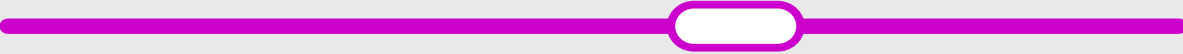


Mini Tramways

Fonctionnalités et conception

Léo VINCENT
Enzo RICHARD
Thomas COULON
Alexandre PERZO-JOLY

Présentation du projet



Mini Tramways est un jeu de type "Arcade" pour PC qui consiste en la construction et la gestion d'un réseau de tramways qui doit desservir l'ensemble des bâtiments de la carte. Dans ce jeu, le joueur devra créer des lignes de tramways pour relier différents quartiers entre eux (résidentiels, commerciaux ou quartiers d'affaires) de manière à ce que les habitants soient satisfaits. Si le réseau n'est pas suffisamment optimisé et que les voyageurs doivent attendre trop longtemps avant d'être transportés, le joueur perd la partie.

Les quartiers seront des zones prédéfinies sur la carte, mais les bâtiments qui les composent seront générés au fil de la partie, de manière aléatoire. Les stations de trams seront, quant à elles, placées par le joueur et reliées entre elles par des lignes. Chaque ligne nouvellement créée se verra attribuée un wagon, qui effectuera des aller-retours incessants d'un bout à l'autre de la ligne. Il sera possible d'ajouter d'autres stations à une ligne existante en la prolongeant.

Pour construire de nouvelles stations et lignes, le joueur devra dépenser de l'argent acquis en transportant des voyageurs. Un voyageur non satisfait (par exemple, s'il a dû attendre trop longtemps avant d'être transporté) demandera le remboursement de son billet, et ne fera donc pas gagner d'argent au joueur.

Éventuellement, un ensemble de fonctionnalités supplémentaires sont envisageables, comme par exemple la sauvegarde de la partie ou la génération aléatoire des quartiers.

Table des matières

Backlog produit	4
Fonctionnalités vitales	4
Fonctionnalités importantes	4
Fonctionnalités additionnelles	5
Cas d'utilisation	6
1. Diagramme de cas d'utilisation	6
Classes métier	7
2. Diagramme de classes métier	7
Détail des classes métier	8
3. Game	8
4. Map	10
5. Cell	12
6. Building	14
7. Station	16
8. Line	17
9. Tramway	19
10. Player	21
11. Vector2	23
12. Area	25
Séquences	27
1. Lancer partie	27
2. Créer une ligne	28

Backlog produit

Fonctionnalités vitales

Ces fonctionnalités sont nécessaires au bon fonctionnement du jeu. Ce sont celles qui seront développées en priorité.

Carte et bâtiments		Transport	
			Création d'une ligne
1	Affichage de la grille	4:1	Sélection des coordonnées de départ/arrivée
2:1	Affichage des quartiers	5:4	Création de la ligne et attribution d'un Tramway
3:2	Génération d'un bâtiment	6:5	Génération de stations aux intersections
		7:5	Étendre une ligne à une autre station
			Tramways
		8:5	Parcours d'une ligne
		9:5,3	Transport des voyageurs
			Stations
		10:5	Affichage du rayon au survol
Voyageurs		Autres	
	Génération et IA		Affichage
11:3	Apparition des voyageurs dans les maisons	14:9	Affichage du nombre de personnes dans un tram ou une station
12:11,8,9	Suivi du chemin jusqu'à la cible		
13:12	Calcul de la satisfaction		

Fonctionnalités importantes

Ces fonctionnalités ajouteraient beaucoup de confort et de re-jouabilité au jeu, mais ne sont pas indispensables à son fonctionnement.

