RAPPORT DU TP DE PROGRAMMATION ORIENTÉE OBJET

Introduction

Ce projet Java vise à créer une application graphique pour simuler des systèmes multi-agents, en mettant en lumière trois automates cellulaires (Jeu de la Vie de Conway, jeu de l'immigration, modèle de ségrégation de Schelling) et le modèle de Boids pour le mouvement d'essaims. Il offre une opportunité d'explorer les principes de la programmation orientée objet, intégrant des concepts tels que l'encapsulation, la délégation, l'héritage, l'abstraction, et l'utilisation des collections Java. La conception orientée objet, axée sur la flexibilité des classes, favorise la généralisation du code entre les différents systèmes multi-agents étudiés, établissant ainsi une plateforme expérimentale pour comprendre les dynamiques émergentes et le comportement collectif des entités autonomes au sein d'un système multi-agents.

2. Descriptions des Systèmes d'automates cellulaires

- **Le Jeu de la Vie de Conway :**

Le Jeu de la Vie de Conway est un automate cellulaire célèbre caractérisé par des règles simples. Chaque cellule d'une grille peut être dans l'un des deux états : vivante ou morte. Une cellule morte entourée exactement de trois voisines vivantes devient vivante (elle naît), tandis qu'une cellule vivante avec deux ou trois voisines vivantes reste en vie, sinon elle meurt. Dans notre projet, les bords de la grille sont considérés comme circulaires, et les règles sont appliquées simultanément à toutes les cellules pour passer d'un état à l'état suivant. Ce modèle, proposé par le mathématicien John Conway, offre une représentation captivante des évolutions complexes émergentes à partir de règles élémentaires.

- **Le Jeu de l'Immigration :**

Le Jeu de l'Immigration constitue une variation légère du Jeu de la Vie, distinguée par sa gestion plus flexible des états des cellules. Chaque cellule peut adopter l'un des n états possibles, et elle passe à l'état k + 1 modulo (n) si elle a trois voisines ou plus dans cet état k + 1. Ces règles, bien que simples, engendrent des dynamiques d'évolution fascinantes au sein de la grille. Cette variante apporte une dimension supplémentaire à la modélisation des automates cellulaires, explorant des comportements émergents au-delà de la dichotomie classique vie/mort de l'automate précédent.

- **Le Modèle de Ségrégation de Schelling :**

Le Modèle de Ségrégation de Schelling propose une approche réaliste d'un automate cellulaire, visant à simuler la dynamique ségrégationniste dans une population. Chaque cellule représente une habitation (soit vacante, soit occupée par une famille de couleur spécifique). Si une famille a plus de K voisines de couleur différente (K étant un seuil paramétrable), elle déménage à la recherche d'une habitation vacante. Ce modèle, initié par l'économiste Thomas Schelling, démontre comment de légères préférences de voisinage peuvent conduire à une ségrégation significative, offrant ainsi des perspectives sur les mécanismes sous-jacents aux phénomènes sociaux complexes.

3. Détails de l'Implémentation des Systèmes d'automates cellulaires

L'implémentation de cette partie du code nous a permis d'utiliser de mieux comprendre des principes de programmation orientée objet vues en cours tels que :

- Classe et Interface

- Encapsulation
- Délégation
- Héritage
- Abstraction
- Liaison statique et Liaison dynamique
- Utilisation des collections Java (HashMap)
- Tableaux bidimensionnels

Le diagramme de classe associé à cette première partie est présenté dans le fichier attaché “ figure1.png”.

4. Description du Modèle de Boids

Ce modèle est un simulateur graphique d’un déplacement libre, dans l’espace 2D d’agents en essaims(groupes d’agents capables de s’auto-organiser) appelés boids . Le comportement de chaque boid est régis par un ensemble de règles et éventuellement d'attributs propres à la population à laquelle il appartient. Par ailleurs, il existe trois règles de base communes à toutes les populations de boids à savoir: la règle de Cohésion(un agent se dirige vers la position moyenne de ses voisins), la règle d’Alignement(un agent tend à se déplacer dans la même direction que ses voisins) et la règle de Séparation(les agents trop proches se repoussent, pour éviter les collisions).
Ainsi Pour différencier les populations de boids, nous avons fait varier les differents attributs tels que le seuil de voisinage(), le seuil de séparation(), le poids de la force de cohésion et aussi ajouté certaines nouvelles règles.

5. Détails de l'Implémentation du Modèle de Boids avec le gestionnaire d'événements

Un gestionnaire d'événement a été impliqué pour gérer le déplacement des agents dans les simulateurs. Le diagramme de classe pour cette partie est présenté dans le fichier attaché “ figure2.png”.

6. Généralisation du Code

- **Pour les Systèmes d'Automates Cellulaires :**

Comme le montre la figure précédente, nous avons utilisé les principes d'héritage et de délégation pour réutiliser au mieux notre code. Par exemple, les classes ImmigraitonSimulator et ShellingSimulator ne contiennent que leur constructeur car leur comportement est exactement celui de leur classe mère ConwaySimulator. De plus, nous avons implémenté la Grille comme étant un ensemble de Cellules, ce qui nous a permis de réutiliser cette grille dans toutes les Classes de Simulation, en changeant juste le comportement de ses cellules. Cette conception permettra par la suite une réutilisabilité plus efficace de notre code et une maintenance plus facile.

- **Pour le modèle de Boids et le gestionnaire d'événement :**

Nous avons une classe abstraite Event qui est héritée par notre classe EventBoids (de même pour les classes EventBalls et EventGrille) pour gérer les simulateurs d'agents et à chaque simulateur est associé un gestionnaire d'événement.

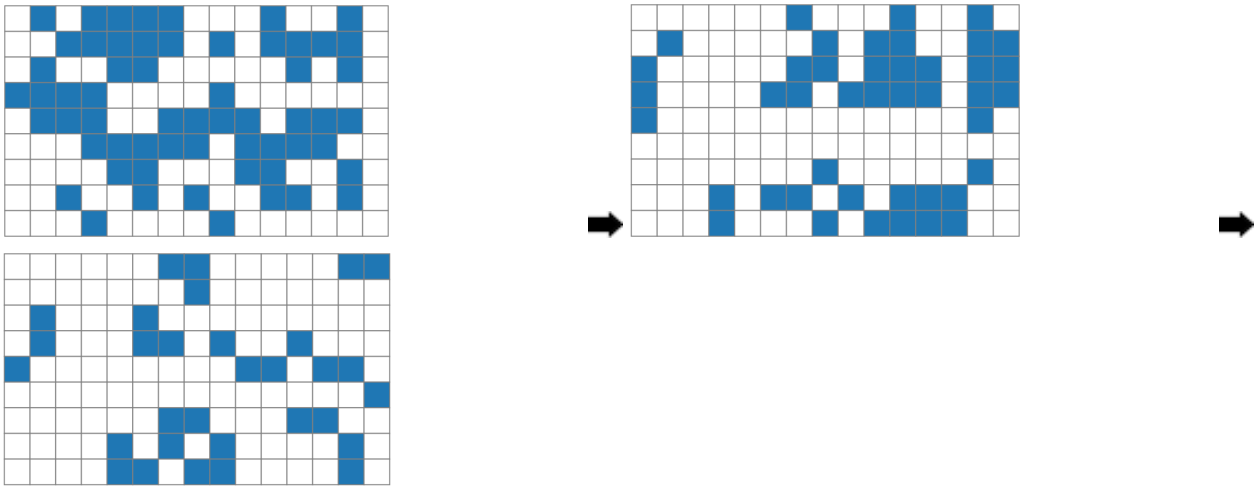
7. Défis et Solutions

Dans ce TP, les principales difficultés rencontrées étaient dues au passage des agents (grille ou boids) d'un état à l'état suivant. En particulier sur la grille, nous avons pour l'identification des voisins des cellules qui se trouvaient sur les bords de la grille et sur les boids, la considération du voisinage d'un boid en fonction de son orientation . Mais grâce à la considération de la grille comme étant circulaire (tel que demandé dans l'énoncé), et en ajoutant un paramètre quant à l'angle de vision du boid, ces problèmes ont pu être résolus. Une autre difficulté que nous avons rencontrée était d'adapter notre grille (son nombre de cellules) à la taille de la fenêtre. Pour résoudre ce problème, nous avons fait en sorte que le nombre de cellules de la grille soit égal à la taille de la fenêtre divisé par la taille d'une cellule.

