S1.01 : Implémentation d'un besoin client
Jeu de Marienbad

L. Naert – T. Ferragut – E. Lemonnier

7 octobre 2024

Jeu de Marienbad

Le jeu de Marienbad est une variante du jeu de Nim (le jeu des allumettes de Fort Boyard) rendue célèbre en 1961 par un film d'Alain Resnais et d'Alain Robbe-Grillet "L'Année dernière à Marienbad" (Source Wikipédia).

Dans ce jeu, il y a quatre rangées, avec respectivement 1, 3, 5, 7 allumettes. À chaque tour, le joueur prend le nombre d'allumettes qu'il veut dans une même rangée (au moins une allumette). **Le joueur qui prend la dernière allumette a gagné**.

Objectifs de la SAÉ

Rappel du programme national : l'objectif de S1.01 est de développer une application informatique simple. Les apprentissages critiques concernés sont :

- **AC 1** Implémenter des conceptions simples
- **AC 2** Élaborer des conceptions simples
- **AC 3** Faire des essais et évaluer leurs résultats en regard des spécifications

La Situation d'Apprentissage et d'Évaluation (SAÉ) S1.01 compte pour **40% de la compétence 1** au premier semestre.

L'objectif de cette SAÉ est d'écrire **par équipe de 2 personnes** un programme java qui permette de jouer des parties du jeu de Marienbad. Le jeu de Marienbad de cette SAÉ sera une variante du jeu de Marienbad du film et pourra avoir un nombre de lignes variant entre 2 et 15. Le nombre d'allumettes par ligne à l'initialisation suivra la suite logique annoncée précédemment : 1, 3, 5, 7, 9, ...

Partie 1 : joueur contre joueur

Dans un premier temps, le programme doit permettre à deux joueurs de jouer l'un contre l'autre.

Déroulement d'une partie

Une partie se déroule souvent le scénario suivant :

1. saisie du nom du joueur qui joue en premier,
2. saisie du nom du joueur qui joue en deuxième,
3. saisie du nombre de ligne
4. affichage du jeu (exemple pour 5 lignes) :

```
0 : |
1 : |  |  |
2 : |  |  |  |  |
3 : |  |  |  |  |  |  |
4 : |  |  |  |  |  |  |  |  |
```

5. affichage du nom du joueur qui joue,
6. saisie de la ligne et du nombre d'allumettes à enlever,
7. changement de joueur,
8. reprise de la séquence d'affichage du jeu et du nom du joueur, de saisie de la ligne et du nombre d'allumettes,
9. arrêt du jeu quand il ne reste plus aucune allumette,
10. affichage du nom du vainqueur (celui qui a pris la dernière allumette).

Le programme doit être commenté (Javadoc en français ou anglais) et composé de plusieurs fonctions. Pour les fonctions dont cela est possible, proposer des méthodes de test associées (`testNomFonc` et `testCasNomFonc`).

Partie 2 : joueur contre ordinateur

Dans un deuxième temps, le programme doit permettre à un joueur d'affronter l'ordinateur. Idéalement, le programme doit chercher à gagner. Mais comme expliqué ci-dessous, suivre une stratégie gagnante n'est pas facile. Voici les stratégies possibles de la plus simple à la plus complexe (il est toujours préférable d'avoir un programme qui fonctionne même s'il suit une stratégie non optimale qu'un programme qui ne marche pas) :

1. le programme fait un déplacement au hasard ;
2. le programme ne cherche une solution gagnante qu'en fin de partie ;
3. le programme suit la stratégie gagnante

L'ordinateur doit bien sûr toujours respecter les règles du jeu : à son tour, il doit retirer au moins une allumette et au maximum le nombre d'allumettes présente sur la ligne visée. Il ne peut pas retirer sur un même tour des allumettes sur des lignes différentes.

Stratégie gagnante

La stratégie gagnante est expliqués sur https://fr.wikipedia.org/.

Nous rappelons que l'objectif est de prendre la dernière allumette.