Carte et bâtiments		Transport	
			Modification d'une ligne
15:2	Quartiers aléatoires	17:7	Faire faire un détour par une autre station
16:3	Plusieurs designs par type de bâtiments	18:5	Supprimer une portion de ligne
		19:18	Détruire les stations inutiles automatiquement
Autres			
	Score		

20:13	Gestion du score
21:20	Sauvegarde du record

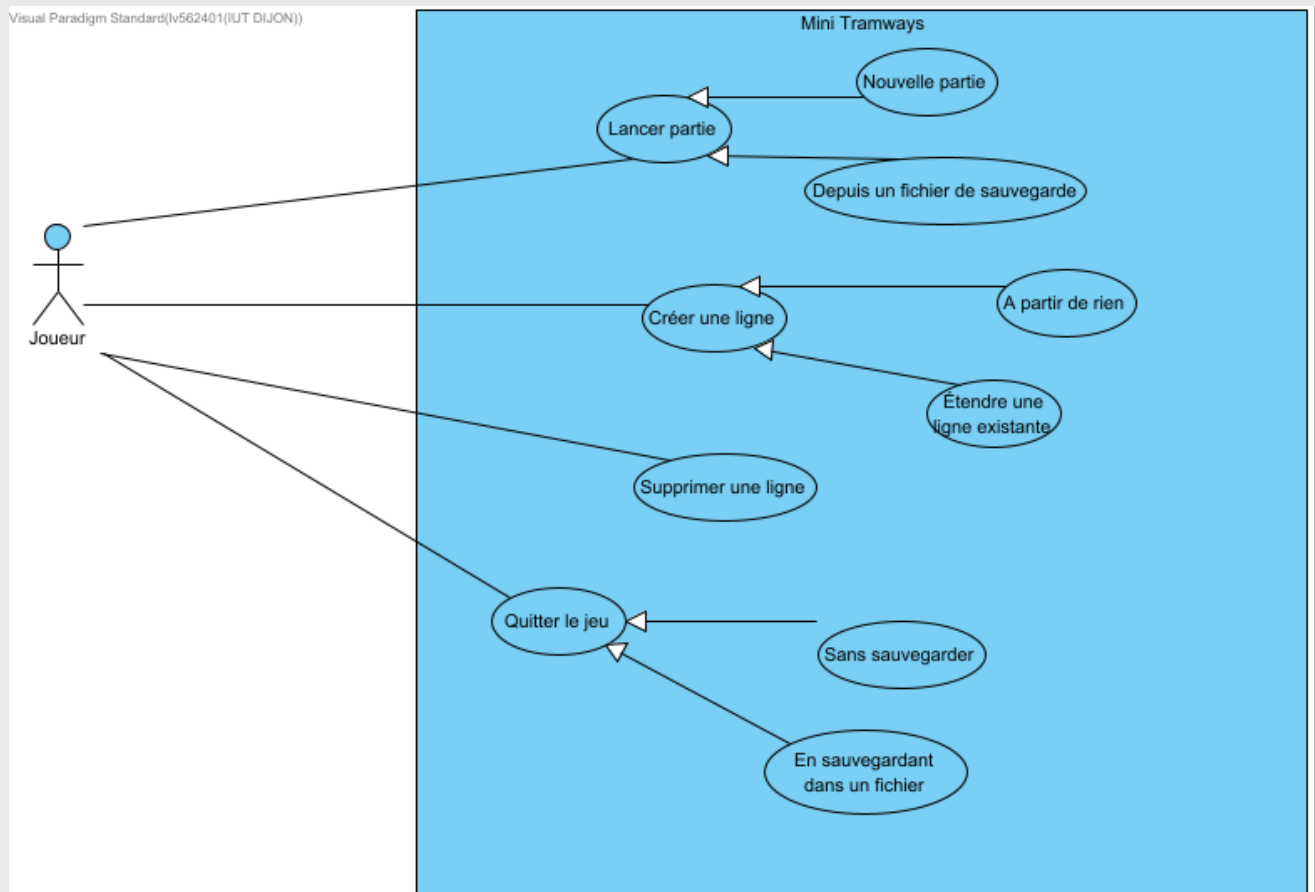
Fonctionnalités additionnelles

Ces fonctionnalités ajoutent encore plus de confort pour le joueur, ainsi que de nouvelles options de progression. Elles ne sont absolument pas indispensables, mais peuvent améliorer grandement la qualité du produit.

