

Nom : _____
Prénom : _____
Groupe : _____

/ 20

Haute École de Bruxelles
École Supérieure d'Informatique
Bachelor en Informatique

18 novembre 2015
DEV1 – LAJ
JLC

DEV1 – Laboratoires de Java

Interrogation 3

Rock Scissors Paper

- Vous réaliserez votre travail sur **linux1** et le déposerez dans le casier **linux** de votre professeur par la commande **casier**.
- Vous disposez de toutes vos notes ainsi que de l'aide en ligne.
- Vous avez une heure de temps.
- Il ne suffit pas que votre code compile. Testez-le pour identifier d'éventuelles erreurs à l'exécution.
- La cotation tiendra compte aussi du style de programmation que vous avez acquis.
- Respectez bien les noms de classes et méthodes demandés dans l'énoncé.

1

Préparation

(-)

Créez un répertoire `~/evaluations/i3/`. Changez les droits sur votre répertoire `i3` pour donner les permissions de lecture et d'exécution aux professeurs mais aucun droit aux autres étudiants. Appelez votre professeur pour lui montrer que vos permissions ont bien été changées.

Vous ne continuerez pas l'interrogation tant que cette question n'a pas été validée par votre professeur.

2

Description du jeu

(-)

```
Bienvenue au jeu de Pierre Papier Ciseau
-----
Entrez le niveau de difficulté du jeu
    0 : Facile
    1 : Moyen
    2 : Difficile
1

Entrez votre choix
    0 : Pierre
    1 : Papier
    2 : Ciseau
0

Vous jouez Pierre                IA joue Pierre
---'_____)                      ---'_____)
      (_____)                    (_____)
      (_____)                    (_____)
VR   (_____)                    VR   (_____)
---'_____)                      ---'_____)

Résultat : Match Null
```

FIGURE 1 – Un match entre un utilisateur et une IA en mode moyen qui se conclut par un match null.

Nous allons programmer un jeu de *Pierre Papier Ciseau*. Le joueur doit affronter une intelligence artificielle (IA) qui possède trois niveaux de difficultés. Le jeu commence par vous demander le niveau de difficulté de cette IA. Ensuite vous devez entrer votre proposition : *Pierre*, *Papier* ou *Ciseau*. Soit vous gagnez le match, soit vous le perdez, soit il y a égalité.

3

Recherche d'un fichier

(2 points)

La classe `IaRockPaperScissors.class` est disponible sur `linux1` dans un sous-dossier de `/eCours`. Quelle commande exacte allez-vous utiliser afin de retrouver le fichier `IaRockPaperScissors.class` ?

4

Lecture de la javadoc

(2 points)

Dans le dossier où vous avez trouvé la classe, vous disposez également de sa documentation.

Via la consultation de cette documentation donnez le package de la classe `IaRockPaperScissors.class` ?

Profitez-en pour lire la documentation des deux méthodes.

- `iaChoice(int difficultyLevel, int userChoiceId)`
- `isIaWins(int iaChoiceId, int userChoiceId)`

5

Création du jeu

(8 points)

Ecrivez la classe `RockPaperScissorsGame` dans le package `game.user`. Cette classe contient deux méthodes :

- `checkWin()` : méthode qui détermine le vainqueur du match ;

```
algorithme checkNumber(userChoice : entier, iaChoice : entier)
    si userChoice = iaChoice alors
        afficher "Résultat : Match null "
    sinon
        si isIaWins(iaChoice, userChoice) alors
            afficher "Résultat : Vous avez perdu ! "
        sinon
            afficher "Résultat : Vous avez gagné ! "
        fin si
    fin si
fin algorithme
```

- `main(String[] args)` : le point d'entrée de la classe ;

```
algorithme main(args : tableau de chaine)
    difficultyLevel : entier
    userChoice : entier
    iaChoice : entier
    afficher "Entrez le niveau de difficulté du jeu"
    afficher "0 : Facile"
    afficher "1 : Moyen"
    afficher "2 : Difficile"
    demander difficultyLevel
    afficher "Entrez votre choix"
    afficher "0 : Pierre"
    afficher "1 : Papier"
    afficher "2 : Ciseau"
    demander userChoice
    iaChoice ← iaChoice(difficultyLevel, userChoice)
    checkWin(userChoice, iaChoice)
fin algorithme
```

6

Gestion des packages

(2 points)

Ecrire ici (l'important dans les questions qui suivent est la cohérence de l'ensemble) :

- votre répertoire de travail (probablement `~/evaluations/i3/<votre choix>`) ;

- l'instruction que doit contenir votre classe pour faire partie du package demandé ;

- la commande (complète et précise) que vous utilisez (à partir de votre répertoire de travail) pour **compiler** votre classe ;

- la commande (complète et précise) que vous utilisez (à partir de votre répertoire de travail) pour **exécuter** votre classe ;

- le contenu minimal de votre variable d'environnement `CLASSPATH`

7

Gestion des erreurs

(4 points)

Si vous n'entrez pas un niveau de difficulté compris entre 1 et 3 ou un choix compris entre 0 et 2 la méthode `iaChoice(int difficultyLevel, int userChoiceId)` envoie une `IllegalArgumentException`. Gérez cette `Exception` dans la méthode `main`

et affichez un message d'erreur adéquat avant de terminer l'exécution du programme via la méthode¹ `System.exit(-1)`.

```
Bienvenue au jeu de Pierre Papier Ciseau
-----
Entrez le niveau de difficulté du jeu
    0 : Facile
    1 : Moyen
    2 : Difficile
4

Entrez votre choix
    0 : Pierre
    1 : Papier
    2 : Ciseau
1

Les paramètres sont incorrects !
Niveau de difficultés est 1, 2 ou 3
Choix de l'utilisateur est 0, 1 ou 2
```

FIGURE 2 – L'utilisateur encode une valeur erronée pour le niveau de difficulté, ce qui provoque l'affichage d'un message d'erreur et l'arrêt du programme.

Si l'utilisateur n'entre pas un nombre entier pour le niveau de difficulté ou pour le choix, une `InputMismatchException` est envoyée. Dans la méthode `main` gérez cette `Exception` et affichez un message d'erreur adéquat avant de terminer l'exécution du programme via la méthode `System.exit(-2)`.

```
Bienvenue au jeu de Pierre Papier Ciseau
-----
Entrez le niveau de difficulté du jeu
    0 : Facile
    1 : Moyen
    2 : Difficile
A
ERREUR vous devez entrer un entier !
```

FIGURE 3 – L'utilisateur encode une lettre au lieu d'un nombre pour le niveau de difficulté de l'IA, ce qui provoque l'affichage d'un message d'erreur et l'arrêt du programme.

8

Documentation

(2 points)

Si vous ne l'avez pas fait au fur et à mesure, écrire la documentation de la classe `PlayGuess` et de toutes ses méthodes.

Générez là dans le sous-répertoire `~/evaluations/i3/doc`.

Quelle est la commande que vous avez utilisé pour générer cette documentation dans le répertoire `~/evaluations/i3/doc` ?

1. Lisez la documentation de cette méthode au besoin.