

Projet de simulations graphiques en Java

Ce projet contient plusieurs simulations graphiques utilisant la bibliothèque GUISimulator. Chaque simulation est autonome et affiche des agents ou cellules dans une fenêtre graphique.

Simulations incluses

1. **Invader**

Mini-jeu graphique représentant un Invader. L'Invader est dessiné avec des rectangles formant sa silhouette et suit un plan de déplacement prédéfini. La simulation peut être avancée étape par étape ou réinitialisée à l'état initial.

2. **Balle avec gravité**

Simulation graphique d'une balle soumise à la gravité et capable de rebondir sur le sol et les murs. La balle est représentée dans la fenêtre GUISimulator et sa trajectoire est mise à jour en continu selon les lois physiques simulées.

3. **Game of Life**

Automate cellulaire classique de Conway. Les cellules évoluent selon les règles du jeu de la vie. La fenêtre graphique permet de visualiser l'état de chaque cellule. Le plateau peut être réinitialisé à tout moment.

4. **Variante de l'immigration**

Variante de Game of Life avec plusieurs états de cellules. Les cellules sont affichées avec un dégradé de couleur selon leur état. Le plateau évolue selon la règle d'immigration et peut être réinitialisé à son état initial.

5. **Modèle de Schelling**

Modèle de ségrégation. Les cellules représentent différents groupes et se déplacent si leur seuil de satisfaction n'est pas atteint. Chaque groupe est représenté par une couleur différente, et la simulation évolue étape par étape.

6. **Swarm Simulation (Boids et prédateurs)**

Simulation de boids et de prédateurs. Les agents se déplacent selon les règles de cohésion, séparation, alignement et évitement des murs. Les prédateurs poursuivent les proies (sardines). Les positions et directions sont calculées à chaque étape et appliquées via un gestionnaire d'événements pour ordonner les mises à jour.

Structure du projet

- `src/` : code source Java, organisé par package (cellularSim, SwarmSim, etc.).
- `bin/` : fichiers compilés.
- `lib/` : bibliothèque graphique `lib/gui.jar`.
- `Makefile` : compilation et exécution des simulations.

Utilisation via le Makefile

- **Compiler tout le projet :**

```
$ make build
```

- **Exécuter une simulation spécifique :**

```
$ make runGameOfLife
```

```
$ make runGameOfImmigration
```

```
$ make runSegregationSim
```

```
$ make runTestGravityBall
```

```
$ make runInvader
```

```
$ make runSwarm
```

- **Nettoyer les fichiers compilés :**

```
$ make clean
```

Utilisation de la fenêtre graphique

Toutes les simulations s'affichent dans une fenêtre GUI Simulator. Cette fenêtre inclut plusieurs boutons et réglages pour contrôler la simulation :

Boutons :

- Lecture : lance la simulation en continu.
- Suivant : avance la simulation d'un pas unique.
- Début : réinitialise la simulation à son état initial.
- Quitter : ferme la fenêtre et arrête le programme.

Réglages disponibles :

- Temps entre deux affichages : définit l'intervalle de temps (en millisecondes) entre chaque rafraîchissement graphique lorsque la simulation est en lecture continue.
- Nombre de pas simulés entre affichages : permet de calculer plusieurs étapes de la simulation avant de mettre à jour l'affichage, utile pour accélérer les simulations longues.