Carte et bâtiments		Transport	
	Quartiers		Modification d'une ligne
22:2	Évolution des quartiers	23:5	Amélioration du niveau d'une ligne (vitesse)
			Stations
		24:5	Amélioration d'une station (rayon)
			Tramways
		25:5	Ajout de tramways à une ligne
		26:5	Amélioration d'un wagon (capacité)
Autres			
	Vitesse de déroulement		
27	Accélérer le temps		
28	Arrêter le temps		
	Sauvegarde		
29	Sauvegarder une partie dans un fichier		
30:29	Reprendre une partie automatiquement après avoir quitté puis relancé le jeu		
31:29	Charger une partie depuis un fichier		

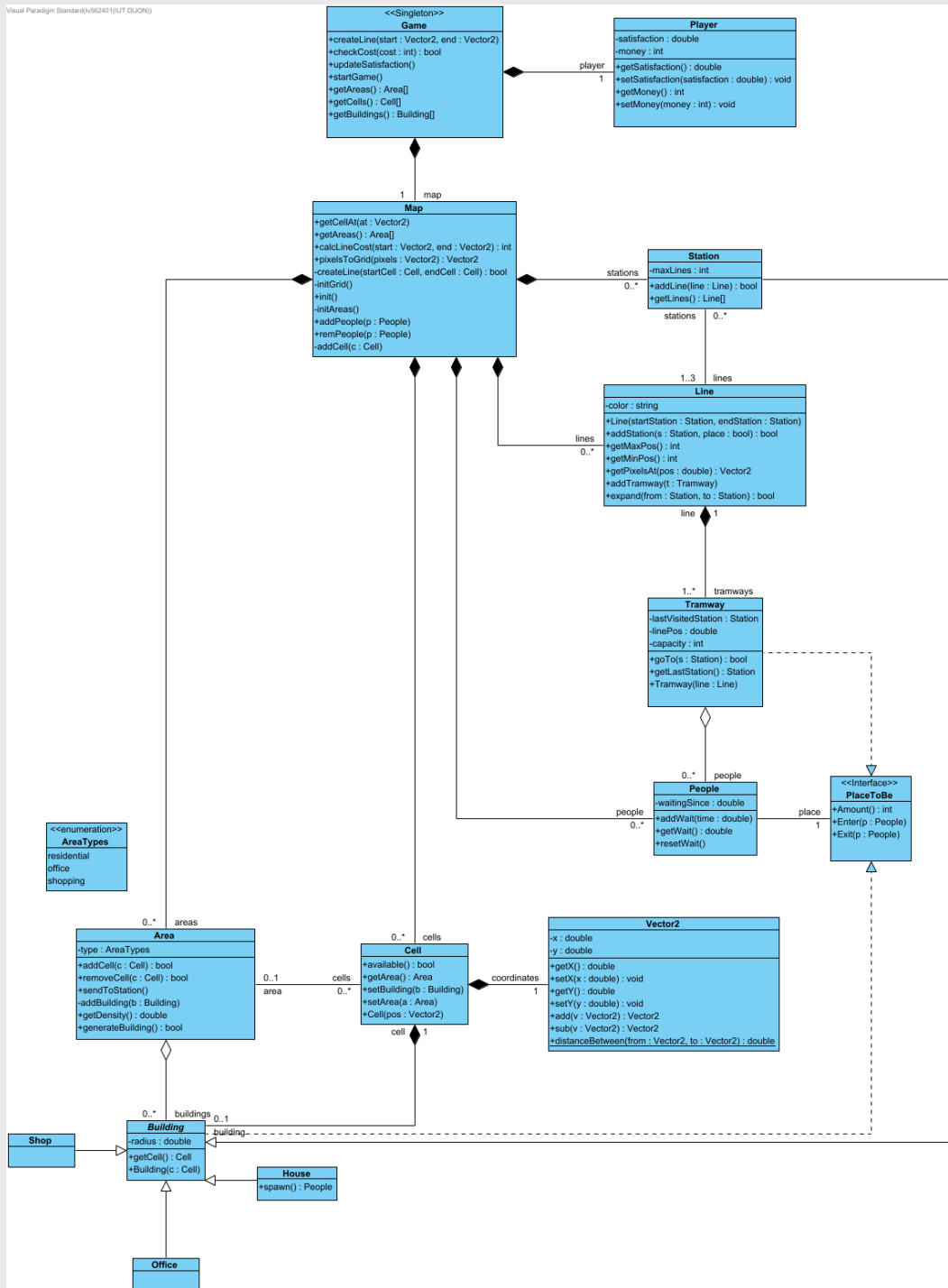
Cas d'utilisation

1. Diagramme de cas d'utilisation



Classes métier

2. Diagramme de classes métier



Détail des classes métier

3. Game

Gère le déroulement du jeu et fait office d'interface pour l'IHM.








3.1. Stéréotypes

<<Singleton>>

3.2. Propriétés

Visibility	public
Abstract	false
Leaf	false
Root	false
Active	false

3.3. Résumé des méthodes

Name	Description
 checkCost	Vérifie si le joueur dispose de suffisamment d'argent
 createLine	
 getAreas	
 getBuildings	
 getCells	
 startGame	
 updateSatisfaction	Met à jour la satisfaction selon l'attente moyenne des personnes à cet instant.

3.4. Méthodes

3.4.1. checkCost

Vérifie si le joueur dispose de suffisamment d'argent

Visibility	public
Return Type	bool

3.4.2. createLine

Visibility	public
-------------------	--------

3.4.3. getAreas

Visibility	public
Return Type	Area[]

3.4.4. getBuildings

Visibility	public
Return Type	Building[]

3.4.5. getCells

Visibility	public
Return Type	Cell[]

3.4.6. startGame





Visibility	public
