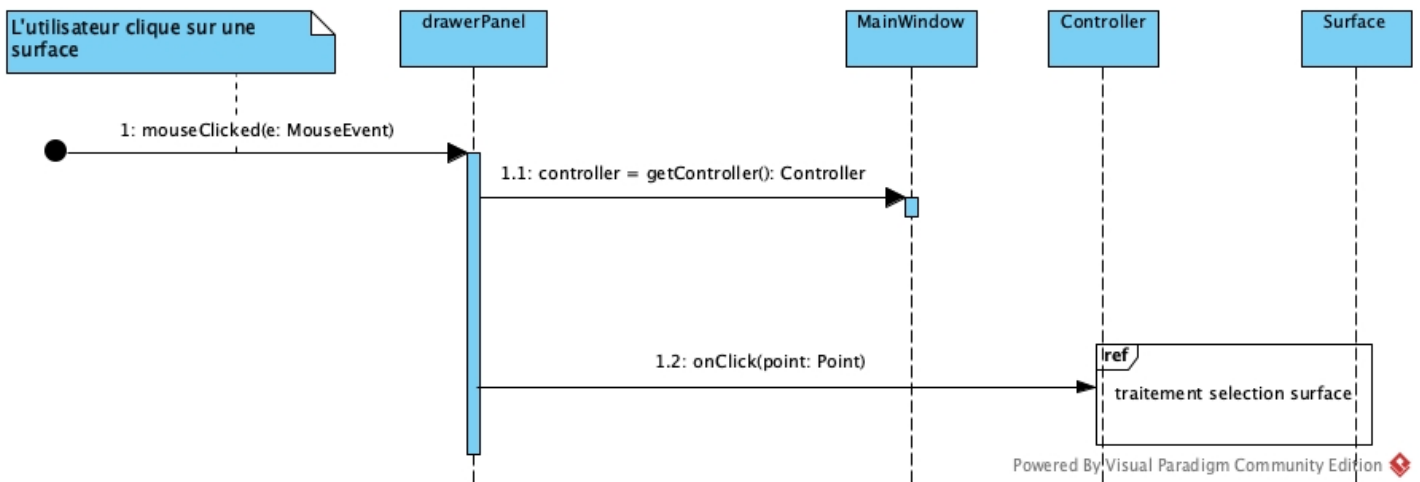
La couche du domaine comportera toute la logique de l'application. Elle appliquera la méthode du contrôleur de Larman qui recevra toutes les requêtes provenant de la couche Présentation et appellera les classes correspondantes afin d'appliquer les changements souhaités.

La couche présentation comportera toute la gestion de l'affichage avec Swing. Elle captera les entrées utilisateurs et les enverra au contrôleur de Larman, puis affichera la vue ainsi que tous les panels.

3. Diagramme de séquence de conception

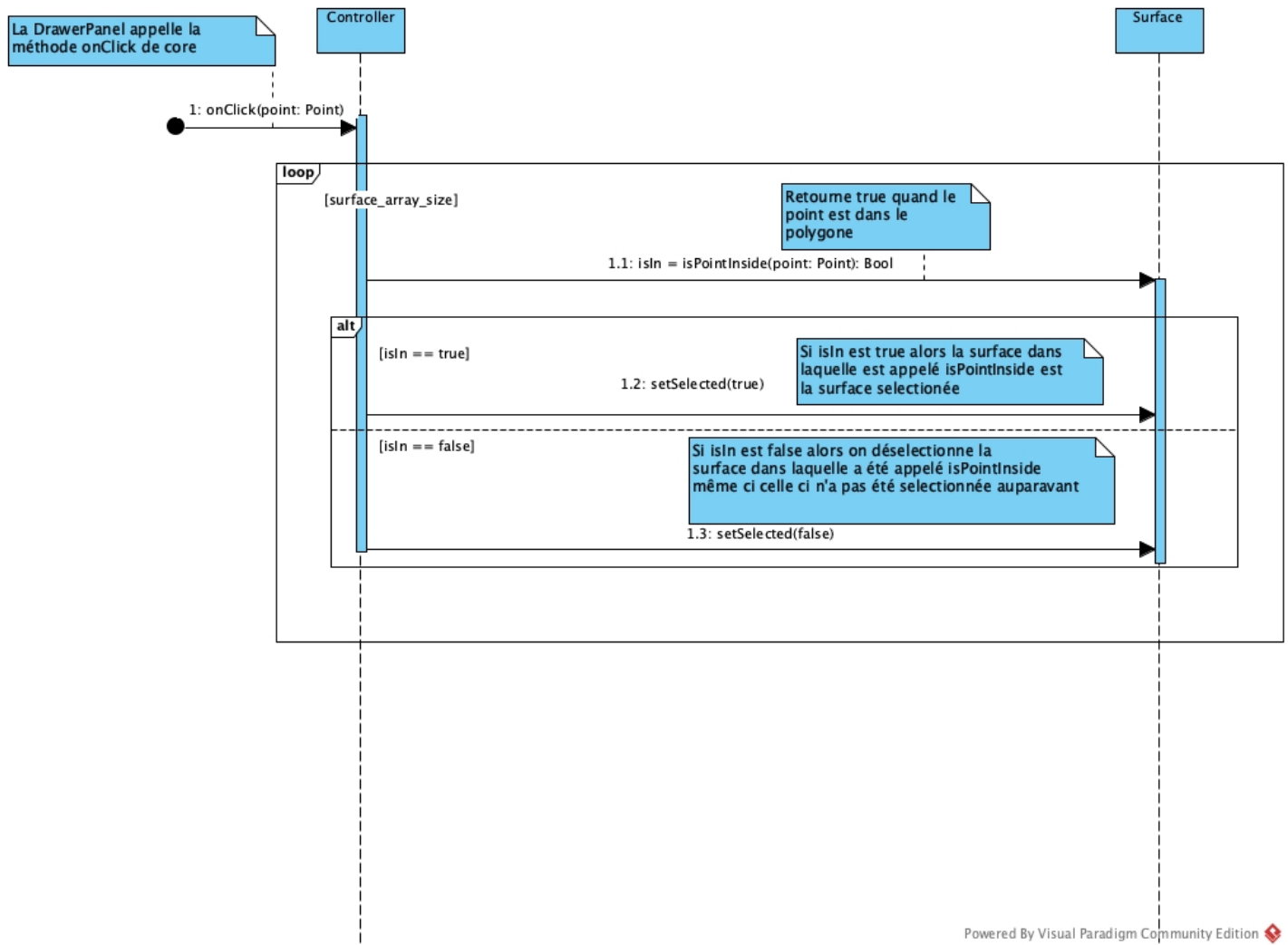
3.1. Déterminer la surface sélectionnée lors d'un clic de souris dans la vue en plan.

3.1.1. Première partie



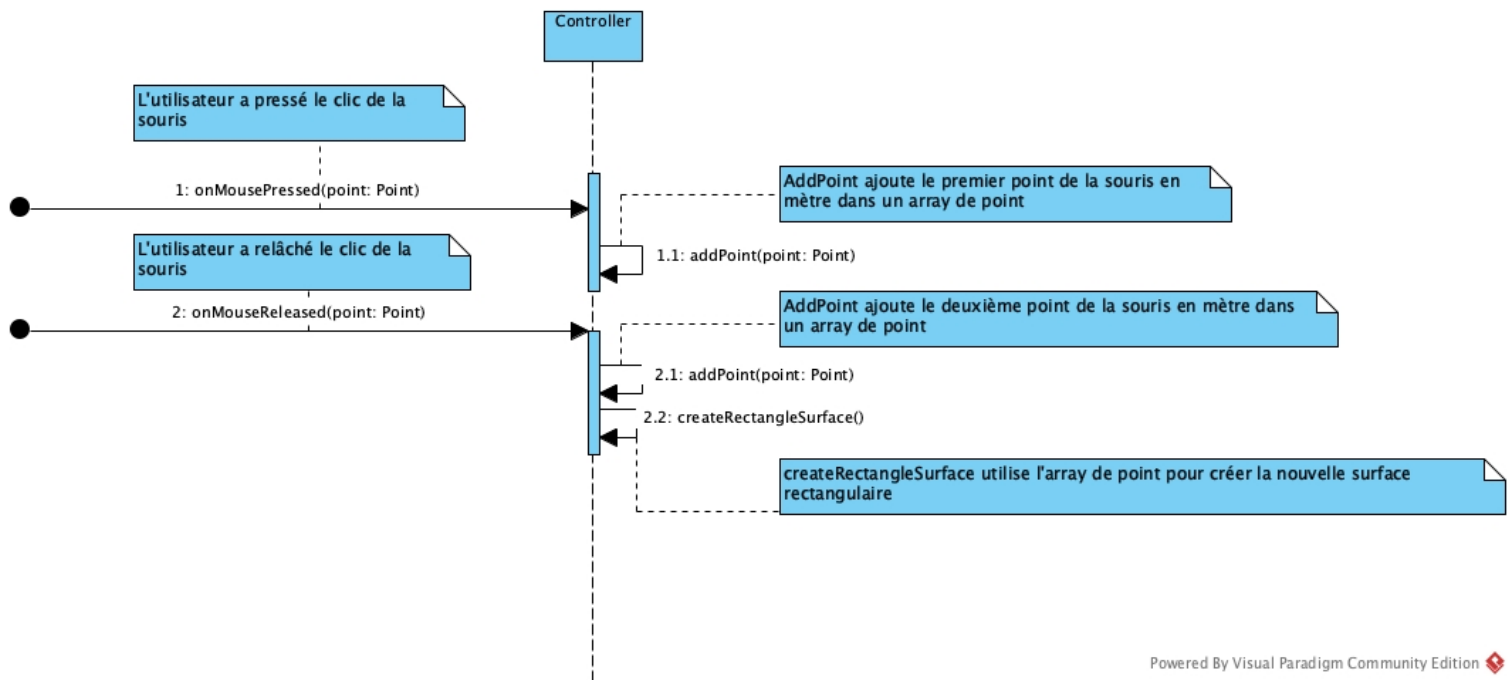
Lorsque l'utilisateur clique sur la vue, la classe **DrawerPanel** reçoit l'événement de la souris dans sa méthode *mouseClicked*. Dans cette méthode, l'événement va être converti en une classe **Point** contenant les positions x et y du clic. Dans cette méthode, **DrawerPanel** récupère le **Controller** grâce à la **MainWindow** et appelle la fonction *onClick* de ce dernier en lui donnant l'objet **Point** créé auparavant.

