

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

Projet jeu de la vie

Cahier de conception

FICHE DE SUIVI DES AUTORISATIONS ET DIFFUSIONS

AUTORISATIONS PRESTATAIRE

	Fonction	Nom	Date	Visa
Auteur	Directeur de projet			
Validé par				
Approuvé par				

AUTORISATIONS CLIENT

	Fonction	Nom	Date	Visa
Approuvé par				

DIFFUSION INTERNE

Nom	Fonction	Action	Date	Nb exemplaire(s)

DIFFUSION EXTERNE

Nom	Fonction	Action	Date	Nb exemplaire(s)

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

Table des matières

Table des matières

Préambule	2
Introduction.....	3
1.1 Evolution.....	4
1.2 Outils utilisés.....	4
Terminologie.....	5
1.1 Abréviations	5
1.2 Définitions des termes employés.....	5
Description et analyse de l'environnement	6
1.1 Description des ressources matérielles	6
1.2 Interfaces	6
1.3 Organisation de l'espace de travail.....	6
Structure statique	7
1.1 Décomposition générale.....	7
1.2 Analyse statique.....	7
1.3 Classes.....	10
Aspects dynamiques	15
1.1 Liens entre les éléments.....	15
1.2 Description des tâches	15
1.3 Comportement du logiciel en erreur.....	17
1.4 Modes de fonctionnement	17

Préambule

Tout au long de ce document de spécifications, nous allons utiliser des règles pour une meilleure compréhension :

- Le nom des Personnage sera en *Italique*
- Le nom des classes sera en **Gras**.

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

Introduction

Ce cahier de conception vise à présenter la conception de notre jeu de la vie.
 Ce projet est réalisé durant la 4eme année du cycle d'ingénieur Informatique à Polytech.
 Ce projet a pour but de créer un jeu, une simulation de vie dans lesquels différents acteurs Interagissent.

Rappel des règles :

Le jeu se présentera dans une fenêtre. L'utilisateur se verra attribuer un espace prédéfini pour pouvoir organiser la manifestation. Il aura un nombre donné de *Manifestant* à placer sur l'espace de jeu. Une fois cela fait, le jeu se lancera, les *Manifestants* se développeront indépendamment. Au bout d'un certain seuil de *Manifestants*, des *CRS* apparaîtront de manière aléatoire sur la carte afin de contrer cette manifestation. Les *Manifestants* gagnent s'ils atteignent un certain nombre que nous avons spécifié dans le fichier paramètre, mais ils perdent s'ils retombent au même nombre de départ. Le jeu sera régi par des règles simples afin de pouvoir lui laisser une autonomie dans son fonctionnement.

Les règles sont celles-ci :

- > Un *Manifestants* devient *Neutre* s'il est entouré de personne *Neutre* sans ami *Manifestant*
- > Un *CRS* est neutralisé s'il est entouré d'au moins six *Manifestants*
- > Une personne *Neutre* devient un *Manifestant* s'il est entouré d'au moins deux *Manifestants*
- > Le jeu se finit si les *Manifestants* atteignent le nombre de base en début de partie ou s'ils atteignent une certaine valeur définie par paramètre (voir description du fichier param.ini définit dans les chapitres suivants)
- > Si un *Manifestant* a pour voisin deux *CRS* alors il est placé en garde à vue et la case devient vide
- > Seul les *CRS* et les *Neutres* peuvent se déplacer de manière aléatoire

Nous allons commencer par décrire les ressources matérielles et logicielles, puis vous présenter un diagramme d'exécution afin d'avoir une vue d'ensemble sur le jeu puis nous détaillerons le fonctionnement du jeu avec un exemple précis. Faut présenter succinctement la structure du document :

- Structure dynamique (synchronisation entre tâches, séquençement des traitements, passages entre les différents états ou mode de traitement, ...),
- Implémentation sur le matériel cible (allocation des ressources CPU mémoire bande passante, adresse des périphériques, couche basse de protocoles spécifiques, ...).