-------------------	--------

3.4.7. updateSatisfaction

Met à jour la satisfaction selon l'attente moyenne des personnes à cet instant.

Visibility	public
-------------------	--------

3.5. Relations

Relationship	From	To
— unnamed	 Game	 Map
— unnamed	 Game	 Player












4. Map

Gère la carte et son contenu

4.1. Propriétés

Visibility	public
Abstract	false
Leaf	false
Root	false
Active	false

4.2. Résumé des méthodes

Name	Description
 addCell	
 addPeople	Ajoute une personne à la carte (l'emplacement sera décidé ensuite)
 calcLineCost	Calcule le coût de construction d'une ligne
 createLine	Crée une ligne entre deux cases
 getAreas	Retourne l'ensemble des zones composant la carte
 getCellAt	Retourne la case correspondant aux coordonnées passées en paramètre
 init	Lance la procédure d'initialisation de la carte (grille, zones)
 initAreas	Initialise les zones de la carte
 initGrid	Initialise la grille
 pixelsToGrid	Convertit des coordonnées en pixels sur l'écran en coordonnées de la grille
 remPeople	Retire une personne de la carte (et de l'emplacement où elle se trouve actuellement)

4.3. Méthodes

4.3.1. addCell

Visibility	private
-------------------	---------

4.3.2. addPeople

Ajoute une personne à la carte (l'emplacement sera décidé ensuite)

Visibility	public
-------------------	--------

4.3.3. calcLineCost

Calcule le coût de construction d'une ligne

Visibility	public
Return Type	int

4.3.4. createLine

Crée une ligne entre deux cases

Visibility	private
Return Type	bool

4.3.5. getAreas

Retourne l'ensemble des zones composant la carte

Visibility	public
Return Type	Area[]