3.1.2. Traitement sélection surface



Le **Controller** reçoit l'objet `Point` envoyé par **DrawerPanel**. Il boucle sur le tableau contenant la liste des surfaces présentes et appelle leur méthode `isPointInside` en donnant comme paramètre le **Point**. Si le click a été fait à l'intérieur de la **Surface**, alors il set l'attribut `selected` à `true`. Sinon (si l'utilisateur n'a pas pressé la touche contrôle), il set `selected` à `false` ce qui permet de sélectionner la/les surface(s) sélectionné précédemment.

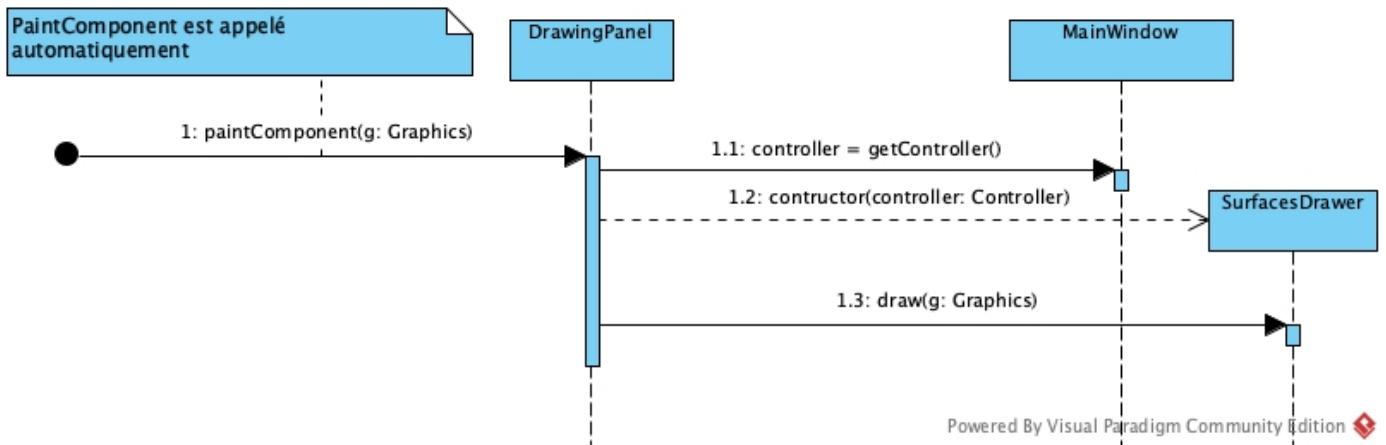
3.2. Création d'une nouvelle surface rectangulaire



Powered By Visual Paradigm Community Edition

Pour créer une nouvelle surface rectangulaire l'utilisateur va presser le clic de sa souris là où il veut que débute sa nouvelle surface. La pression du clic va appeler la méthode `onMousePressed` de la classe **Controller** et fournir les informations du clic en paramètre. La méthode `onMousePressed` va rajouter la position du clic dans une liste de point contenu dans la classe **Controller**. Lorsque la pression du clic va être relâchée une autre méthode de la classe **Controller** va être déclenchée. Cette méthode va ajouter la position de la souris au relâchement du clic à la liste de point puis grâce à cette liste de point nous allons pouvoir créer la surface rectangulaire. C'est dans la méthode `createRectangleSurface` de la classe **Controller** que les traitements pour créer la surface vont être fait.

3.3. Affichage de la vue en plan



La classe **DrawingPanel** reçoit l'événement **paintComponent** avec un objet Swing **Graphics**. Elle récupérera le **Controller** grâce à la **MainWindow** et initialisera une classe **SurfacesDrawer** afin d'y appeler sa méthode *draw* pour qu'elle affiche la vue en plan.

4. Pseudo code

La fonction s'appellera `isPointInside`, elle retournera un boolean et prendra 2 paramètres : un point et un tableau qui contient tous les points d'un polygone (dans l'ordre de traçage des segments).

Elle appartiendra à la classe `surface`.

Fonction `isPointInside (Point polygone[], Point p)`

`nbSegment <- 0`

Pour `i` de 0 jusqu'à `polygone.length` avec un pas de 1

Si `isOnLine(polygone[i], polygone[i + 1], p)` **est égal à** `true`

`nbSegment <- nbSegment + 1`

Si `isOnLine(polygone[0], polygone[polygone.length - 1], p)` **est égal à** `true`

`nbSegment <- nbSegment + 1`

Si `nbSegment` est impair

Retourne `true`

Sinon

Retourne `false`

// La fonction `isOnLine` vérifie si une demie droite de vecteur (`p3.x + 1`, `p3.y`) partant de `p3`, entre en intersection avec le segment [`p1`, `p2`]

Fonction `isOnLine(p1, p2, p3)`

Si `p3.x < max(p1.x, p2.x)` **et que** `p3.y < max(p1.y, p2.y)` **et que** `p3.y > min(p1.y, p2.y)`

Retourne `true`

Sinon si `p1.x >= p3.x` **et que** `p1.y == p3.y`

Sinon

Retourne `false`

5. Diagramme de Gantt

	Diagramme de Gantt										
Date de début d'itération	01/10/2019	08/10/2019	15/10/2019	22/10/2019	29/10/2019	05/11/2019	12/11/2019	19/11/2019	26/11/2019	03/12/2019	10/12/2019
Numéro d'itération	2	3	4	5	6	7	8	9	10	11	12
Squelette de l'application											
Créer une surface											
Sélectionner une surface											
Supprimer une surface											
Repositionner manuellement une surface											
Afficher les propriétés de la surface											
Assigner un matériau à une surface											
Modifier les propriétés de la surface											
Créer un matériau											
Indiquer comment le motif débute											
Zoomer / Dezoomer											
Sélectionner un motif pour une surface											
Editer les propriétés des matériaux											
Afficher les propriétés d'un matériau											
Repositionner manuellement le motif											
Editer les propriétés des joints											
Afficher les propriétés des joints											
Afficher la grille magnétique											
Spécifier le nombre de tuiles par boîte											
Afficher le nombre de tuiles et le nombre de boîtes requises											
Spécifier les coordonnées d'une surface par rapport au coin supérieur gauche d'une autre surface											
Combiner 2 surfaces (incluant un trou)											
est inférieure à N unités											
Calculer instantanément pour toutes les surfaces utilisant ce même matériau											
Aligner deux surfaces haut/bas/droite/gauche											
Undo / Redo											
Centrer entre elle 2 surfaces verticalement ou horizontalement											
Enregistrer un projet dans un fichier											
Charger un fichier											
Coller / décoller deux surfaces horizontalement et verticalement											

6. Contribution des membres de l'équipe

- 1 – Diagramme de classe de conception : **Antoine, Valerian** et **Matthieu**
- 2 – Architecture logique : **Zakaria**
- 3 – Diagramme de séquence : **Antoine** et **Valerian**
- 4 – Pseudo code : **Valerian** et **Matthieu**
- 5 – Diagramme de Gantt : **Matthieu**

Le code a été réalisé par l'ensemble de l'équipe, avec le lead de **Zakaria** qui avait déjà une bonne expérience dans le Java.

Une fois chaque livrable effectué, l'ensemble du document a été revu par l'équipe afin d'y apporter quelques modifications.

ANNEXE

A. Énoncé de vision

a. Description du projet

L'objectif de ce projet est de réaliser une application du nom de **VirtuTuile**, qui servira à planifier la pose de revêtements de plancher et de revêtements muraux.

L'application permet de modéliser, dans un même projet, une ou plusieurs surfaces à couvrir. Une surface élémentaire prend la forme d'un rectangle ou d'un polygone irrégulier (défini par ses sommets). Deux surfaces peuvent être sélectionnées et combinées pour donner une nouvelle surface. Une fois deux surfaces combinées elles peuvent être manipulées avec la souris comme s'il s'agissait d'une seule surface. Une surface issue de la combinaison de deux surfaces peut à son tour être combinée avec une autre surface, etc.

Chaque surface a une propriété qui définit si elle est réellement à couvrir ou s'il s'agit plutôt d'un « trou », par exemple une fenêtre. Les trous peuvent être combinés ensemble ou avec d'autres surfaces qui ne sont pas des trous.

b. Les principales fonctionnalités

- Créer une surface, dont on pourra sélectionner le matériau et la dimension.
- Pouvoir coller ou combiner 2 surfaces via la souris ou un panneau d'édition.
- Pouvoir undo ou redo n'importe quelle action.
- Pouvoir enregistrer ou charger un projet dans un fichier.
- Pouvoir repositionner manuellement, avec la souris, le motif sur la surface

c. Contexte de développement

Le projet *VirtuTuile* sera réalisé entre le 10 Septembre 2019 et le 17 Décembre 2019. Il est composé de 4 livrables qui seront à rendre un par un le 24 Septembre, 15 Octobre, 26

Novembre et 17 Décembre. Les livrables seront décomposés en plusieurs itérations d'une semaine.

Le premier livrable (celui pour lequel nous rendons ce document) sera constitué des artefacts du Business Modeling et Requirements. Ils seront dans l'état d'une fin de première itération de la phase d'élaboration ;

Le projet sera réalisé en groupe de 4 personnes et développé sur un dépôt Gitlab sur lequel il ne sera pas autorisé de faire des branches.