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

1.1 Evolution

Utilisation d'un fichier JSON permettant de stocker des informations tel que les paramètres du programme sur le nombre de manifestant au départ ou pour que la partie se finisse, mais aussi pour stocker les informations du plateau afin que l'utilisateur puisse reprendre sa partie

1.2 Outils utilisés

Pour la réalisation de ce projet, nous utiliserons principale le langage JAVA pour la partie programmation, GITHUB pour stocker le projet afin qu'il soit accessible par tous et pour finir de Word ou Google Docs pour la partie administrative afin de remplir les documents techniques liés au projet.

Terminologie

1.1 Abréviations

UML Unified Modeling Language

1.2 Définitions des termes employés

Attribut	Un attribut est une information caractéristique mémorisée par un objet.
Cas d'utilisation	Cas d'utilisation du système, par extension il représente également la technique de modélisation mise en œuvre dans UML (use case).
Catégorie	Une catégorie consiste en un regroupement logique de classes à forte cohérence interne et faible couplage externe, associée au concept UML de package. Ce concept permet une présentation plus synthétique du diagramme des classes d'un système réel.
Classe	Une classe définit un ensemble d'objets similaires potentiels. Elle fournit le modèle de la structure et les possibilités de chaque objet.
Objet	Un objet est une instance d'une classe, c'est une entité informatique unique possédant ses propres attributs et opérations
Opération ou méthode	Une opération est un traitement spécifique qu'un objet est en charge de fournir.
Tâche	Une tâche représente un élément manipulé par le système et ordonnançable de manière individuelle. cela peut représenter un processus d'un système Unix, une tâche d'un moniteur temps-réel, un thread d'une application.

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

Description et analyse de l'environnement

Nous utiliserons Eclipse IDE qui est un environnement de développement intégré libre extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java, et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

1.1 Description des ressources matérielles

Nous utiliserons nos Ordinateur personnels ou les postes de l'école.
Notre jeu doit pouvoir être utilisé sans ralentissement sur tous ces ordinateurs

1.2 Interfaces

1.2.1 Description et définition des interfaces

- Nous allons utiliser la Bibliothèque Graphique SWING de Java afin d'afficher les différentes informations du jeu : que ce soit pendant le déroulement, pour indiquer des messages d'informations aux joueurs ou pour les différentes options/ possibilités apporté au joueur.
- Le joueur utilisera sa souris pour interagir avec le jeu.

1.3 Organisation de l'espace de travail

Nous travaillerons principalement dans les salles de TP de l'écoles ou à distance en utilisant un logiciel de discussion type Skype ou Discord.

Structure statique

1.1 Décomposition générale

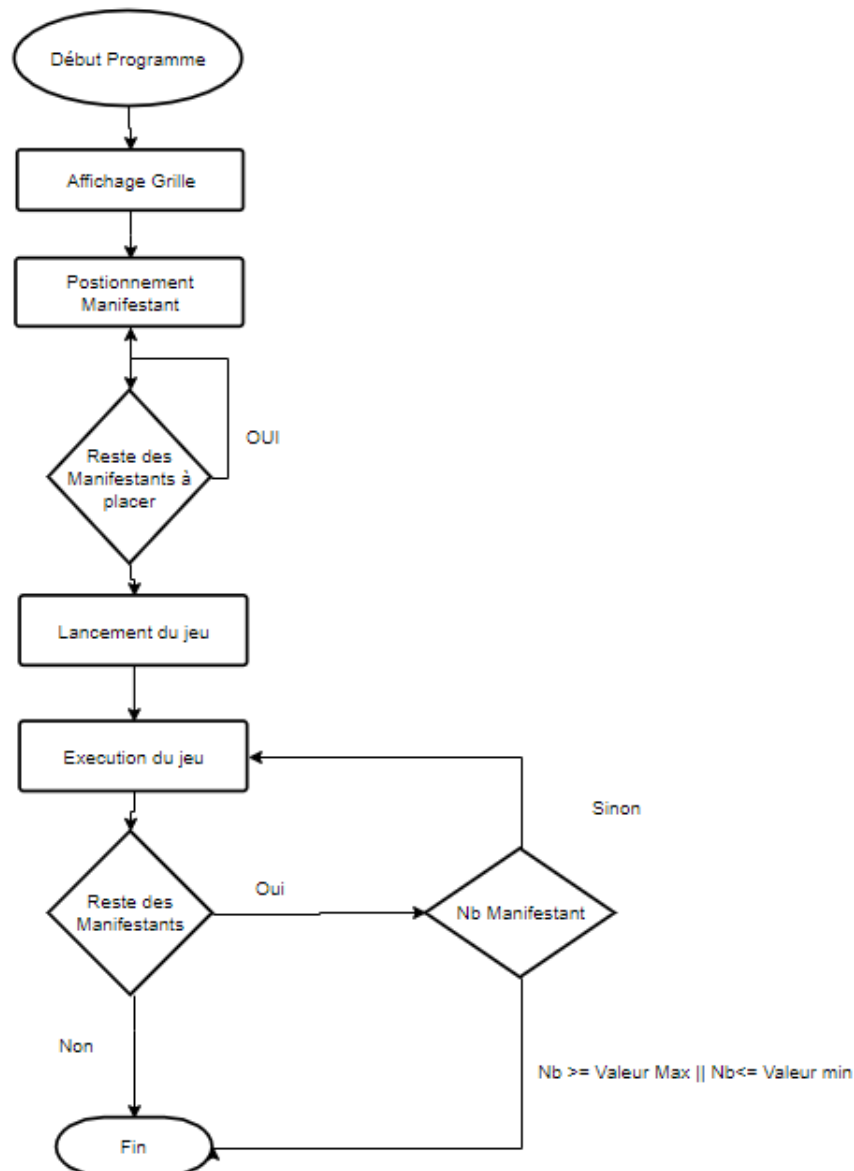
La décomposition générale doit présenter les grandes classes ou les catégories qui sont par la suite développées.

Les catégories regroupent un ensemble de classes ayant le même centre d'intérêt et permettent de clarifier la présentation générale pour un projet de taille importante.