4.3.6. getCellAt

Retourne la case correspondant aux coordonnées passées en paramètre

Visibility	public
-------------------	--------

4.3.7. init

Lance la procédure d'initialisation de la carte (grille, zones)

Visibility	public
-------------------	--------

4.3.8. initAreas

Initialise les zones de la carte

Visibility	private
-------------------	---------

4.3.9. initGrid

Initialise la grille

Visibility	private
-------------------	---------

4.3.10. pixelsToGrid

Convertit des coordonnées en pixels sur l'écran en coordonnées de la grille













Visibility	public
Return Type	Vector2

4.3.11. remPeople

Retire une personne de la carte (et de l'emplacement où elle se trouve actuellement)

Visibility	public
-------------------	--------

4.4. Relations

Relationship	From	To
— unnamed	 Map	 Cell
— unnamed	 Map	 People
— unnamed	 Map	 Line
— unnamed	 Map	 Station
— unnamed	 Map	 Area
— unnamed	 Game	 Map






5. Cell

Représente une case de la grille

5.1. Propriétés

Visibility	public
Abstract	false
Leaf	false
Root	false
Active	false

5.2. Résumé des méthodes

Name	Description
 available	Retourne true si aucun bâtiment n'est présent sur cette case
 Cell	
 getArea	Retourne la zone à laquelle la case appartient
 setArea	Définit la zone à laquelle appartient cette case
 setBuilding	Définit le bâtiment occupant la zone

5.3. Méthodes

5.3.1. available

Retourne true si aucun bâtiment n'est présent sur cette case

Visibility	public
Return Type	bool

5.3.2. Cell

Visibility	public
-------------------	--------

5.3.3. getArea

Retourne la zone à laquelle la case appartient

Visibility	public
Return Type	Area

5.3.4. setArea

Définit la zone à laquelle appartient cette case









Visibility	public
-------------------	--------

5.3.5. setBuilding

Définit le bâtiment occupant la zone

Visibility	public
-------------------	--------

5.4. Relations

Relationship	From	To
— unnamed	 Cell	 Building
— unnamed	 Cell	 Vector2
— unnamed	 Map	 Cell
— unnamed	 Area	 Cell


6. Building

Représente un bâtiment



6.1. Propriétés

Visibility	public
Abstract	true
Leaf	false
Root	false
Active	false

6.2. Attributes Summary

Name	Description
 radius	Rayon dans lequel la station est en mesure de desservir les bâtiments

6.3. Résumé des méthodes

Name	Description
 Building	
 getCell	Retourne la case correspondant au bâtiment

6.4. Attributes

6.4.1. radius

Rayon dans lequel la station est en mesure de desservir les bâtiments

Visibility	private
Type	double

6.5. Méthodes

6.5.1. Building















Visibility	public
-------------------	--------

6.5.2. getCell

Retourne la case correspondant au bâtiment

Visibility	public
Return Type	Cell

6.6. Relations

Relationship	From	To
↳ unnamed	 Building	 House
↳ unnamed	 Building	 Office
↳ unnamed	 Building	 Shop
↳ unnamed	 Building	 Station
↳ ^R unnamed	 PlaceToBe	 Building
— unnamed	 Cell	 Building
— unnamed	 Area	 Building


7. Station

Représente une station de trams



7.1. Propriétés

Visibility	public
Abstract	false
Leaf	false
Root	false
Active	false

7.2. Attributes Summary

Name	Description
 maxLines	Nombre maximal de lignes qui peuvent desservir la station

7.3. Résumé des méthodes

Name	Description
 addLine	Ajoute une ligne à la station @return true si succès
 getLines	Retourne l'ensemble des lignes qui desservent par la station

7.4. Attributes

7.4.1. maxLines

Nombre maximal de lignes qui peuvent desservir la station

Visibility	private
Type	int

7.5. Méthodes

7.5.1. addLine

Ajoute une ligne à la station

Retourne true si succès







Visibility	public
Return Type	bool

7.5.2. getLines

Retourne l'ensemble des lignes qui desservent par la station

Visibility	public
Return Type	Line[]