8. Tests et Résultats

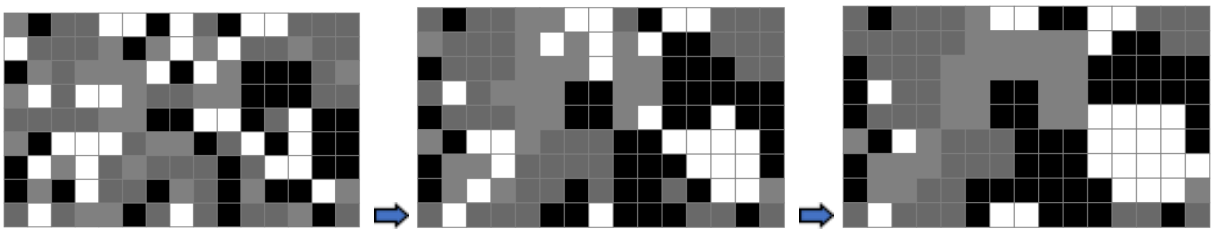
- **Pour le Jeu de la vie de Conway**

Partant d'une grille générée de façon aléatoire, nous avons lancé à plusieurs repises le Simulateur et l'état des cellules change jusqu'à ce qu'on atteigne un état stationnaire. Voici le résultat après deux itérations d'une simulation du jeu de la vie :



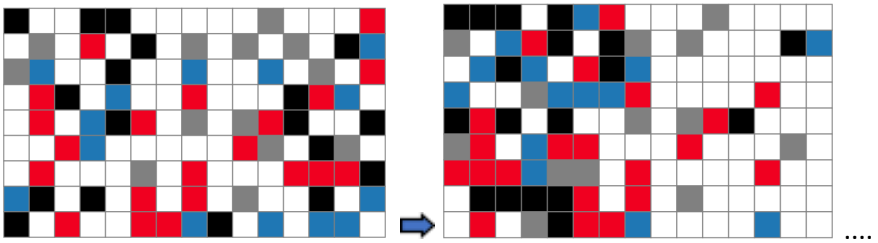
- **Pour le jeu de l'immigration**

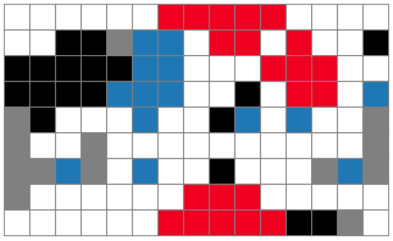
Partant également d'une grille générée de façon aléatoire, nous avons lancé à plusieurs reprises le Simulateur et l'état des cellules change en fonction des règles établis. Voici le résultat après deux itérations d'une simulation du jeu de l'immigration :



- **Pour le modèle de Schelling**

Partant d'une grille générée de façon aléatoire, nous avons fait varier le parametre K (le seuil) ainsi que le nombre de couleur pour voir à partir de quel seuil K est-ce qu'on obtient une ségrégation : c'est à partir d'un seuil **K = 5**. (On constate que lorsque le seuil K est fixé à une valeur <= 5, on atteint une ségrégation totale).
Voici le résultat après plusieurs itérations d'une simulation du Modèle de Schelling avec un seuil K = 5 et 04 couleurs (on aboutit à une ségrégation totale) :


....



10. Conclusion

En somme, la programmation orientée objet a permis une implémentation plus rapide et efficace et ces systèmes Multi agent de part ses principes de Classe et Interface, d'encapsulation, de délégation, d'héritage, d'abstraction. Mais aussi grâce aux structures de données qu'elle offre.