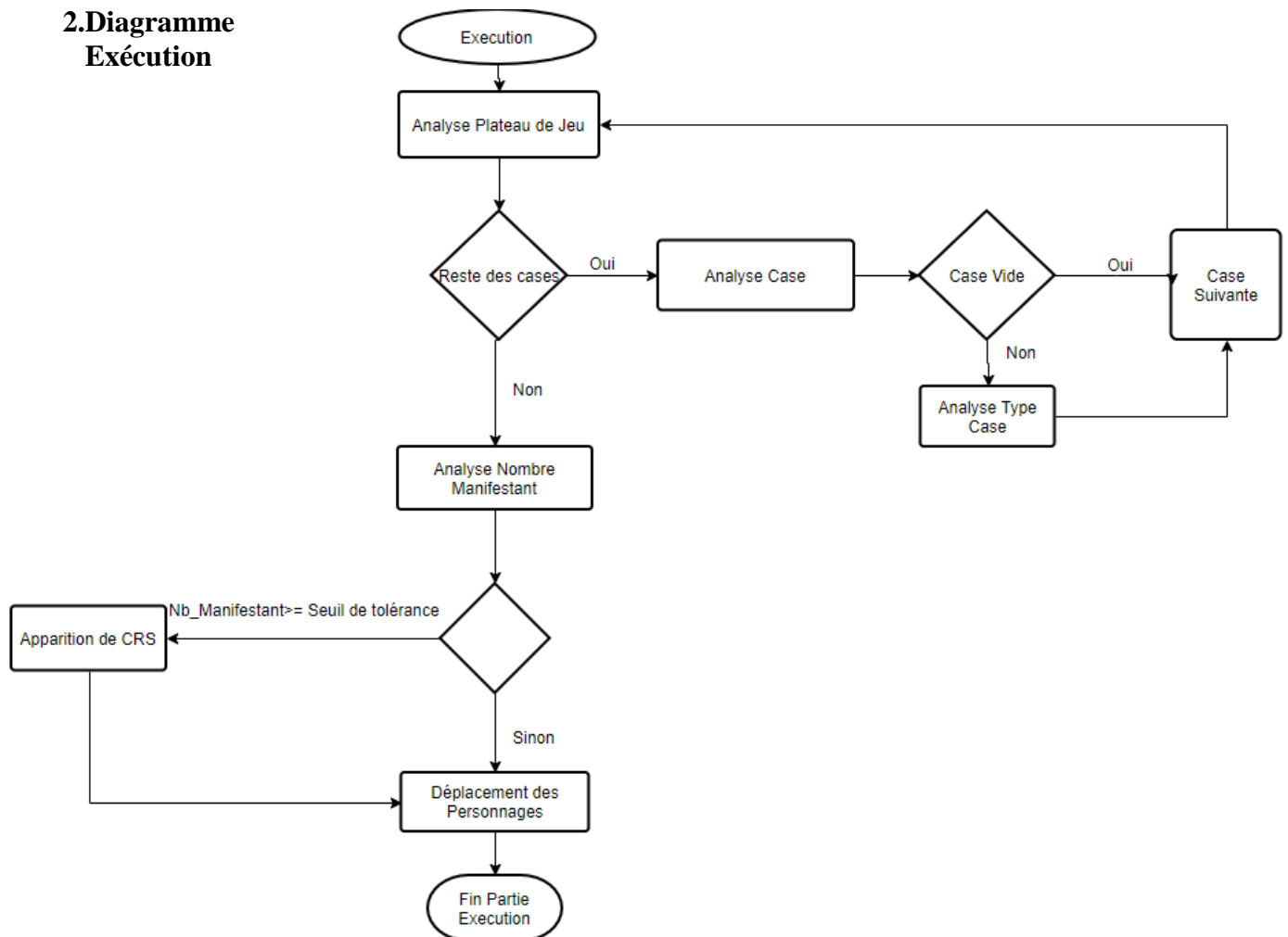
1.2 Analyse statique

Nous avons développé le fonctionnement du jeu de façon plus imagée pour faciliter la compréhension. Nous avons simplifié les parties les plus complexes en les expliquant dans un nouveau diagramme afin de ne pas alourdir le diagramme principal.