7.6. Relations

Relationship	From	To
— unnamed	 Station	 Line
← unnamed	 Building	 Station
— unnamed	 Map	 Station


8. Line

Représente une ligne de trams








8.1. Propriétés

Visibility	public
Abstract	false
Leaf	false
Root	false
Active	false

8.2. Attributes Summary

Name	Description
 color	Code couleur de la ligne

8.3. Résumé des méthodes

Name	Description
 addStation	Ajoute une station au début ou à la fin de la ligne @param place définit si on doit placer la station en début (true) ou en fin (false) de ligne @return true si succès
 addTramway	Ajoute un tramway à la ligne
 expand	Étend une ligne @from la station depuis laquelle la ligne doit être étendue @to la station vers laquelle la ligne doit être étendue
 getMaxPos	Retourne la valeur de position maximale possible sur cette ligne
 getMinPos	Retourne la valeur de position minimale possible sur cette ligne
 getPixelsAt	Retourne les coordonnées sur l'écran correspondant à une position sur la ligne
 Line	Crée une nouvelle ligne entre deux stations

8.4. Attributes

8.4.1. color

Code couleur de la ligne

Visibility	private
Type	string

8.5. Méthodes

8.5.1. addStation

Ajoute une station au début ou à la fin de la ligne

Le paramètre place définit si on doit placer la station en début (true) ou en fin (false) de ligne

Retourne true si succès

Visibility	public
Return Type	bool

8.5.2. addTramway

Ajoute un tramway à la ligne

Visibility	public
-------------------	--------

8.5.3. expand

Étend une ligne

Le paramètre from la station depuis laquelle la ligne doit être étendue

Le paramètre to la station vers laquelle la ligne doit être étendue

Visibility	public
Return Type	bool

8.5.4. getMaxPos

Retourne la valeur de position maximale possible sur cette ligne

Visibility	public
Return Type	int

8.5.5. getMinPos

Retourne la valeur de position minimale possible sur cette ligne

Visibility	public
Return Type	int

8.5.6. getPixelsAt

Retourne les coordonnées sur l'écran correspondant à une position sur la ligne







Visibility	public
Return Type	Vector2

8.5.7. Line

Crée une nouvelle ligne entre deux stations

Visibility	public
-------------------	--------

8.6. Relations




Relationship	From	To
— unnamed	 Line	 Tramway
— unnamed	 Map	 Line
— unnamed	 Station	 Line

9. Tramway




9.1. Propriétés

Visibility	public
Abstract	false
Leaf	false
Root	false
Active	false

9.2. Attributes Summary

Name	Description
 capacity	Nombre maximal de personnes que peut accueillir ce tram
 lastVisitedStation	Station visitée avant la station courante Cette donnée permet de savoir dans quelle direction le tram se dirige
 linePos	Position sur la ligne, par tranches de pourcentages Par exemple, si le tram se trouve à mi-chemin entre la deuxième et la troisième station de la ligne, alors cette donnée vaut 250

9.3. Résumé des méthodes

Name	Description
 getLastStation	Retourne la station visitée avant la station courante
 goTo	
 Tramway	

9.4. Attributes

9.4.1. capacity

Nombre maximal de personnes que peut accueillir ce tram

Visibility	private
Type	int

9.4.2. lastVisitedStation

Station visitée avant la station courante

Cette donnée permet de savoir dans quelle direction le tram se dirige

Visibility	private
Type	Station

9.4.3. linePos

Position sur la ligne, par tranches de pourcentages

Par exemple, si le tram se trouve à mi-chemin entre la deuxième et la troisième station de la ligne, alors cette donnée vaut 250