La méthode repose sur le système binaire. La position de départ, précisée par le dessin ci-dessous, s'analyse à l'aide des calculs suivants (généralisable à n lignes (n>0)).

Position de départ :

```
0 : |
1 : |  |  |
2 : |  |  |  |  |
3 : |  |  |  |  |  |  |
```

Équivalent binaire :

```
1 = 0 0 1   en binaire
3 = 0 1 1   en binaire
5 = 1 0 1   en binaire
7 = 1 1 1   en binaire
```

Si on effectue les sommes des chiffres du binaire colonne par colonne en base dix, on trouve :

```
S = 2 2 4
```

Selon le théorème de Sprague-Grundy, une position est gagnante **pour le joueur qui l'atteint** si et seulement si tous les chiffres de S sont pairs. Le joueur qui commence son tour sur une telle position sera dans une position perdante à la fin de son tour. Ainsi, dans l'exemple donné, la position initiale est perdante pour le premier joueur, son adversaire ayant la possibilité de conserver cette propriété de S tout le long de la partie jusqu'à ce qu'il ne reste plus d'allumette.

Déroulement d'une partie

Une partie se déroule souvent le scénario suivant :

1. saisie du nom du joueur,
2. choix de celui qui joue en premier joueur ou ordinateur,
3. saisie du nombre de ligne
4. affichage du jeu (ici, pour 4 lignes) :

```
0 : |
1 : |  |  |
2 : |  |  |  |  |
3 : |  |  |  |  |  |  |
```

5. affichage du nom du joueur qui joue,
6. saisie de la ligne et du nombre d'allumettes à enlever s'il s'agit du joueur.
7. changement de joueur,
8. déroulé du tour de l'ordinateur (choix automatique d'une ligne et d'un nombre d'allumette)
9. arrêt du jeu quand il ne reste plus aucune allumette,
10. affichage du nom du vainqueur (ordi ou nom du joueur).

Déroulé de la SAÉ

Cette SAÉ sera à faire en binôme. **Les membres du binôme feront forcément partie d'un même groupe TP.**

Vous avez 12h de créneaux en autonomie dédiés à S1.01 (visible sur l'emploi du temps en semaine 42).

Le rendu devra être fait avant le **jeudi 17/10 à 23h59** (voir section "À rendre").

Vous devrez présenter votre travail à votre enseignant **le vendredi 18/10** pendant une séance visible à votre emploi du temps. Pendant cette séance, chaque groupe, chacun a leur tour et pendant 10 min fera une démonstration de son travail. La démonstration n'a pas besoin d'être préparée car ce sera l'enseignant qui testera le code. De plus, les étudiants devront répondre à des questions sur la SAÉ S1.01 et sur le cours de R1.01 en général.

À rendre

- Fichier source : MarienbadJvsJ_Nom1_Nom2.java (pour la version joueur contre joueur). Par exemple : MarienbadJvsJ_Dupont_Martin.java
- Fichier source : MarienbadJvsO_Nom1_Nom2.java (pour la version joueur contre ordinateur). Par exemple : MarienbadJvsO_Dupont_Martin.java
- la trace de l'exécution de plusieurs parties.
- un fichier pdf décrivant 1) la structure utilisée pour stocker le jeu, 2) la stratégie de l'ordinateur et 3) indiquera la quantité de travail réalisé par chaque membre du groupe en pourcentage (50-50 si le travail a été également réparti).

Les quatre fichiers doivent être compressés dans une archive Nom1Prenom1_Nom2Prenom2.zip, par exemple DupontDimitri_MartinFerdinand.zip.

Critères d'évaluation

La note de SAÉ rendra compte des critères suivants :

- respect du format du rendu, nom du fichier, et compression en zip,
- respect de la date de rendu,
- respect des conventions java,

- décomposition en méthodes,
- présence de méthodes de test,
- exécution correcte et sans erreur du programme,
- qualité du code rendu.
- qualité des réponses aux questions
- taux de participation à la SAÉ

Cette liste n'est pas exhaustive et d'autres critères pourront être ajoutés par les enseignants de la SAÉ.