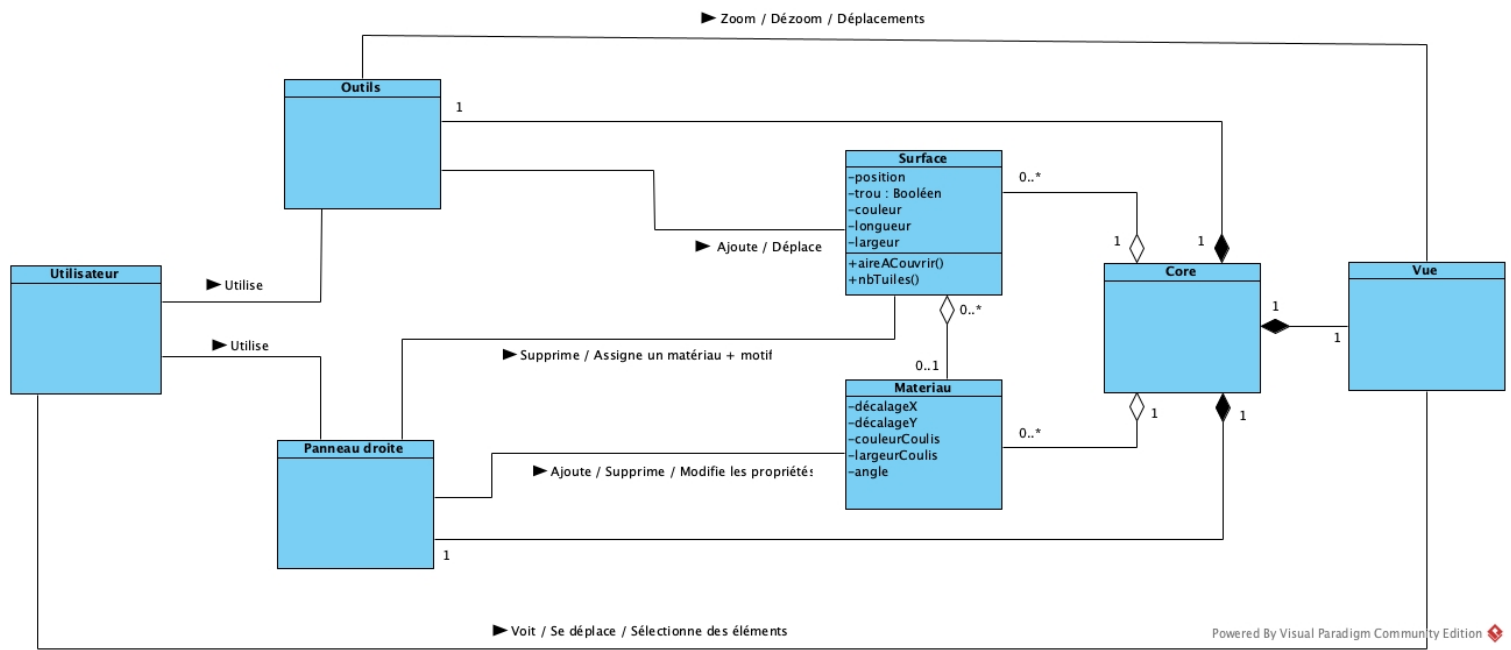
Il sera réalisé avec le langage Java et l'environnement de développement IntelliJ et l'interface graphique Swing.

Seulement 2 membres de l'équipe ont déjà eu une expérience avec le langage Java. Mais les 2 autres membres s'engagent à faire le travail nécessaire afin de rattraper leur retard et pouvoir fournir la même charge de travail que les 2 membres expérimentés. Le projet est donc réalisable dans son ensemble dans le temps imparti.

Une réunion hebdomadaire sera mise en place le Mercredi à 14h afin de faire le point sur le travail fourni par chaque membre et pouvoir aider si quelqu'un bloque sur une problématique bien précise.

B. Modèle du domaine

a. Diagramme de classes conceptuelles



b. Texte explicatif

Utilisateur : L'utilisateur est la personne qui utilise notre logiciel. Il peut grâce aux outils mis en place effectué des actions. il peut sans utiliser les outils supprimer, créer des matériaux qu'il pourra par la suite assigner à une surface. L'utilisateur peut se déplacer et sélectionner des éléments dans la vue.

Outils : Les outils contiennent les différentes fonctionnalités permettant d'effectuer des actions nécessaires pour l'utilisateur. Il regroupe donc les différentes actions suivantes : déplacer, supprimer et ajouter des surfaces.

Core : Le cœur du logiciel, le core fait la liaison entre les différentes parties du projet, les actions de l'utilisateur s'effectuent dessus et ici sont stockées les informations du projet en cours comme les surfaces créées. Ces informations sont ensuite transmises au module de vue.

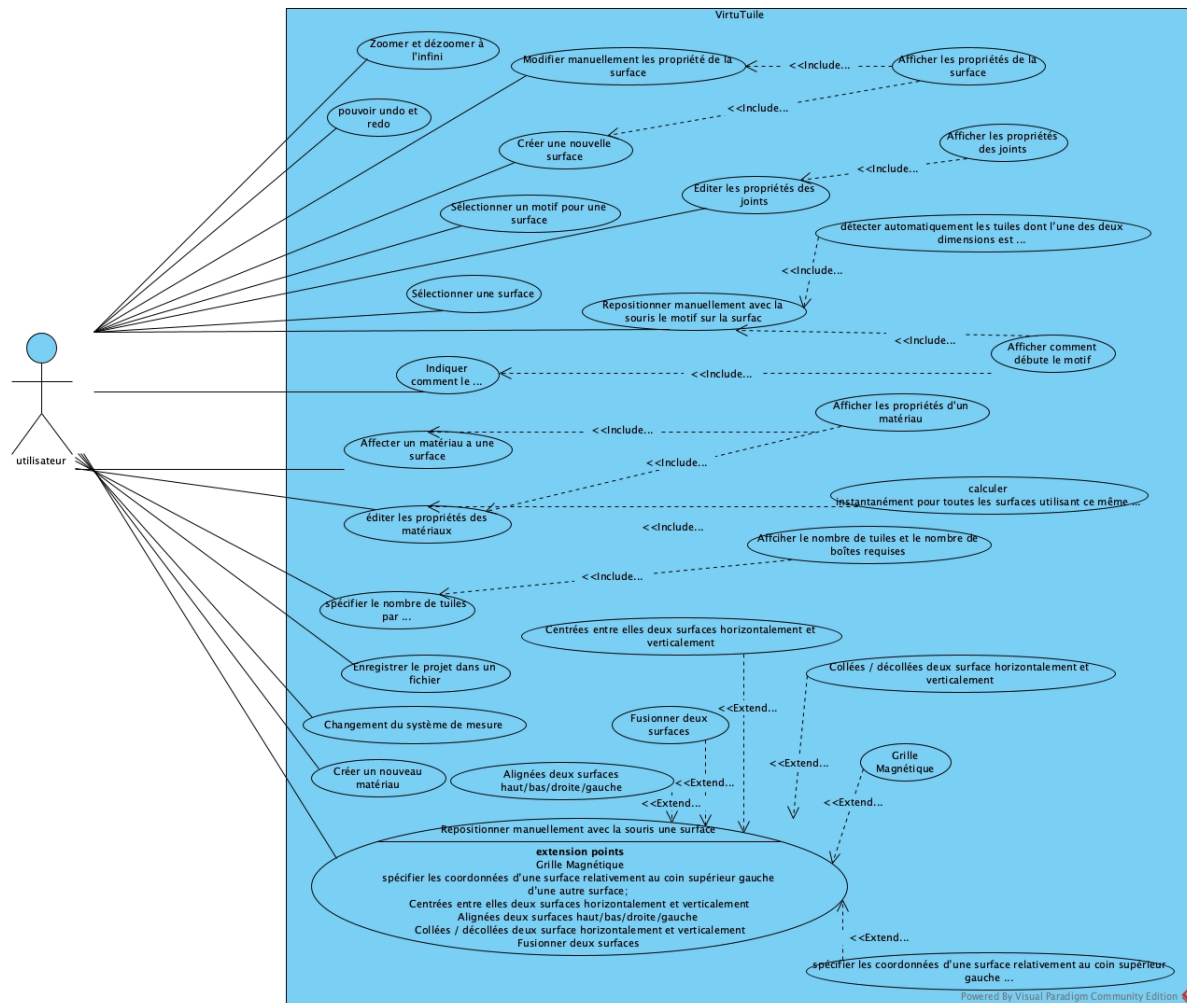
Vue : La vue est l'espace de travail dans lequel on pourra apercevoir le résultat des différentes actions de l'utilisateur sur le logiciel. Notamment l'affichage des surfaces. La vue est cliquable et permet d'effectuer des actions selon l'outil choisi.

Surface : La surface représente la zone d'une pièce à recouvrir par un motif et un matériau. Grâce à la surface on va pouvoir récupérer les informations liées au recouvrement tel que le nombre de tuile nécessaires.

Matériau : Le matériau est la tuile avec des propriétés tel que sa taille et sa couleur. Le matériau peut être assigné à une surface.

C. Modèle des cas d'utilisation

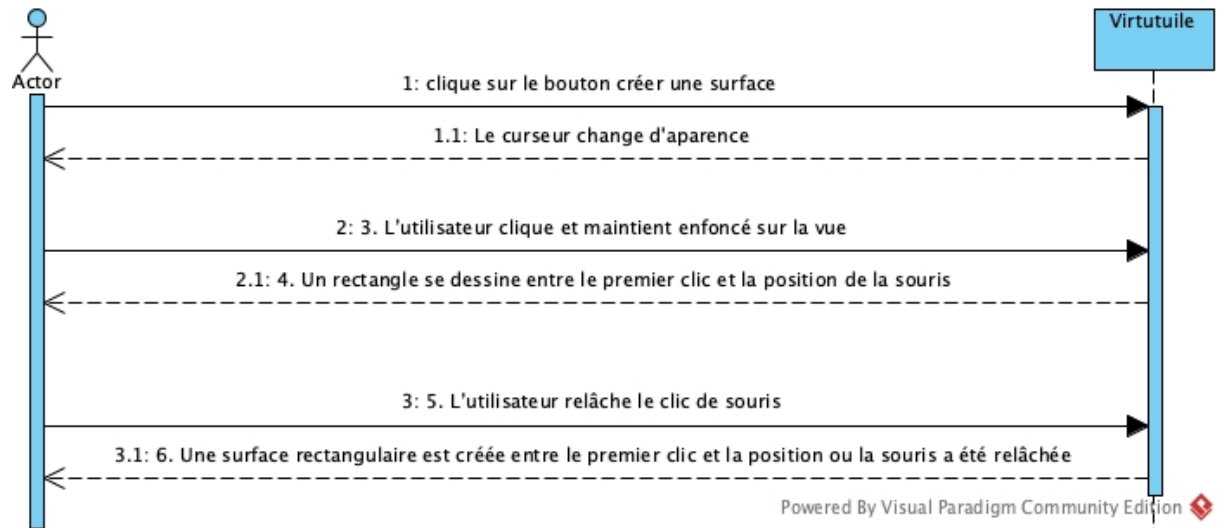
a. Diagramme des cas d'utilisation



b. Texte des cas d'utilisation

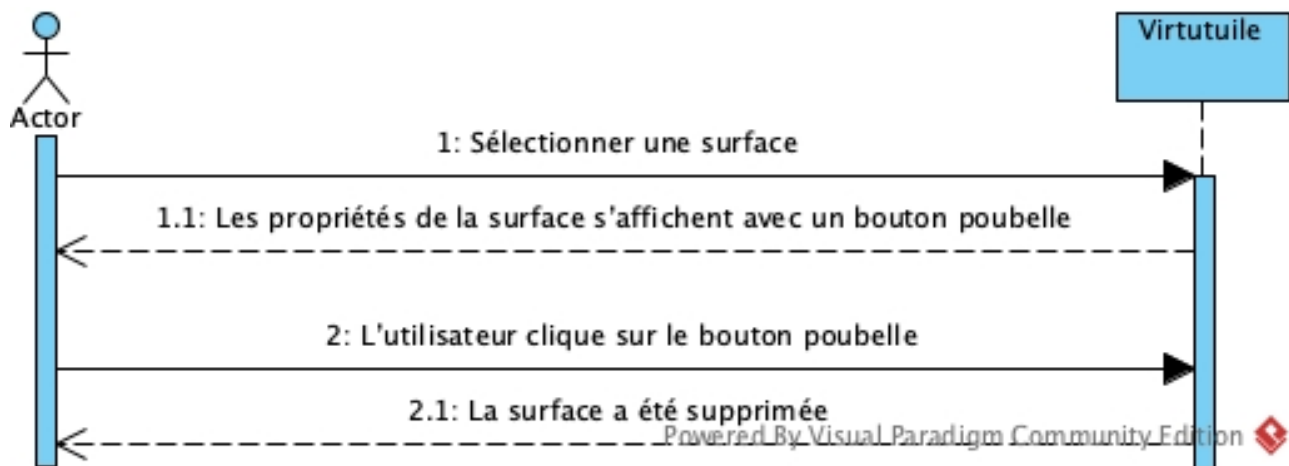
1. Créer une surface

Cas d'utilisation:	Créer une surface
Système:	VirtuTuile
Acteur(s)	Utilisateur
Partie prenante et intérêts:	Utilisateur: Il veut ajouter une surface pour lui assigner un motif
Précondition(s):	Aucune
Garanties en cas de succès :	La surface est représentée dans la vue.
Scénario principal:	<p>1.L'utilisateur clique sur le bouton créer une surface rectangulaire.</p> <p>2.Le curseur change d'apparence.</p> <p>3.L'utilisateur clique et maintient enfoncé sur la vue.</p> <p>4.Un rectangle se dessine entre le premier clic et la position de la souris.</p> <p>5.L'utilisateur relâche le clic de la souris.</p> <p>6.Une surface rectangulaire est créée entre le premier clic et la position où la souris a été relâchée.</p>
Scénario(s) alternatif(s):	<p>1.L'utilisateur clique sur le bouton créer une surface polygonale</p> <p>2.Le curseur change d'apparence.</p> <p>3.L'utilisateur clique à plusieurs endroits de la vue</p> <p>4.A chaque clic à partir du deuxième, un segment est dessiné reliant le nouveau sommet au précédent</p> <p>5. L'utilisateur clique sur le premier point.</p> <p>6. Un polygone est créé à partir des sommets donnés.</p>



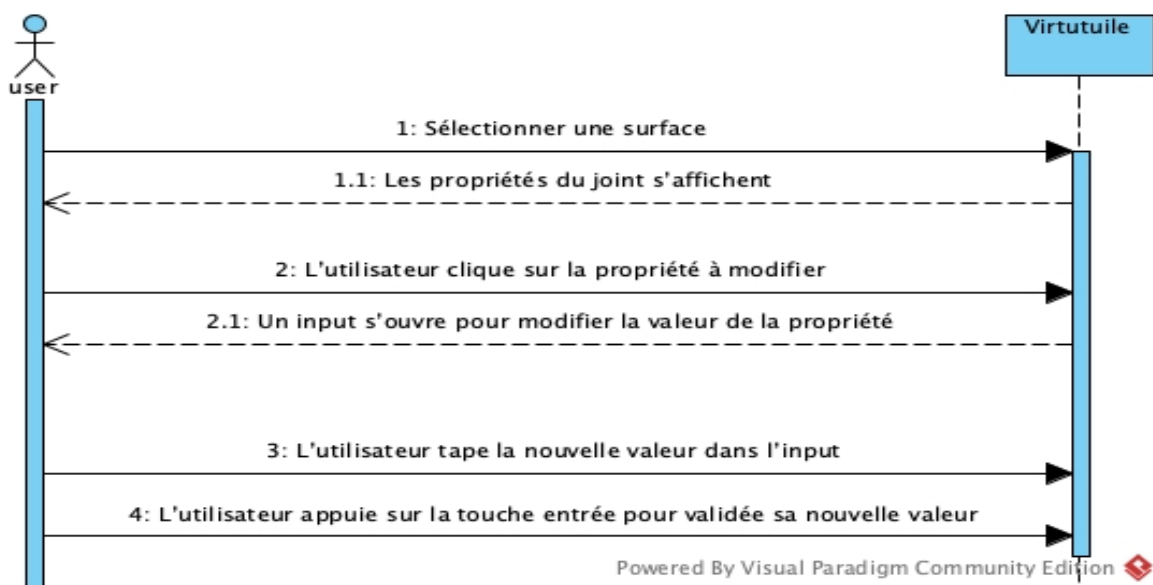
2. Supprimer une surface

Cas d'utilisation:	Supprimer une surface
Système:	VirtuTuile
Acteur(s)	Utilisateur
Partie prenante et intérêts:	Utilisateur: L'utilisateur veut supprimer une surface qu'il ne veut plus
Précondition(s):	Avoir créer une surface
Garanties en cas de succès :	La surface n'est plus présente dans la vue
Scénario principal:	<p>1.Sélectionner une surface.</p> <p>2.Les propriétés de la surface s'affichent avec un bouton poubelle.</p> <p>3.L'utilisateur clique sur le bouton poubelle.</p> <p>4.La surface précédemment sélectionnée a été supprimée.</p>
Scénario(s) alternatif(s):	<p>1.Sélectionner une surface.</p> <p>2.Les propriétés de la surface s'affichent avec un bouton poubelle.</p> <p>3.L'utilisateur appuie sur la touche « suppr » du clavier</p>



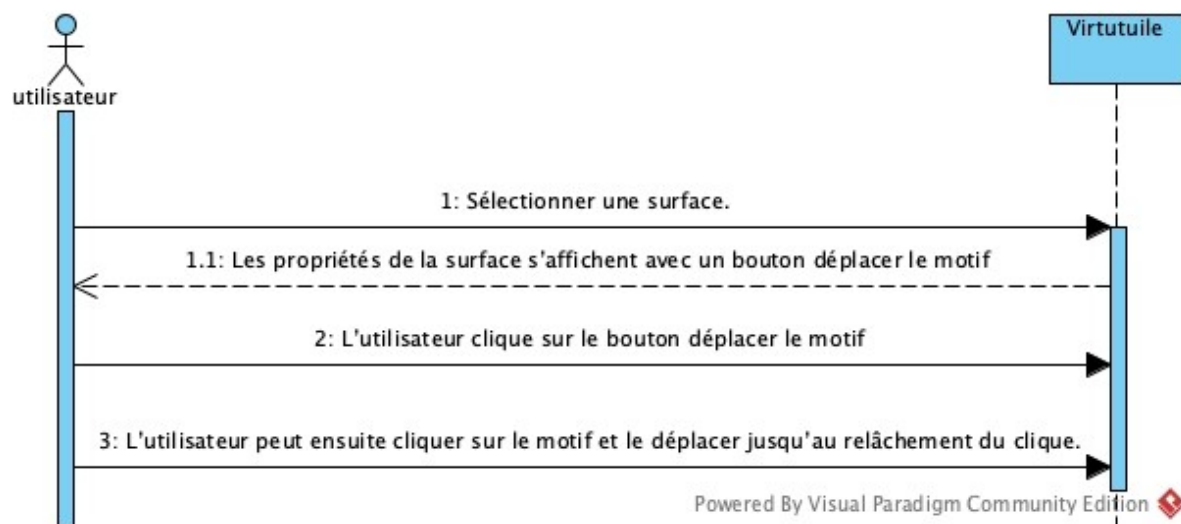
3. Éditer les propriétés des joints

Cas d'utilisation:	Éditer les propriétés des joints
Système:	VirtuTuile
Acteur(s)	Utilisateur
Partie prenante et intérêts:	Utilisateur: L'utilisateur essaye un nouveau style de joint
Précondition(s):	Avoir créer une surface et avoir sélectionné un motif pour cette surface
Garanties en cas de succès :	La jointure a été modifiée sur la surface
Scénario principal:	<p>1.Sélectionner une surface.</p> <p>2.Les propriétés du joint s'affichent.</p> <p>3.L'utilisateur clique sur la propriété à modifier</p> <p>4.Un input s'ouvre pour modifier la valeur de la propriété</p> <p>5.L'utilisateur tape la nouvelle valeur dans l'input.</p> <p>6.L'utilisateur appuie sur la touche entrée pour valider sa nouvelle valeur.</p>
Scénario(s) alternatif(s) :	