Visibility	private
Type	double

9.5. Méthodes

9.5.1. getLastStation

Retourne la station visitée avant la station courante

Visibility	public
Return Type	Station



9.5.2. goTo

Visibility	public
Return Type	bool

9.5.3. Tramway

Visibility	public
-------------------	--------

9.6. Relations

Relationship	From	To
— unnamed	 Tramway	 People
 unnamed	 PlaceToBe	 Tramway
— unnamed	 Line	 Tramway



10. Player

Contient toutes les données du joueur


10.1. Propriétés

Visibility	public
Abstract	false
Leaf	false
Root	false
Active	false

10.2. Attributes Summary

Name	Description
 money	
 satisfaction	

10.3. Résumé des méthodes

Name	Description
 getMoney	
 getSatisfaction	
 setMoney	
 setSatisfaction	

10.4. Attributes

10.4.1. money

Visibility	private
Type	int

10.4.2. satisfaction

Visibility	private
Type	double

10.5. Méthodes

10.5.1. getMoney

Visibility	public
Return Type	int

10.5.2. getSatisfaction

Visibility	public
Return Type	double



10.5.3. setMoney

Visibility	public
Return Type	void

10.5.4. setSatisfaction

Visibility	public
Return Type	void

10.6. Relations

Relationship	From	To
— unnamed	 Game	 Player

11. Vector2



Classe qui représente un vecteur à deux valeurs.

Elle peut être utilisée pour représenter des coordonnées, des mouvements, ou encore calculer des distances.








11.1. Propriétés

Visibility	public
Abstract	false
Leaf	false
Root	false
Active	false

11.2. Attributes Summary

Name	Description
 x	
 y	

11.3. Résumé des méthodes

Name	Description
 add	Retourne la somme du vecteur courant et de celui passé en paramètre
 distanceBetween	
 getX	
 getY	
 setX	
 setY	
 sub	Retourne la soustraction du vecteur passé en paramètre au vecteur courant

11.4. Attributes

11.4.1. x

Visibility	private
Type	double

11.4.2. y

Visibility	private
Type	double

11.5. Méthodes

11.5.1. add

Retourne la somme du vecteur courant et de celui passé en paramètre

Visibility	public
Return Type	Vector2

11.5.2. distanceBetween

Visibility	public
Return Type	double

11.5.3. getX

Visibility	public
Return Type	double

11.5.4. getY

Visibility	public
Return Type	double

11.5.5. setX

Visibility	public
Return Type	void

11.5.6. setY



Visibility	public
Return Type	void

11.5.7. sub

Retourne la soustraction du vecteur passé en paramètre au vecteur courant

Visibility	public
Return Type	Vector2

11.6. Relations

Relationship	From	To
— unnamed	 Cell	 Vector2


12. Area

Représente un quartier d'un type donné (résidentiel, commercial ou d'affaires)







12.1. Propriétés

Visibility	public
Abstract	false
Leaf	false
Root	false
Active	false

12.2. Attributes Summary

Name	Description
 type	Type de zone

12.3. Résumé des méthodes

Name	Description
 addBuilding	Ajoute un bâtiment à la table de hachage. Cette méthode ne génère aucun bâtiment, elle l'ajoute simplement à la table pour simplifier l'accès.
 addCell	Ajoute une case de la grille à la zone
 generateBuilding	Demande à la zone de générer un nouveau bâtiment
 getDensity	Retourne la densité de la zone (rapport cases disponibles/cases occupées)
 removeCell	Retire une case de la grille de la zone
 sendToStation	Envoie chaque personne présente dans les maisons (House) de la zone dans la station accessible la moins congestionnée

12.4. Attributes

12.4.1. type

Type de zone

Visibility	private
Type	AreaTypes

12.5. Méthodes

12.5.1. addBuilding

Ajoute un bâtiment à la table de hachage.

Cette méthode ne génère aucun bâtiment, elle l'ajoute simplement à la table pour simplifier l'accès.