1.Diagramme Générale

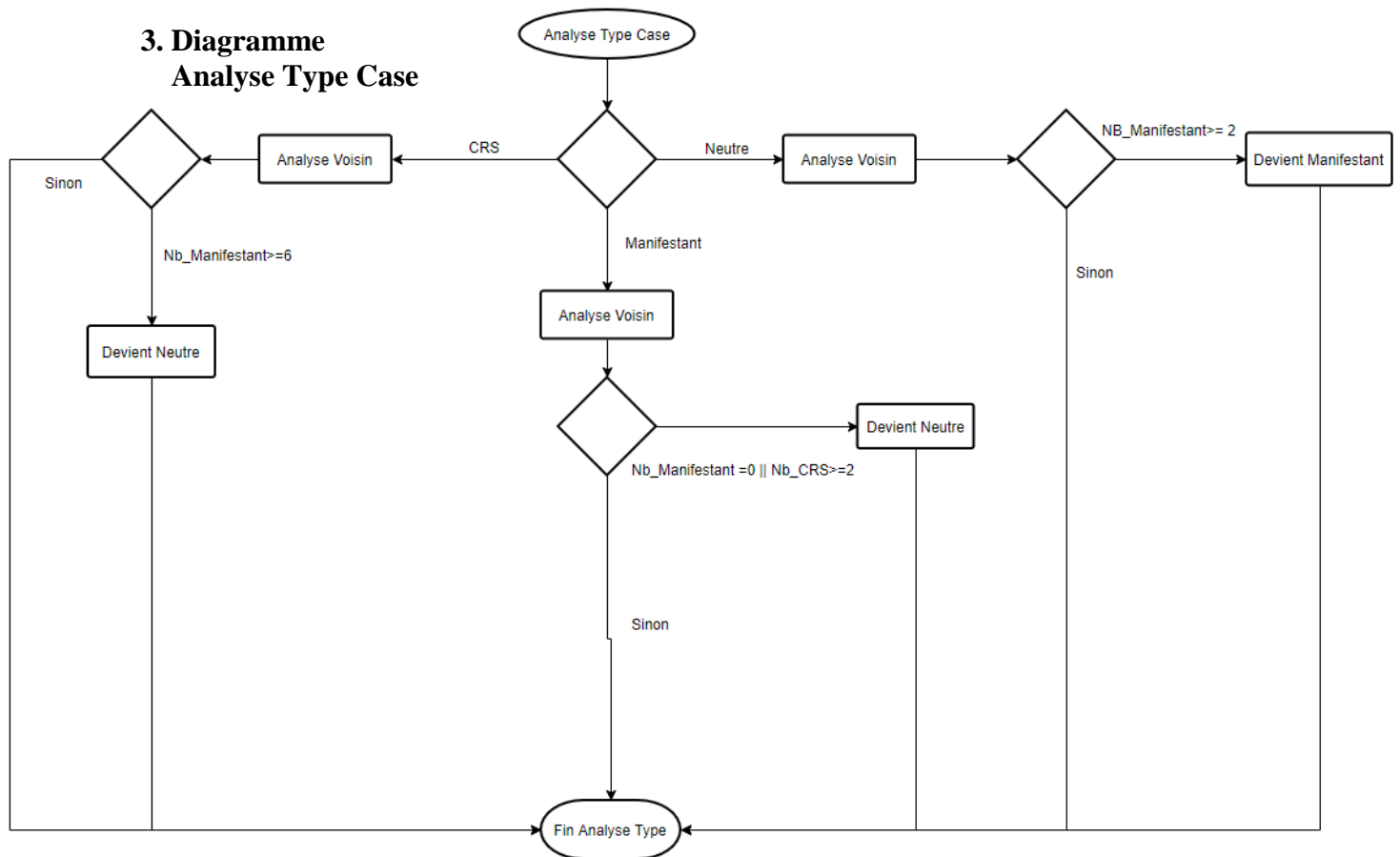


2.Diagramme Exécution



Nous explicitons le fonctionnement de l'exécution du programme.

3. Diagramme Analyse Type Case



Cette partie explique comment fonctionne l'analyse d'une case par rapport à ses voisins et l'action qui lui est alors appliquée.

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

1.3 Classes

1.3.1 Classe Personne

Définition

Personne est une classe abstraite qui est mère des différentes entités du jeu, chaque personne peut se déplacer, et se transformer.

Attributs

Nous avons six attributs : 3 compteurs pour connaître le nombre de voisin *CRS*, *Manifestant*, *Neutre* de notre Objet, un statut pour déterminer si la personne est *CRS Manifestante* ou *Neutre*, une **Grille** afin d'associer une personne à un plateau de jeu et enfin une **Case** pour lui attribuer ses coordonnées sur le plateau.

Méthodes

La classe aura des méthodes abstraites qu'hériteront les classes filles tel que la méthode transform mais aussi des méthodes définies comme déplacement et analyse qui comptera le nombre de voisin autour d'elles.

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

1.3.2 Classe Manifestant

Définitions

Les *Manifestants* sont ceux qui manifestent et essayent de convertir un maximum de Personnes à leur cause.

Attributs

Elle héritera de la classe **Personne** donc elle aura les attributs pour obtenir ces coordonnées et tout ce qu'on a vu précédemment.

Méthode

Elle implémente la méthode transformation de la classe mère **Personne** et modifie la fonction déplacement car les *Manifestants* restent là où ils sont (Par exemple comme la manifestation à Barcelone avec les indignés).

