

# Automate cellulaire

## Description de l'application

Notre application permet de jouer avec jusqu'à 5 joueurs sur une même machine à combattre avec des automates cellulaires.

## Mode d'emploi

Déroulement d'une partie :

- Dans un premier temps, il faut paramétrer la partie, c'est à dire donner les caractéristiques souhaitées, ou laisser celles par défaut.
- Chaque joueur choisit son pseudonyme et son automate, parmi les 5 actuellement disponibles.
- Ensuite, ils sont invités à placer leurs cellules initiales sur la grille de jeu, chacun leur tour.
- Le gagnant est celui dont l'automate parviendra à survivre le plus longtemps et les points sont déterminés par le nombre de cases possédées par ce joueur.

## Patrons de conception

Voici une liste des patrons de conception utilisés :

- Le patron MVC, autour duquel s'articule notre application. Il permet la séparation des classes selon leur rôle et évitant les inter-relations rapidement chaotiques.
- Le patron Stratégie, utilisé à deux reprises, pour les extensions et pour les règles.

Il permet d'obtenir des comportements radicalement différents tout en les masquant derrière une unique interface comme des classes.

- Le patron Factory, utilisé, lui aussi, à deux reprises pour les extensions et les règles.

Il permet la création d'objets implémentant une interface sans avoir à manipuler leurs constructeurs.

## Étendre l'application

Voici comment étendre notre application pour y ajouter de la matière :

- **Ajouter de nouveaux automates (règles) :**

Pour cela, il vous faut modifier dans un premier temps le fichier **"RuleFactory.java"** situé dans le package "control" en ajoutant au switch le nom de votre nouvelle règle en String et en le créant avec une des classes à votre disposition pour ce faire, "Unistate" est le meilleur compromis entre modulabilité et facilité d'usage. Ensuite, il vous faudra ajouter cette nouvelle règle dans **"RuleManager.java"** également situé dans "control", et ce dans son constructeur.

Vous pouvez maintenant jouer avec votre nouvelle règle !

- **Ajouter des méthodes d'extensions :**

Tout d'abord, il vous faudra ajouter la possibilité de créer l'extension, ce via une nouvelle classe implémentant **"Extension.java"**, et ce dans "model".

Alors on ajoute cette nouvelle classe au panel de classes de **"ExtensionFactory.java"** dans "control" pour pouvoir finalement l'afficher dans **"SetupGUI.java"**, celui-ci situé dans "view".

- **Tout type de voisinage peut déjà être créé via "Neighborhood.java", utilisé dans les règles actuelles.**

- **Ajouter un état aux automates :**

Le fichier **"State.java"**, dans "model", sera celui dans lequel vous devrez ajouter le nouvel état.

Il sera immédiatement présent mais jamais utilisé !

Pour palier à ce problème, modifiez les règles pour l'inclure (dans le paramètre "results" lors de conditions optimales ou selon l'âge, qui est déjà implémenté !).