Visibility	private
-------------------	---------

12.5.2. addCell

Ajoute une case de la grille à la zone

Visibility	public
Return Type	bool

12.5.3. generateBuilding

Demande à la zone de générer un nouveau bâtiment

Visibility	public
Return Type	bool

12.5.4. getDensity

Retourne la densité de la zone (rapport cases disponibles/cases occupées)

Visibility	public
Return Type	double

12.5.5. removeCell

Retire une case de la grille de la zone

Visibility	public
Return Type	bool

12.5.6. sendToStation

Envoie chaque personne présente dans les maisons (House) de la zone dans la station accessible la moins congestionnée

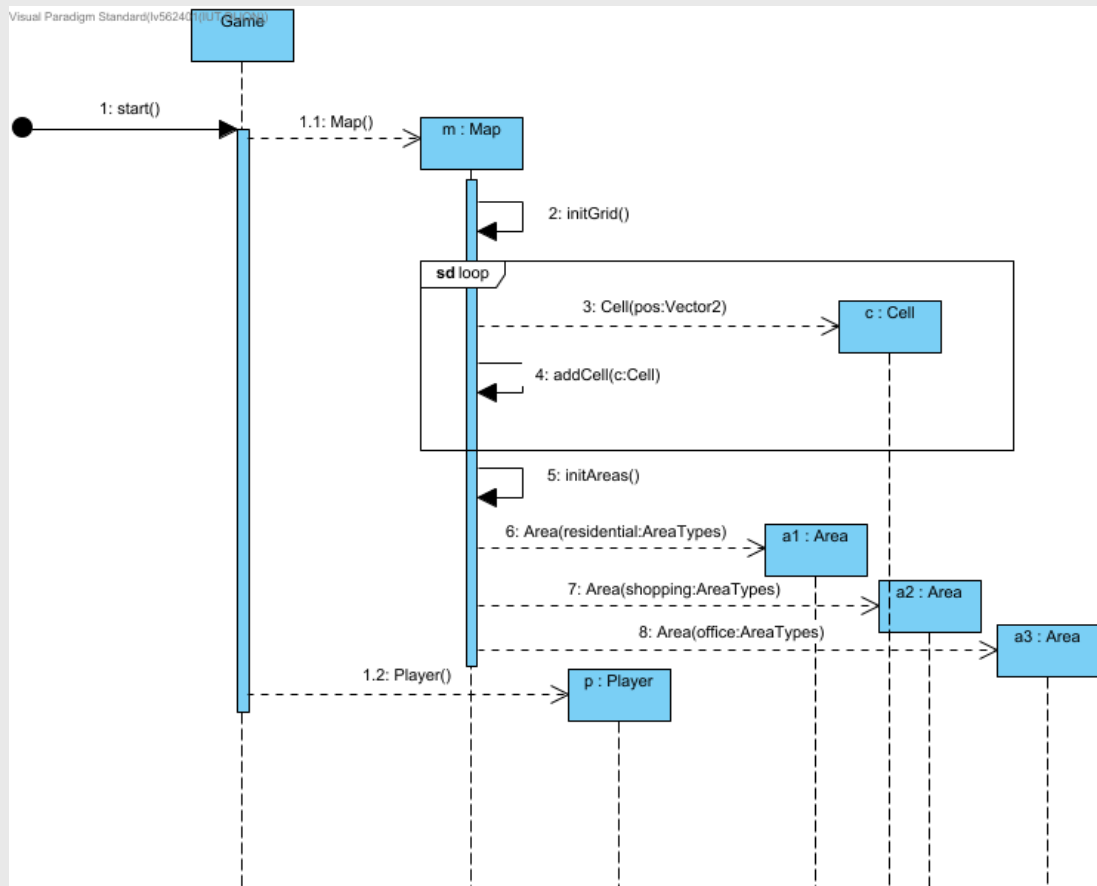
Visibility	public
-------------------	--------

12.6. Relations

Relationship	From	To
— unnamed	 Area	 Cell
— unnamed	 Area	 Building
— unnamed	 Map	 Area

Séquences

1. Lancer partie



2. Créer une ligne