4. Repositionner manuellement avec la souris le motif sur la surface

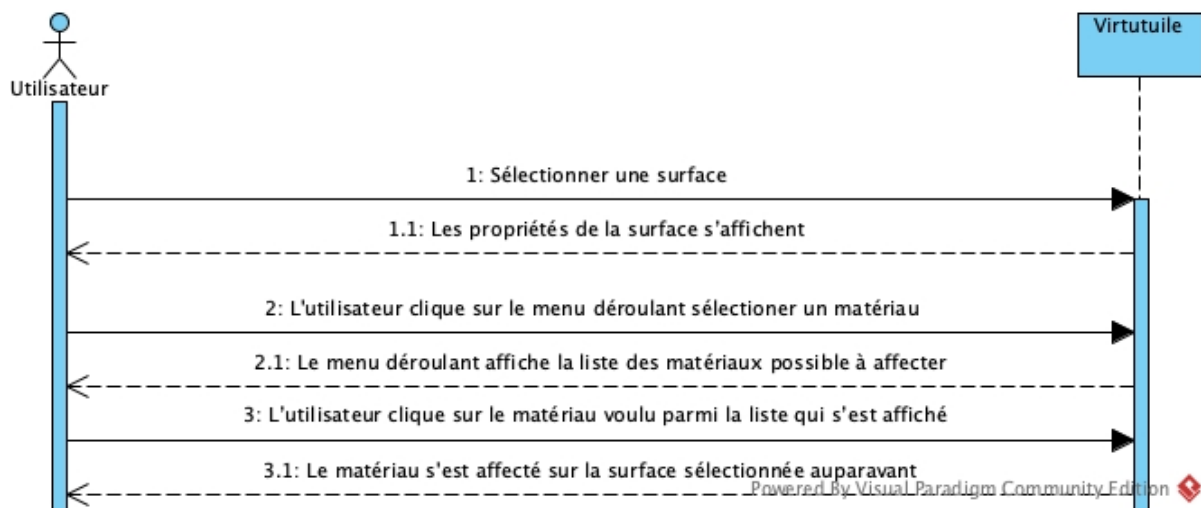
Cas d'utilisation:	Repositionner manuellement avec la souris le motif sur la surface
Système:	VirtuTuile
Acteur(s)	Utilisateur
Partie prenante et intérêts:	Utilisateur: Il veut essayer une nouvelle disposition du motif sur la surface.
Précondition(s):	Avoir créer une surface et avoir sélectionné un motif pour cette surface
Garanties en cas de succès :	Le motif a bien la nouvelle disposition
Scénario principal:	<p>1.Sélectionner une surface.</p> <p>2.Les propriétés de la surface s'affichent avec un bouton déplacer le motif.</p> <p>3.L'utilisateur clique sur le bouton déplacer le motif.</p> <p>4.L'utilisateur peut ensuite cliquer sur le motif et le déplacer jusqu'au relâchement du clique.</p>
Scénario(s) alternatif(s):	



Powered By Visual Paradigm Community Edition

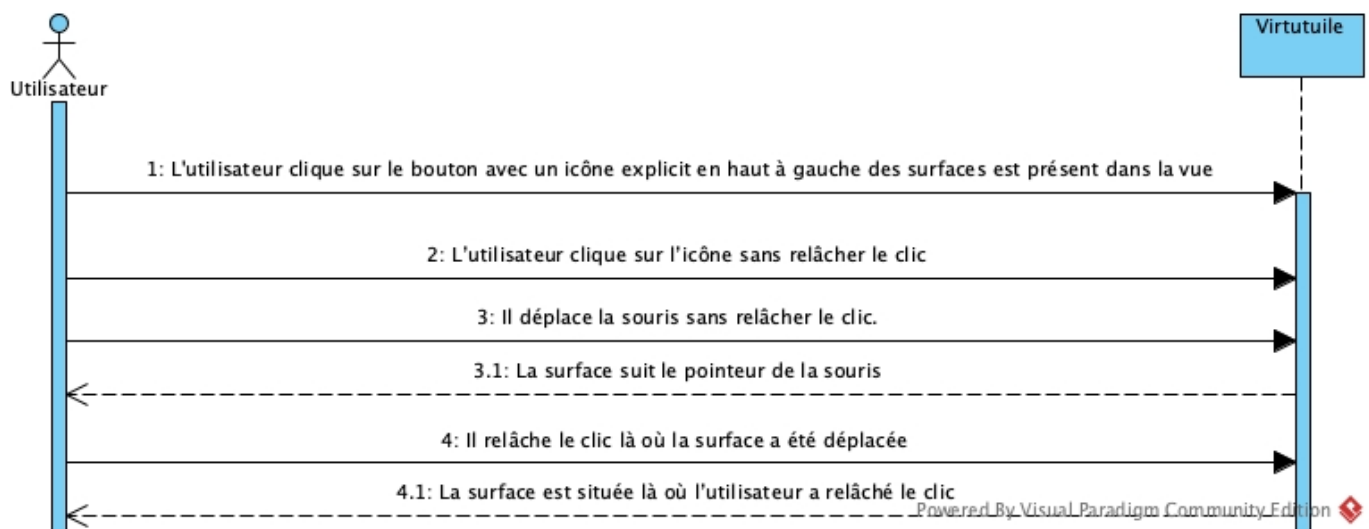
5. Assigner un matériau à une surface

Cas d'utilisation:	Assigner un matériau à une surface
Système:	VirtuTuile
Acteur(s)	Utilisateur
Partie prenante et intérêts:	Utilisateur: L'utilisateur veut voir le rendu des tuiles avec un matériau.
Précondition(s):	Avoir créer une surface et avoir sélectionné un motif pour cette surface
Garanties en cas de succès :	Le motif a bien la nouvelle disposition
Scénario principal:	<p>1.Sélectionner une surface.</p> <p>2.Les propriétés de la surface s'affichent.</p> <p>3.Un menu déroulant est présent dans les propriétés de la surface avec écrit sélectionner un matériau.</p> <p>4.L'utilisateur clique sur le menu déroulant.</p> <p>5.La liste des matériaux s'affiche.</p> <p>6.L'utilisateur clique sur le matériau voulu parmi la liste qui s'est affiché.</p>
Scénario(s) alternatif(s):	



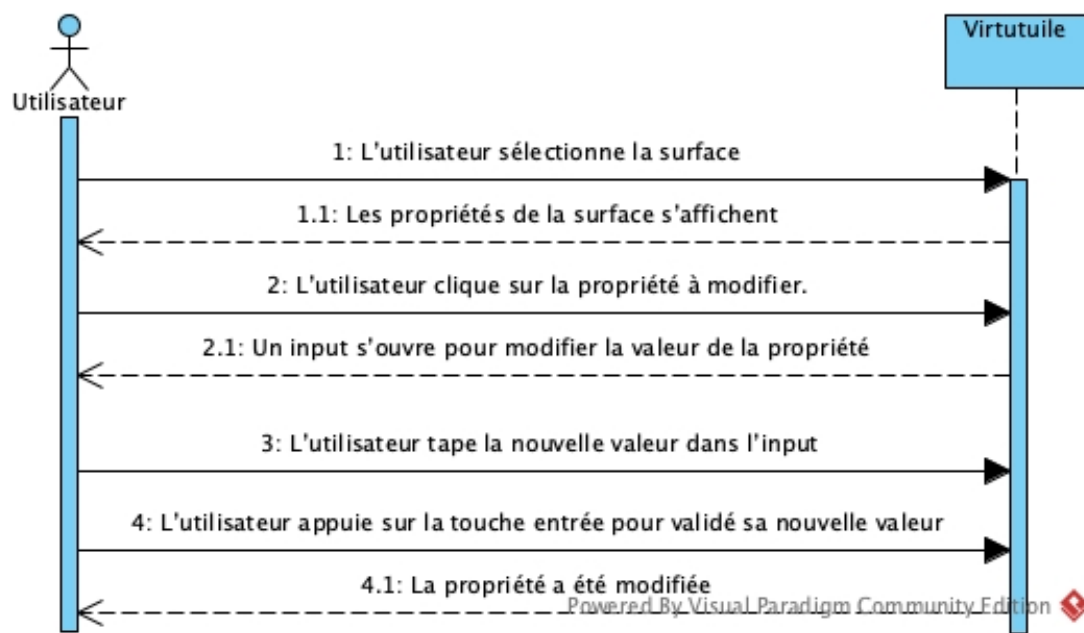
6. Repositionner manuellement avec la souris une surface