1.3.3 Classe Neutre

Définitions

Les Personnes **Neutre** sont ceux qui restent impassible et qui se promènent durant la manifestation.

Attributs

Elle héritera de la classe **Personne** donc elle aura les attributs pour obtenir ces coordonnées et tout ce qu'on a vu précédemment.

Méthode

Elle implémente la méthode transformation de la classe mère **Personne**.

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

1.3.4 Classe CRS

Définition

Les *CRS* sont les entités qui chassent les manifestants. Un *CRS* peut se déplacer et encercler un manifestant

La classe hérite de **Personne**.

La fonction déplacement forces les *CRS* à se diriger vers les *Manifestants* que le groupe de *CRS* peut interpeller.

Attributs

Elle hérite des attributs de **Personne** vu précédemment.

Méthodes

Le *CRS* implémente la méthode transform de **Personne**.

1.3.5 Classe Canevas

Définition

Définis les différents affichages sur la grille en récupérant les informations contenues dans la liste de **Personne** de la **Grille**. Elle implémente **Observer** afin d'être tenu au courant après chaque modification de la *Grille* afin de changer l'affichage.

Attributs

Elle aura 5 attributs : 3 compteurs pour connaître le nombre de *CRS*, *Manifestant*, *Neutre* sur le terrain de jeu, une grille qui lui est associé, et une matrice de bouton pour le placement en début de partie.

Méthode :

Les Méthodes principales seront : affichage de la grille avec les différentes **Personne** en colorant les boutons, placement des *Manifestants*, et changer l'affichage de la grille après chaque modification.

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

1.3.6 Classe Grille

Définition

Grille définit une matrice de *Case* dans lesquelles seront présentes, ou non les entités.

Attributs

Une matrice contenant des *Cases* fera partie du tableau afin de savoir positionner les **Personnes**. Dans chaque *Cases* sera défini un type d'objet : *CRS*, *Manifestant*, *Neutre*, vide

Il aura aussi un nombre minimum de *Manifestant*, qui servira à connaître le nombre de manifestant à placer et le seuil à ne pas atteindre par la suite. Il aura enfin pour finir un nombre maximum de *Manifestant* afin que la partie puisse se finir.

Il aura enfin un nombre de ligne et de colonne pour construire la matrice et aussi un ArrayList d'Observer afin que **Canvas** puisse contrôler s'il y a des modifications.

Méthodes

Elle a comme méthode : des getters et des setters pour obtenir ou modifier les *Cases* de la matrice, un ajout d'observer et une méthode pour notifier à l'Observer qu'il y a eu une modification.

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

1.3.7 Classe Case

Définition

Case permet de savoir rapidement si la case est vide ou non et qu'elle entité s'y trouve. Si la *Case* n'est pas vide alors une personne lui est rattachée.

Attributs

Elle a 4 attributs : 2 attributs qui correspondent à ses coordonnées, un attribut Statut qui permet de savoir rapidement ce qui se trouve dans la *Case* et une *Personne* si la *Case* n'est pas vide.

Méthodes

Les seules méthodes employées ici sont des Getter & Setters pour modifier et obtenir les informations de ses attributs.

1.3.8 Classe Fenetre

Définition

La **Fenetre** permet d'afficher la *Grille* de jeu dans une fenêtre Windows ainsi que d'associer à la fenêtre des boutons pour faciliter l'utilisation.

Attributs

Elle a 9 attributs : 1 MenuBar dans lequel est contenu 2 Menu dans lesquelles sont contenues 4 MenuItem. Cela permet d'emboîter les menus afin d'optimiser la place et faciliter la compréhension. Les MenuItem servent à indiquer à l'utilisateur le nombre de *Manifestant* à placer, le nombre qui sont déjà placé, une aide et un contact si l'utilisateur les sélectionne. Enfin la **Fenetre** est associée à un **Canvas** et une **Grille**.

