

Nom : _____

Prénom : _____

Identifiant : _____ Groupe : _____ Enseignant : _____

/40



Haute École Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

juin 2024
DEV2
Equipe dev2

DEV2 – Développement II

Examen

Examen – dev2

Consignes

1. L'examen dure 1h45 et se fait sur une machine du réseau pédagogique.
2. Vous n'avez pas accès à l'Internet
3. Vous pouvez utiliser vos fichiers disponibles localement et vos notes papier.
4. Avant de commencer, clonez le dépôt (sur Gitesi) qui a été créé pour vous.
5. Tous les fichiers devront être placés dans le clone, créé ci-dessus.
6. Lorsque vous avez fini, créez un commit contenant vos fichiers. Depuis le clone :
 - ▷ `git status`
 - ▷ `git add .`
 - ▷ `git commit -m "remise examen"` (La première commande sert à vérifier que tout est là, et rien d'autre.)
7. Poussez le commit vers le dépôt (sur Gitesi) avec `git push`

TRIOLET est un jeu de chiffres qui consiste à former des ensembles de deux ou trois jetons qui s'entrecroisent sur une grille. Chaque joueur pioche 3 jetons numérotés qu'il pose sur son chevalet. A l'aide des jetons le joueur doit réunir des suites de 2 ou 3 chiffres pour ensuite les poser sur le plateau de jeu. La somme ne doit jamais dépasser les 15 points.

Remarque importante : Pour des raisons de simplicité, il n'y a qu'un et un seul joueur.

MATÉRIEL :

- ▷ 1 plateau composé d'une grille de 225 cases avec entre autres des cases JAUNES et ROUGES attribuant des avantages (nous n'allons pas en tenir compte pour simplifier) ;
- ▷ 81 jetons numérotés de 0 à 15 ayant chacun la valeur correspondant au chiffre principal c'est-à-dire à son numéro. Les 81 jetons sont composés comme suit : 0(9), 1(9), 2(8), 3(8), 4(7), 5(8), 6(6), 7(6), 8(4), 9(4), 10(3), 11(3), 12(2), 13(2), 14(1), 15(1) ;
- ▷ 1 chevalet ;
- ▷ 1 sac renfermant la réserve de jetons.



BUT DU JEU :

A chaque coup, le joueur cherche à obtenir le plus de points possible en posant sur une même ligne horizontale ou verticale, un, deux ou trois jetons accolés à un ou plusieurs jetons déjà placés. L'objectif est de totaliser le plus grand nombre de points à la fin de la partie du jeu.

RÈGLE DE BASE ADOPTÉE :

- ▷ Le total des valeurs des jetons placés côte à côte sur la grille doit être inférieur ou égal à 15 ;
- ▷ En aucun cas, il ne peut y avoir plus de trois jetons placés côte à côte et dans la même direction sur la grille.

FIN DE LA PARTIE : La partie s'arrête dans l'une des situations suivantes :

- ▷ lorsque le joueur a joué un certain nombre donné de coups ;
- ▷ ou lorsque le sac est vide.

CE QUI EST DEMANDÉ

1 Création du projet et du package (1 point)

Créez via IntelliJ un nouveau projet java, maven, nommé dev2-exam-s1-24 dans votre clone. Il vous est demandé d'utiliser le design pattern MVC tout au long du développement de votre application et vous devez travailler dans le package nommé `g12345.triolet` (où vous remplacez 12345 par votre matricule). Tous les autres packages seront placés dans celui-ci.

1 Le modèle

Dans le package `g12345.triolet.model`

1.1 Création des classes

2 Record, énumération et exception (4 points)

- ▷ Créez un record `Position(int row, int col)`, celui-ci représente une position sur un plateau de jeu ;
- ▷ Créez une énumération `Direction` avec comme constantes : `LEFT`, `RIGHT`, `UP`, `DOWN` ;
- ▷ Créez une classe `TrioletException` pour la gestion de toutes les exceptions du jeu de Triolet. Cette classe hérite de la classe `Exception` et a un constructeur qui reçoit en paramètre une chaîne de caractères (message).

3 La classe Token (3 points)

Créez une classe `Token`. Cette classe représente un jeton du jeu. Cette classe aura :

- ▷ un attribut `value` de type entier ;
- ▷ un constructeur à un paramètre ;
- ▷ un getter pour `value` ;
- ▷ une méthode `toString()` qui retourne `value` sous forme de chaîne de caractères.

4

La classe Bag

(4 points)

Créez la classe `Bag` qui contiendra tous les jetons du jeu et permettra de tirer les jetons au hasard pour remplir le chevalet du joueur ou de la joueuse. Cette classe aura :

- ▷ un attribut `tokens` de type liste de `Token` ;
- ▷ un constructeur sans paramètre et qui crée tous les jetons (les 81 jetons) et les ajoute à la liste ; cette liste sera ensuite mélangée de manière aléatoire ;
- ▷ deux méthodes permettant de tirer des jetons au hasard ; la première tire un seul jeton et la seconde en tire `n` jetons :
 - ▷ `public Token draw()`
 - ▷ et `public List<Token> draw(int n)`

5

La classe Board

(4 points)