Cas d'utilisation:	Repositionner manuellement avec la souris une surface
Système:	VirtuTuile
Acteur(s)	Utilisateur
Partie prenante et intérêts:	Utilisateur: il veut essayer une nouvelle disposition des surfaces pour mieux représenter la pièce à recouvrir.
Précondition(s):	Avoir créer une surface.
Garanties en cas de succès :	La surface a bien la nouvelle position voulue.
Scénario principal:	<p>1.Un bouton avec un icône explicite en haut à gauche des surfaces est présent dans la vue.</p> <p>2.L'utilisateur clique sur l'icône sans relâcher le clic.</p> <p>3.Il déplace le pointeur de sa souris sans relâcher le clic.</p> <p>4.La surface suit le pointeur de la souris.</p> <p>5.Il relâche le clic là où la surface a été déplacé.</p> <p>6.La surface est située là où l'utilisateur a relâché le clic.</p>
Scénario(s) alternatif(s):	



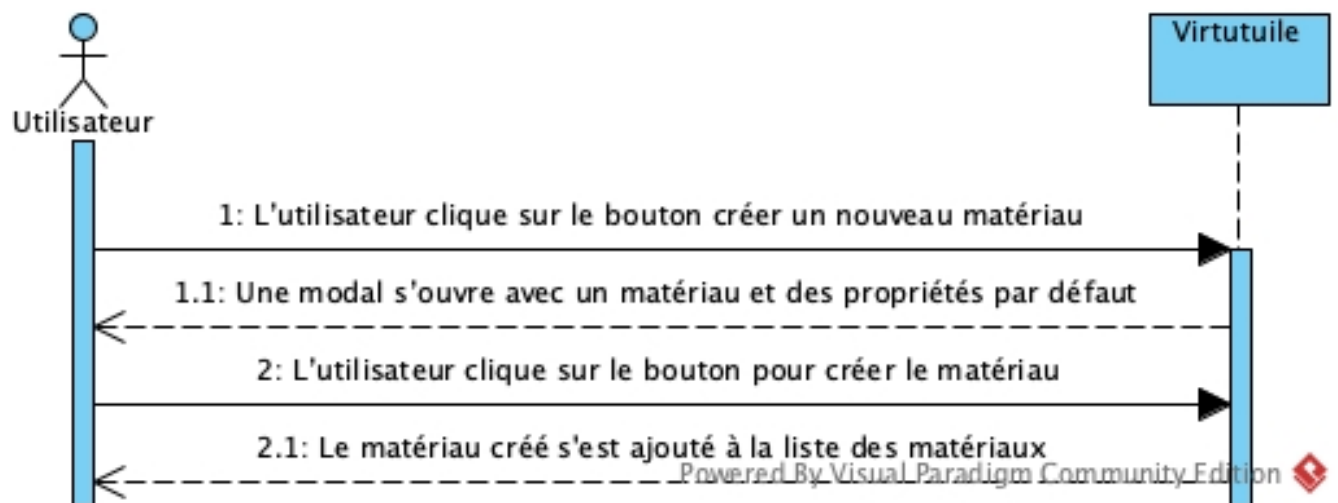
7. Modifier manuellement les propriétés de la surface

Cas d'utilisation:	Modifier manuellement les propriétés de la surface
Système:	VirtuTuile
Acteur(s)	Utilisateur
Partie prenante et intérêts:	Utilisateur: il veut que la surface soit de la même taille que la surface de sa pièce.
Précondition(s):	Avoir créer une surface.
Garanties en cas de succès :	La surface a bien sauvegardé les nouvelles propriétés.
Scénario principal:	<p>1.L'utilisateur sélectionne la surface.</p> <p>2.Les propriétés de la surface s'affichent.</p> <p>3.L'utilisateur clique sur la propriété à modifier.</p> <p>4.Un input s'ouvre pour modifier la valeur de la propriété</p> <p>5.L'utilisateur tape la nouvelle valeur dans l'input.</p> <p>6.L'utilisateur appuie sur la touche entrée pour validée sa nouvelle valeur.</p>
Scénario(s) alternatif(s):	



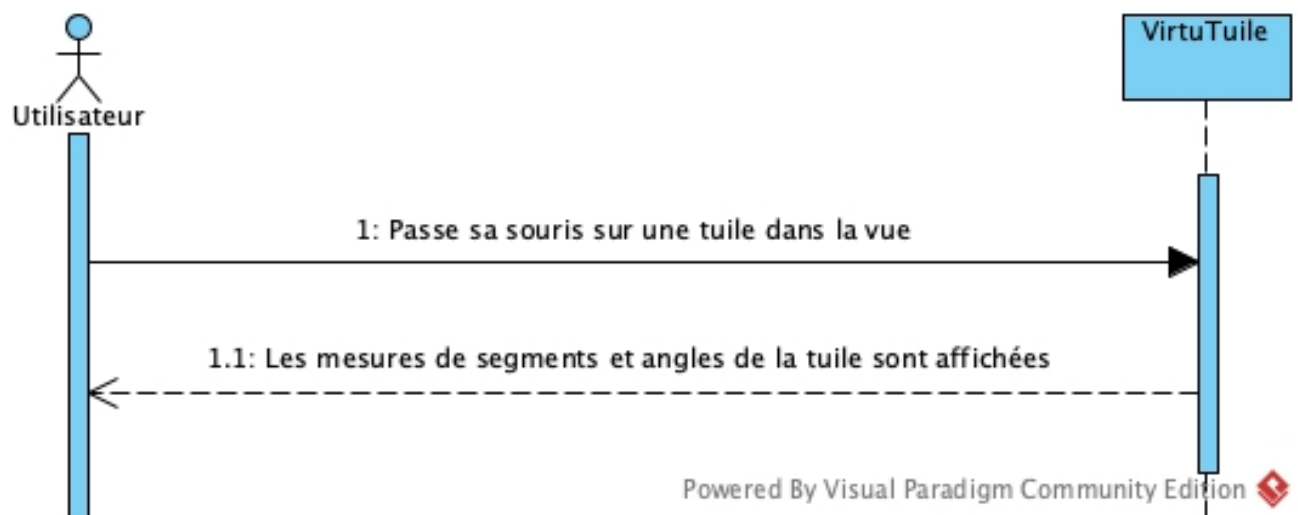
8. Créer un matériau

Cas d'utilisation:	Créer un matériau
Système:	VirtuTuile
Acteur(s)	Utilisateur
Partie prenante et intérêts:	Utilisateur: Il veut créer un nouveau matériau semblable à celui qu'il veut poser dans sa pièce.
Précondition(s):	Aucune
Garanties en cas de succès :	Le matériau a bien ajouté à la liste des matériaux
Scénario principal:	<p>1.L'utilisateur clique sur le bouton créer un nouveau matériau.</p> <p>2.Une modal s'ouvre avec un matériau et des propriétés par default.</p> <p>3.L'utilisateur clique sur le bouton créer le matériau.</p>
Scénario(s) alternatif(s):	



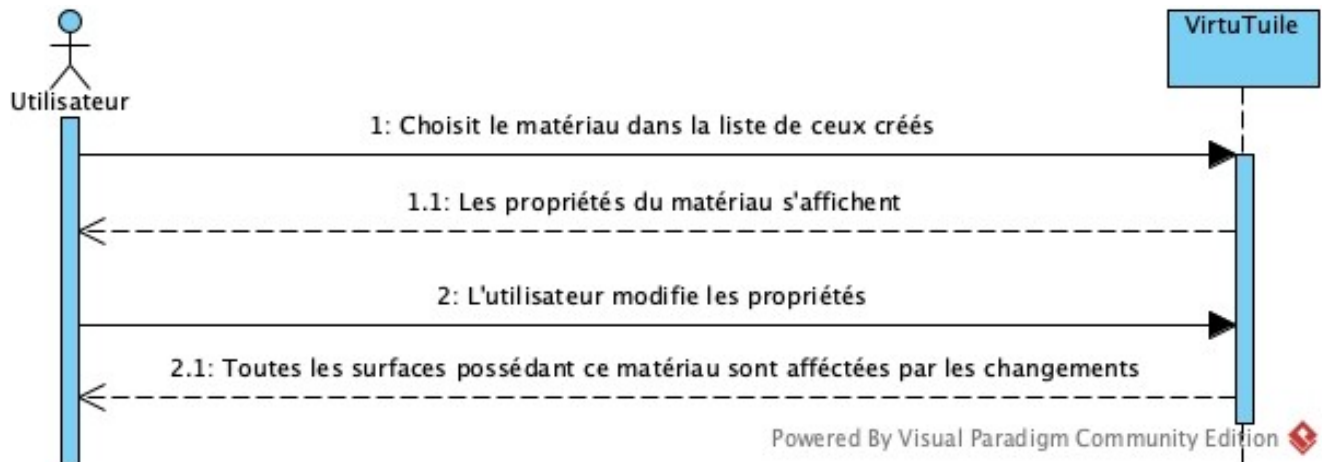
9. Afficher les dimensions d'une tuile

Cas d'utilisation :	Afficher les dimensions d'une tuile
Système :	VirtuTuile
Acteurs(s) :	Utilisateur
Partie prenante et intérêts :	Utilisateur : L'utilisateur veut connaître les dimensions d'une tuile
Préconditions :	<ul style="list-style-type: none"> - L'utilisateur passe sa souris sur une tuile dans la vue - L'outil 'sélection' est actif
Garanties en cas de succès :	Les mesures et les angles entre les segments s'affichent (sur la tuile ? dans la box ?)
Scénario principal :	1. L'utilisateur passe sa souris sur une tuile de la vue. 2. Ses mesures de segments et angles sont affichées
Scénario(s) alternatifs :	Aucun



