

Nom : _____
Prénom : _____
Groupe : _____
Identifiant : _____

/ 50

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

2015 – 2016

Développement – 1^{ère}

Examen Star Wars

Consignes

Pour l'ensemble de l'examen,

- Vous avez 4 heures de temps ;
- Vous gérerez ce temps selon votre meilleure convenance pour réaliser les solutions en algorithmique et en Java.

Le jeu à implémenter consiste en la mission de deux *jedis* (Luke et Yoda) sur un plateau de jeu mêlé d'ennemis. Il s'agit d'un jeu de plateau à deux joueurs, le premier joueur jouant Luke représenté à gauche sur la figure 1 et le second joueur jouant Yoda représenté à droite sur la figure 1. Comme on le voit sur la figure 1 le plateau de jeu est divisé en 10 zones¹ représentées par des positions allant de 1 à 10. Chaque personnage du jeu (*jedis* ou ennemis) possède une position. Deux ennemis

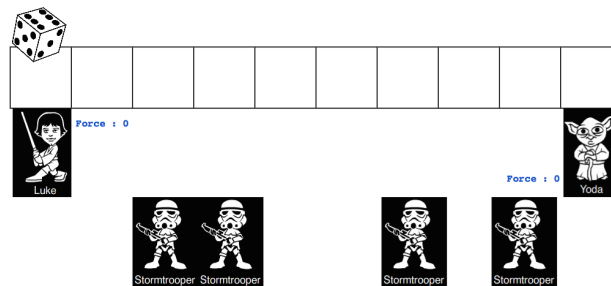


FIGURE 1 – En début de jeu les deux *jedis* sont placés au bord du plateau de jeu. Les ennemis (Stormtroopers) sont placés au hasard sur le plateau de jeu (sauf en première et dernière case). Le dé permettra de mettre en mouvement les *jedis*.

ne peuvent occuper la même case. En début de partie les deux *jedis* sont placés aux bords opposés du terrain (première et dernière case du plateau). Ces deux cases sont choisies comme case départ car aucun ennemi ne peut s'y trouver.

À chaque tour un joueur (Luke ou Yoda) lance le dé et a le droit de se déplacer du

1. Par la suite la taille du plateau de jeu sera variable.

nombre de case(s) indiqué par le dé, soit vers la droite soit vers la gauche, comme il le désire. Luke débute la partie. