Université de Paris – Département Informatique

Bases de la programmation orientée objet

– Projet –

'Crazy Circus', un jeu de Dominique Ehrhard

(http://crazycircus.free.fr/)

1 Présentation du jeu

Le jeu a pour but de déterminer quel est le meilleur dompteur parmi les joueurs. Il s'agit de réussir un exercice de domptage mettant en œuvre trois animaux : un lion, un ours blanc et un éléphant. Ces animaux sont répartis sur deux podiums – l'un est bleu, l'autre est rouge. Lorsque deux ou trois animaux sont sur le même podium, ils forment une pile (au sens commun du terme) comme cela est illustré à la fin de ce document.

Les animaux réagissent uniquement à 5 ordres distincts :

- **KI** – L'animal se trouvant en haut de la pile du podium bleu saute pour rejoindre le sommet de la pile du podium rouge.
- **LO** – L'animal se trouvant en haut de la pile du podium rouge saute pour rejoindre le sommet de la pile du podium bleu.
- **SO** – Les deux animaux se trouvant au sommet des piles des deux podiums échangent leur place.
- **NI** – L'animal se trouvant en bas de la pile du podium bleu monte et se place en haut de la pile de ce même podium.
- **MA** – L'animal se trouvant en bas de la pile du podium rouge monte et se place en haut de la pile de ce même podium.

Le but du jeu est de trouver le plus rapidement la bonne séquence d'ordres qui, réalisés par les animaux, les conduiront d'une situation de départ à une situation à atteindre. Le joueur ayant trouver le premier une bonne séquence remporte le tour. Un joueur ayant donné une mauvaise séquence ne peut plus en proposer d'autre pour ce tour. Si, durant un tour, il n'y a plus qu'un joueur à ne pas avoir proposer de séquence, celui-ci remporte le tour.

La situation initiale de début du jeu et la situation à atteindre à chaque tour sont déterminées en tirant au hasard l'une des 24 cartes représentant chaque situation possible. La situation atteinte après un tour est la situation de départ du tour qui suit. Une carte une fois tirée n'est pas remise en jeu et la partie se termine lorsque les cartes sont épuisées. Le joueur ayant remporté le plus de tours gagne la partie.

Les règles complètes du jeu sont accessibles en suivant ce lien :
http://crazycircus.free.fr/goodies/regles_crazy_circus.pdf

2 Travail à faire

Vous devez réaliser un programme permettant à des joueurs de s'affronter. Votre programme devra gérer une partie dans sa totalité.

Les identités des joueurs (leurs prénoms ou surnom par exemple) doivent être reçus sur la ligne de commande au lancement du programme. Votre programme doit afficher la situation de jeu sous la forme reproduite ci-contre.

La situation de départ est donnée en haut à gauche (ici le lion est sous l'ours sur le podium bleu alors que l'éléphant est seul sur le podium rouge) et la situation à atteindre est indiquée à droite (ici le lion est sur l'éléphant sur le podium bleu alors que l'ours est seul sur le podium rouge). Les différents ordres possibles sont rappelés en bas (les sauts d'un sommet à l'autre sont représentés par des flèches et un déplacement du bas vers

le haut est symbolisé par le caractère ^). Les séquences KIMALONI et SONI sont deux solutions pour cette situation de jeu.

```
  OURS                          LION
  LION     ELEPHANT          ELEPHANT     OURS
  ----       ----     ==>     ----         ----
  BLEU       ROUGE             BLEU        ROUGE

-----------------------------------------
KI : BLEU --> ROUGE     NI : BLEU   ^
LO : BLEU <-- ROUGE     MA : ROUGE  ^
SO : BLEU <-> ROUGE
```

Lorsqu'un joueur veut jouer, il saisi son identité suivie de la séquence d'ordres qu'il propose. Si DP est une identité connue, le joueur peut saisir '`DP KIMALONI`' par exemple.

Si l'identité n'est pas connue, la séquence doit être ignorée et un message d'erreur affiché. Si le joueur n'avait pas le droit de jouer (*i.e.* il a déjà fait une erreur) ou que la séquence n'est pas la bonne, un message informatif doit être affiché par le programme. Si la séquence est correcte, le joueur remporte un point, une nouvelle situation de jeu est choisie aléatoirement par le programme parmi celles qui n'ont pas encore été jouées et un nouveau tour peut débuter.

En fin de partie, le score et le rang des joueurs sont affichés (par score décroissant et par ordre alphabétique en cas d'égalité).

Qui, quoi et quand ?

Votre projet doit être fait en binôme. Les groupes de 3 ne seront pas acceptés. Évitez de faire votre projet tout seul (soit vous êtes très fort et des personnes ont besoin de votre aide, soit vous avez des difficultés et il faut vous faire aider).

Vous devez porter une attention particulière à la rédaction de votre dossier. Sa qualité est déterminante pour l'évaluation de votre travail. Votre dossier doit être un unique document `pdf` dont la composition est la suivante :

- Une page de garde indiquant le nom et **le groupe** des membres du binôme, l'objet du dossier.
- Une table des matières de l'ensemble du dossier.
- Une brève introduction du projet.
- Le diagramme UML des classes formant vos applications. Seuls les noms des classes et leurs dépendances sont à reporter, il est inutile de préciser leurs attributs et méthodes. Toutefois, vous devez y représenter l'organisation en paquetage.
- Le code Java des tests unitaires de vos classes (en précisant lesquels s'exécutent avec succès).
- Le code Java complet de votre projet. L'ordre dans lequel vous présentez vos classes facilite la lecture. Aller des classes élémentaires (celles qui ne dépendent d'aucune autre classe) aux classes plus complexes (en respectant l'ordre de dépendance) est un bon choix.
- Un bilan du projet (les difficultés rencontrées, ce qui est réussi, ce qui peut être amélioré).

Nous vous rappelons que le critère principal de notation est la structuration de votre application. Votre rapport doit mettre en avant cette qualité.

Vous devez rendre votre rapport complet imprimé le **vendredi 17 mars 2023**. De plus, vous déposerez à la même date une archive portant votre nom et contenant l'ensemble de vos fichiers sources (et uniquement cela) dans le puits `BPO1`. Seules les archives au format `jar` ou `zip` seront acceptées.