10. Éditer les propriétés des matériaux

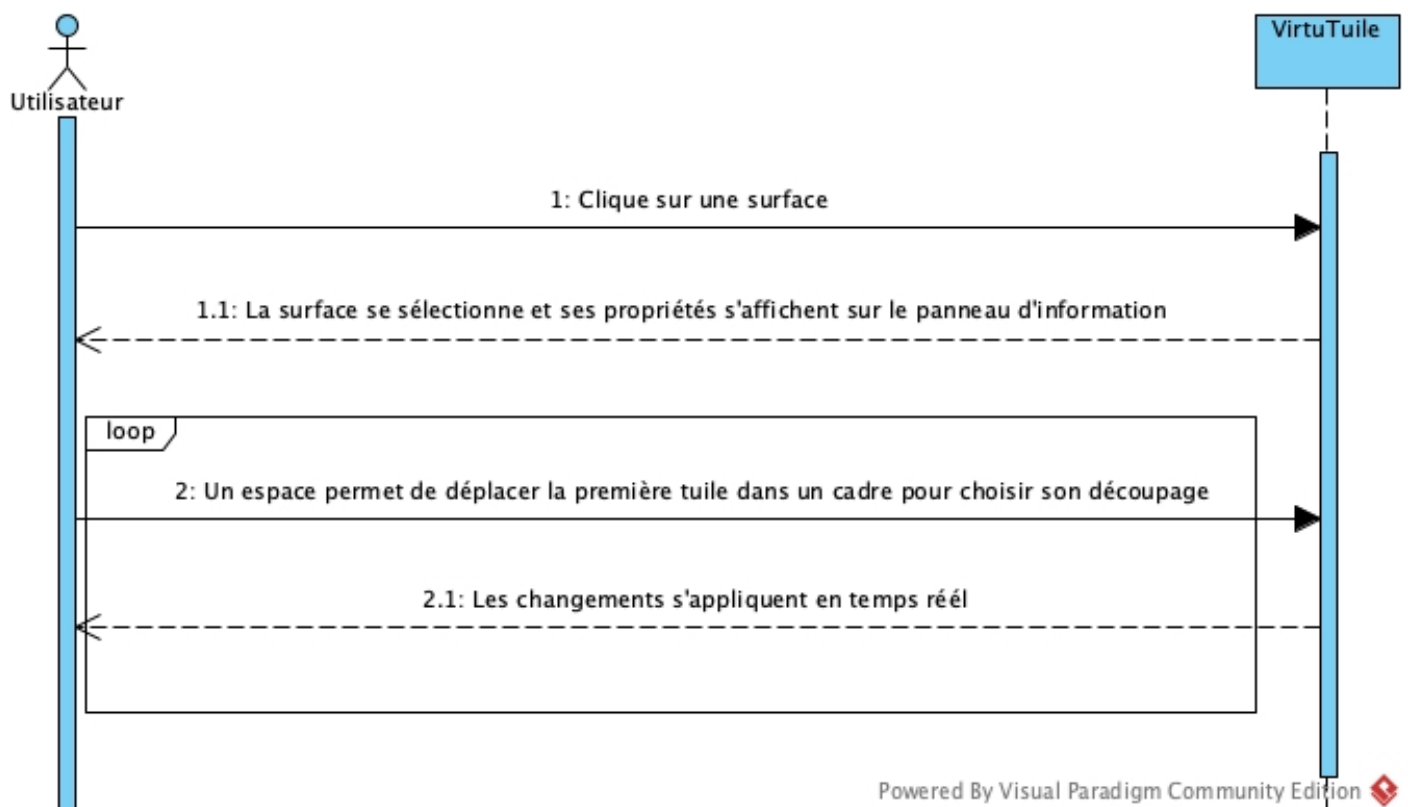
Cas d'utilisation :	Editer les propriétés des matériaux
Système :	VirtuTuile
Acteurs(s) :	Utilisateur
Partie prenante et intérêts :	Utilisateur : Corriger ou modifier les propriétés d'un des matériaux de son ouvrage
Préconditions :	Utilisateur : Avoir créé au moins un matériau
Garanties en cas de succès :	Les modifications sont appliquées sur toutes les surfaces possédant ce matériau
Scénario principal :	<p>1.L'utilisateur choisit le matériau dans la liste de matériaux créés</p> <p>2.Les propriétés du matériau s'affichent dans un modal</p> <p>3.L'utilisateur modifie les propriétés</p> <p>4.Toutes les surfaces possédant ce matériau sont affectées par les changements</p>
Scénario(s) alternatifs :	Aucun



Powered By Visual Paradigm Community Edition

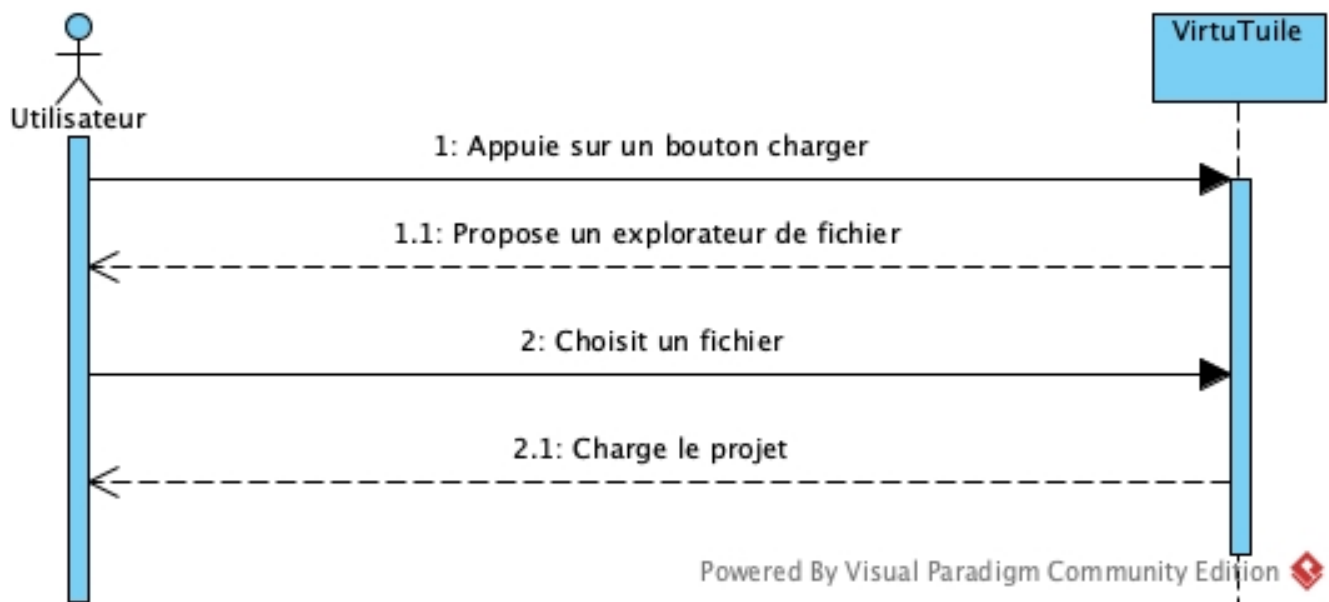
11. Indiquer comment le motif débute

Cas d'utilisation :	Indiquer comment le motif débute
Système :	VirtuTuile
Acteurs(s) :	Utilisateur
Partie prenante et intérêts :	Utilisateur : Arranger l'aspect visuel de son ouvrage
Préconditions :	Avoir une surface avec un motif
Garanties en cas de succès :	Le motif est déplacé sur la surface
Scénario principal :	<p>1.L'utilisateur clique sur une surface</p> <p>2.La surface se sélectionne et ses propriétés s'affichent sur le panneau de droite</p> <p>3.Un espace permet de déplacer la première tuile dans un cadre pour choisir son découpage</p> <p>4.Les changements s'appliquent en temps réel</p>
Scénario(s) alternatifs :	Aucun



12. Charger le projet depuis un fichier

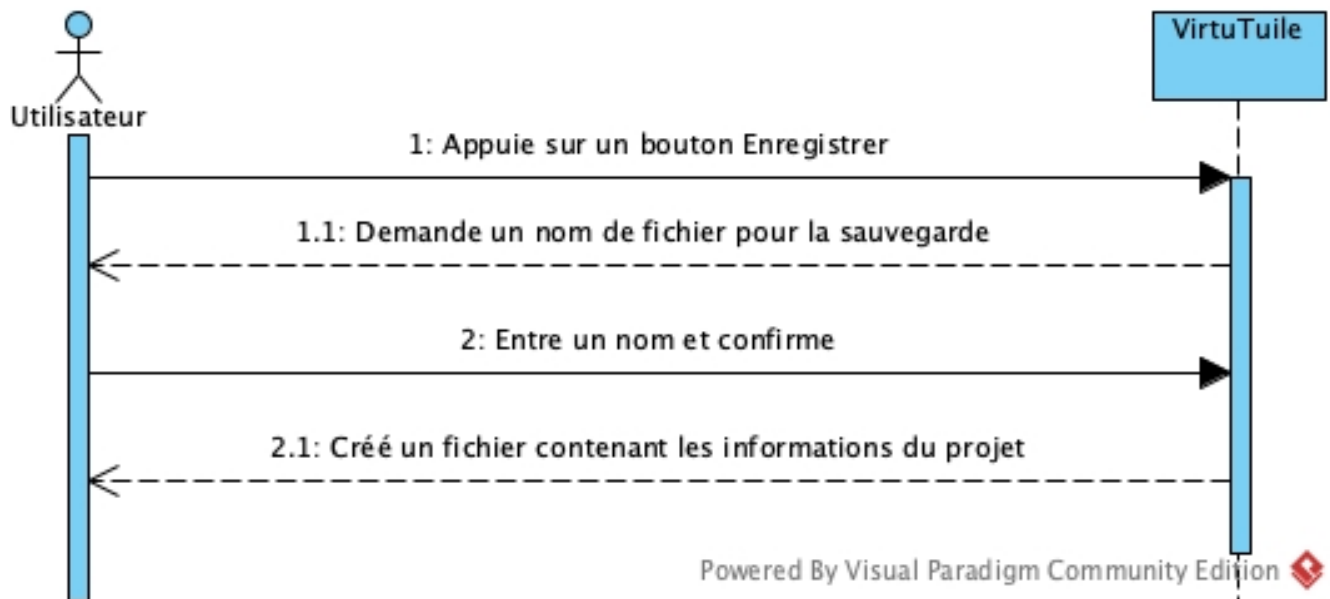
Cas d'utilisation :	Charger le projet depuis un fichier
Système :	VirtuTuile
Acteurs(s) :	Utilisateur
Partie prenante et intérêts :	Utilisateur : Pouvoir visualiser et manipuler un projet sauvegardé auparavant
Préconditions :	Avoir un fichier de sauvegarde
Garanties en cas de succès :	Le projet enregistré dans le fichier et chacun des éléments le composant sont chargés dans la vue
Scénario principal :	1.L'utilisateur appuie sur un bouton 'charger' 2.Le logiciel propose des fichiers depuis un explorateur 3.L'utilisateur choisit un fichier 4.Le logiciel charge le projet
Scénario(s) alternatifs :	1. Le fichier est invalide, l'opération est invalidée et l'utilisateur notifié



