

Rapport TP1 : SMA - Billes

François LEPAN - Alexis LINKE

17 septembre 2013

1 Choix

Nous avons fait le choix de partir sur un systeme de deplacement sur une grille case par case. Deux agents ne peuvent se retrouver sur une meme case.

Les couleurs des agents sont fixe lors du placement de ceux-ci dans l'environnement.

2 UML + explication

=====UML=====

L'organisation est la suivante :

Le design pattern utilis  pour r aliser ce programme est le MVC.

il y a 3 packages : core, view et model. Core contient la classe principale **Simulation** qui execute le programme. Ensuite view contient les classes **BallsPanel** qui represente le panel sur lequel on dessine les billes et **EnvironmentRepresentation** qui est la vue du mvc ainsi que la fenetre dans laquelle on ajoute le panel **BallsPanel**. Et enfin on a le package model qui lui contient toutes les classes necessaire aux calculs de collisions ainsi que de position tout au long de l'execution. La classe Agent contient les donnees n cessaire pour situer et identifier un agent au sein de l'environnement. La classe **Environment** contient une grille d'Agent et poss de une m thode d'allocation de place sur cette grille qui est utilis  lors de l'ajout d'un agent dans la classe **MultiAgentSystem**. Cette classe est le model. Elle contient une liste d'Agent ainsi que l'**Environment** dans lequel les agents se meuvent et la m thodes run qui   chaque tour donne la parole   chaque agents de fa on  quitable.

3 Compilation + fonctionnement

Compilation

Se mettre dans le dossier src → `javac core/Simulation.java`

Execution

Ne pas oublier de compiler le dossier src

`java core.Simulation <taille> <nb agent> <nb tour> <delai entre chaque tour>`

Si on rentre un nombre de tour = -1 alors c'est infini

Exemples

```
java core.Simulation 100 50 -1 5  
java core.Simulation 10 5 100 5  
java core.Simulation 50 40 -1 5
```