Méthodes

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

Les méthodes utilisées ici servent à réaliser les différentes actions mentionnées dans les attributs sur les MenuItem.

1.3.9 Classe Main

Définition

La classe **Main** ne sert uniquement pour lancer le programme, c'est la classe **Grille** et **Canvas** qui s'occupent du bon déroulement du jeu.

Aspects dynamiques

1.1 Liens entre les éléments

Le *Plateau* manipule les différentes entités présentes en les faisant se mouvoir et en les changeant en fonction des situations expliqués dans le cahier de Spécifications en stockant toutes ces informations dans son tableau. Ensuite cela permettra au *Canevas* de dessiner la grille en fonction du positionnement de ces éléments.

1.2 Description des tâches

Le plateau doit gérer le changement de Personne, si un Neutre devient Manifestant et toutes les autres possibilités expliquées dans notre schéma au début du rapport, diagramme numéro 3.

Exemple :

CRS 1.1	... 1.2	... 1.3
... 2.1	... 2.2	Manifestant 2.3

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

...	Neutre	Manifestant
3.1	3.2	3.3

Le programme va tout d'abord parcourir le tableau et analyser chaque case :

Parcours tableau :

On parcourt l'intégralité du tableau avec une Boucle For.

Si la case n'est pas vide, on analyse ses voisins.

Sinon on passe à la case suivante.

Analyse Voisin :

Nous devons d'abord repérer la case dans l'espace, c'est à dire déterminer qu'elles sont les cases qui l'entourent. Nous savons qu'au maximum une case a 8 voisins, 3 au-dessus, 2 à côtés et 3 en dessous. Cela forme un petit carré de dimension 3x3 autour de la case analysée.

	Case Analysée	

Nous créons un compteur qui aura comme valeur le nombre de case *Manifestant*. Ensuite nous parcourons les cases entourant notre case à tester. Pour faciliter la compréhension, nous donnons comme coordonnées

INFO4A Yves Jehanno Client M.Mavromatis Projet jeu de la vie	Projet Conception générale	Date : 08/02/19
-----------------------------------------------------------------------	-------------------------------	-----------------

à la case à tester X et Y. Nous allons parcourir les case X-1,Y-1 jusqu'à X+1,Y+1 en excluant notre case X Y. Bien sûr nous devons vérifier que les cases testées sont dans le *Plateau*.

Si la case contient un *Manifestant*, alors on incrémente le compteur, sinon on passe à la suivante.

Enfin, nous analysons la contenance de notre case à analyser. Comme vu dans le diagramme 3, le nombre de manifestant détermine l'action à réaliser.

S'il est supérieur ou égal à 8 et que la case est un *CRS*, alors la case devient *Neutre*. S'il est supérieur à 2 et que la case est *Neutre* alors elle devient un *Manifestant*. S'il est égal à 0 et que la case est un manifestant alors elle devient *Neutre*.

Pour finir, si nous devons modifier la case (par rapport aux règles établis ci-dessus), nous supprimons l'élément au coordonnée X, Y et nous remplaçons l'élément par la nouvelle entité.

Pseudo-code:

Pour (i=X-1, i < X+2, i++):

Pour(j=Y-1, j < Y+2, j++):

Si (j>0 et j+2<max_tableau et i>0 et i+2< max tableau et i!=X et j!=Y):

Si (tableau[i][j] est Manifestant):

voisin_manifestant + 1:

Dans notre exemple, *Neutre* deviendra *Manifestant* si nous déroulons l'algorithme décrit ci-dessus.

1.3 Comportement du logiciel en erreur

Des messages d'erreurs apparaîtront en cas de besoin ainsi que l'arrêt de programme pour éviter qu'il boucle à l'infini. Tout cela se passe dans la classe Plateau.

1.4 Modes de fonctionnement

Mode de placement :

Le joueur place les manifestants sur la grille avec sa souris. Il peut en placer un nombre limité.

Mode jeu ;

le jeu se lance et le joueur n'est plus que spectateur.