Powered By Visual Paradigm Community Edition

13. Enregistrer le projet dans un fichier

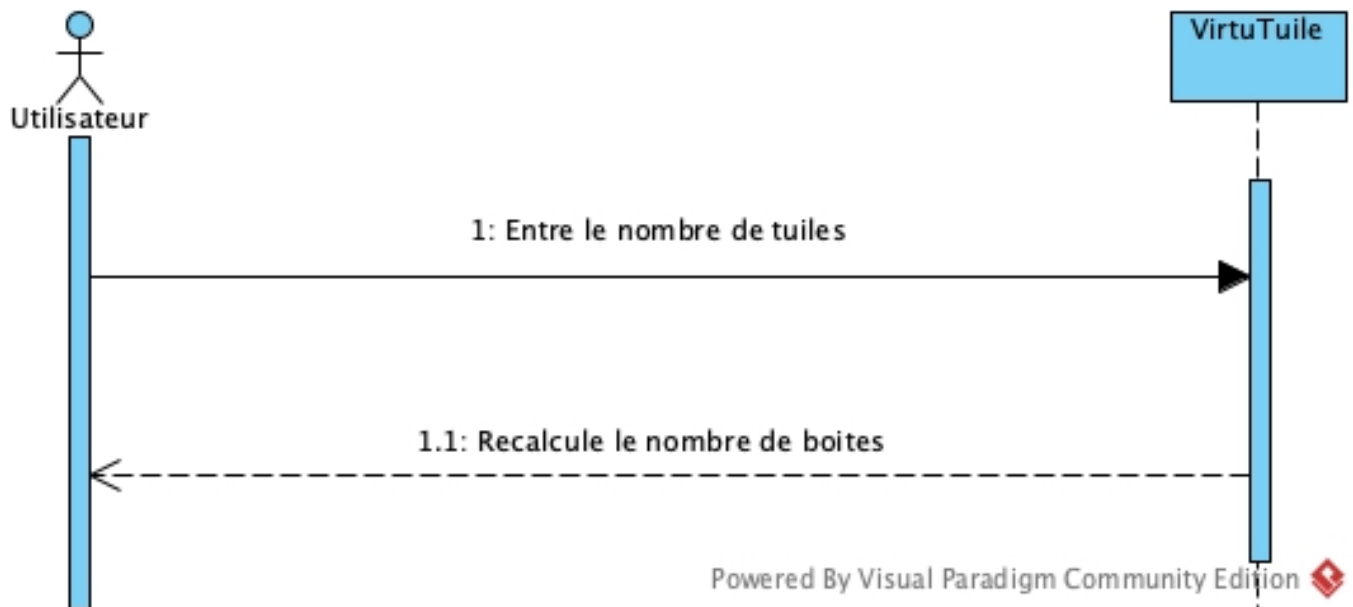
Cas d'utilisation :	Enregistrer le projet dans un fichier
Système :	VirtuTuile
Acteurs(s) :	Utilisateur
Partie prenante et intérêts :	Utilisateur : L'utilisateur veut pouvoir enregistrer son projet pour le réouvrir plus tard, le partager à d'autres etc...
Préconditions :	Aucune
Garanties en cas de succès :	Un fichier est écrit avec les informations nécessaires pour décrire l'entièreté du projet en cours
Scénario principal :	<p>1.L'utilisateur appuie sur un bouton 'enregistrer'</p> <p>2.Le logiciel demande un nom pour le fichier de sauvegarde</p> <p>3.L'utilisateur entre le nom du fichier et confirme</p> <p>4.Le logiciel crée un fichier contenant les informations du projet</p>
Scénario(s) alternatifs :	Aucun



Powered By Visual Paradigm Community Edition

14. Spécifier le nombre de tuiles par boîtes

Cas d'utilisation :	Spécifier le nombre de tuiles par boîtes
Système :	VirtuTuile
Acteurs(s) :	Utilisateur
Partie prenante et intérêts :	Utilisateur : L'utilisateur veut pouvoir savoir combien de boîtes de tuiles doit t-il acheter pour couvrir sa surface
Préconditions :	Aucune
Garanties en cas de succès :	Le nombre de boîtes de tuiles pour chaque surface est affiché
Scénario principal :	1.L'utilisateur entre le nombre de tuiles (champs permanent) 2.Le logiciel recalcule le nombre de boîtes
Scénario(s) alternatifs :	Aucun



15. Afficher le nombre de tuiles, boîtes et de surface d'enduit requises

Cas d'utilisation :	Afficher le nombre de tuiles, boîtes et de surface d'enduit requises
Système :	VirtuTuile
Acteurs(s) :	Utilisateur
Partie prenante et intérêts :	Utilisateur : Connaitre la quantité de matériaux à acheter pour réaliser son ouvrage
Préconditions :	Aucune
Garanties en cas de succès :	Les informations de tuiles, boîtes et surface d'enduit sont affichées
Scénario principal :	Aucun – affiché en permanence
Scénario(s) alternatifs :	Aucun

16. Afficher la grille magnétique

Cas d'utilisation	Afficher la grille magnétique
Acteur	Utilisateur
Type	Secondaire
Description	En cliquant sur une icône, il est possible pour l'utilisateur d'afficher ou d'enlever une grille.

17. Afficher les propriétés de la surface

Cas d'utilisation	Afficher les propriétés de la surface
Acteur	Aucun
Type	Secondaire
Description	Dans le panel de droite, les propriétés de la surface (dimension, motif) seront affichées

18. Coller et décoller deux surfaces

Cas d'utilisation	Coller et décoller deux surfaces horizontalement et verticalement
Acteur	Utilisateur
Type	Secondaire
Description	En sélectionnant deux surfaces avec la souris, il doit être possible pour l'utilisateur de les coller. Et en sélectionnant deux surfaces collées, il doit être possible de les décoller.

19. Aligner deux surfaces

Cas d'utilisation	Aligner deux surfaces haut/bas/gauche/droite
Acteur	Utilisateur
Type	Secondaire
Description	En déplaçant une surface avec la souris, il doit être possible de l'aligner avec une autre surface.

20. Spécifier les coordonnées d'une surface

Cas d'utilisation	Spécifier les coordonnées d'une surface en fonction du coin supérieur gauche d'une autre surface
Acteur	Utilisateur
Type	Secondaire
Description	Spécifier les coordonnées d'une surface en fonction du coin supérieur gauche d'une autre surface

21. Afficher les propriétés des joints

Cas d'utilisation	Afficher les propriétés des joints
Acteur	Aucun
Type	Secondaire
Description	Dans le panel de droite, il doit être possible d'afficher les différentes propriétés des joints afin d'en informer l'utilisateur

22. Détection de tuile

Cas d'utilisation	Détection de tuile
Acteur	Utilisateur
Type	Secondaire
Description	Il doit être possible de détecter automatiquement les tuiles dont l'une des deux dimensions est inférieure à N unités entré par l'utilisateur

23. Afficher les propriétés d'un matériau

Cas d'utilisation	Afficher les propriétés d'un matériau
Acteur	Aucun
Type	Secondaire
Description	Dans le panel de droite, l'utilisateur pourra voir les propriétés d'un matériau (type) d'une surface

24. Combiner deux surfaces

Cas d'utilisation	Combiner deux surfaces
Acteur	Utilisateur
Type	Secondaire
Description	À l'aide d'un clic droit après avoir sélectionné 2 surfaces, l'utilisateur pourra combiner deux surfaces entre elles

25. Zoom

Cas d'utilisation	Zoom et dezoom
Acteur	Utilisateur
Type	Secondaire

Description	Avec la molette de la souris, l'utilisateur doit pouvoir zoomer ou dezommer la vue afin d'avoir plusieurs niveaux de détails sur son travail
-------------	--

26. Défaire une action

Cas d'utilisation	Défaire une action
Acteur	Utilisateur
Type	Secondaire
Description	À l'aide de boutons sur la barre d'outils, l'utilisateur doit pouvoir annuler sa dernière action, ou refaire une action annulée

27. Sélectionner un motif

Cas d'utilisation	Sélectionner un motif
Acteur	Utilisateur
Type	Secondaire
Description	L'utilisateur doit pouvoir sélectionner un motif qui sera appliqué sur la surface

28. Centrer deux surfaces

Cas d'utilisation	Centrer deux surfaces
Acteur	Utilisateur
Type	Secondaire
Description	À l'aide d'un clic droit après avoir sélectionné deux surfaces, l'utilisateur pour centrer deux surfaces entre elles

29. Aligner deux surfaces

Cas d'utilisation	Aligner deux surfaces haut/bas/gauche/droite
Acteur	Utilisateur
Type	Secondaire
Description	En déplaçant une surface avec la souris, il doit être possible de l'aligner avec une autre surface.

30. Spécifier les coordonnées d'une surface

Cas d'utilisation	Spécifier les coordonnées d'une surface en fonction au coin supérieur gauche d'une autre surface
Acteur	Utilisateur
Type	Secondaire
Description	