Créez une classe `Board`, qui représente le plateau de jeu. Le plateau de jeu est constitué de 15x15 cases sur lesquelles l'on pose les jetons. Cette classe aura :

- ▷ un attribut `squares` de type un tableau à deux dimensions de `Jeton`, représentant toutes les cases du plateau de jeu ;
- ▷ un constructeur sans paramètre et qui crée le plateau de jeu de taille 15×15 ;
- ▷ une méthode `public Token getToken(Position p)` qui retourne le Jeton à une certaine position donnée sur le plateau de jeu ;
- ▷ une méthode `public void setToken(Position p, Token token)` qui place un jeton donné sur une position donnée du plateau de jeu ;
- ▷ une méthode `public void setTokens(Position p, Direction d, Token[] tokens)` qui place une série de jetons dans l'ordre sur le plateau de jeu, dans une direction donnée à partir d'une position donnée ;

6

La classe Game

(6 points)

Créez une classe `Game`, qui représente le jeu. Cette classe aura :

- ▷ un attribut `board` de type `Board` ;
- ▷ un attribut `bag` de type `Bag` ;
- ▷ un attribut `rack` de type liste de `Token` ;
- ▷ un attribut `score` de type entier ;
- ▷ un constructeur sans paramètre qui permet de créer une nouvelle partie : un plateau de jeu, un sac de jetons, , un score nul et un chevalet déjà rempli à l'aide du sac préalablement créé ;
- ▷ un getter pour `score` et un getter pour `rack` ;
- ▷ une méthode `public Token getToken(Position p)` qui retourne le jeton à la position donnée sur le plateau de jeu ;
- ▷ une méthode `public void putToken(Position p, Token token)` qui permet au joueur de tenter de placer un jeton donné issu de son rack à une position donnée sur le plateau de jeu ;
- ▷ une méthode `public void putTokens(Position p, Direction d, int[] indexesRack)` qui permet au joueur de tenter de placer certains jetons issu du rack sur le plateau de jeu dans une direction donnée à partir d'une position donnée ;

Les deux méthodes `putToken()` et `putTokens()` sont susceptibles chacune de lancer une `TrioletException` dans les deux situations suivantes :

- ▷ tentative de placer plus de 3 jetons côte à côte dans la même direction ;
- ▷ ou bien le total des valeurs des jetons placés côte à côte est supérieur à 15 ;

7 Les tests (4 points)

Créez huit (8) tests unitaires pour la méthode

`public void putTokens(Position p, Direction d, int[] indexesRack)`, deux(2) tests pour chacune des directions (dont un des deux tests gère l'exception).

2 La vue

8 La classe View (4 points)

Dans le package `g12345.triolet.view`, créez la classe `View` représente une vue du jeu. Cette classe aura :

- ▷ un attribut `game` de type `Game` ;
- ▷ un constructeur à un paramètre ;
- ▷ une méthode `public void displayBoard()` qui affiche le plateau de jeu ;
- ▷ une méthode `public void displayRack()` qui affiche le contenu du chevalet de jeu ;

3 Le contrôleur

9 La classe Controller (5 points)

Dans le package `g12345.triolet.controller`, créez la classe `Controller` qui va prendre la responsabilité de la boucle du jeu et des entrées de l'utilisateur ou de l'utilisatrice. Cette classe aura :

- ▷ un attribut `game` de type `Game` ;
- ▷ un attribut `view` de type `View` ;
- ▷ un constructeur à deux paramètres ;
- ▷ une méthode `public int readInteger(String message, int min, int max)` qui permet de lire de manière robuste, un entier borné au clavier ;
- ▷ une méthode `public void run(int nbCoups)` qui gère la boucle de jeu, elle effectue les action suivantes pour le joueur :
 - Tant que le jeu n'est pas fini Faire :
 - afficher le plateau de jeu, le rack et le score ;
 - placer un jeton ou des jetons sur le plateau de jeu ;
 - compléter le rack,
 - Fin tant que.

4 L'application

10 La classe Triolet et la méthode principale main() (5 points)

Dans le package `g12345.triolet`, créez la classe `Triolet`. Dans cette classe `Triolet`, écrivez une méthode principale `main()` qui permet de jouer une partie entière du jeu. La méthode `main()` réalise ce qui suit :

- ▷ crée un `game` ;
- ▷ crée une `view` ;
- ▷ crée un `controller` et fait appel de manière adéquate à la méthode `run()` ;
- ▷ publie le score final de la partie.

```
Triolet x
=====
LE JEU DU TRIOLET
=====
+---+---+---+---+---+---+---+
| | | | | | | | |
+---+---+---+---+---+---+---+
| | | | | | | | |
+---+---+---+---+---+---+---+
| | | | 7 | 1 | | | |
+---+---+---+---+---+---+---+
| | | | | 2 | | | |
+---+---+---+---+---+---+---+
| | | | | 0 | 3 | 7 | |
+---+---+---+---+---+---+---+
| | | | | | | 8 | |
+---+---+---+---+---+---+---+
| | | | | | | | |
+---+---+---+---+---+---+---+
| | | | | | | | |
+---+---+---+---+---+---+---+
Rack de Roro:[ 1 7 15 ]
Rack de Jojo:[ 1 0 3 ]
```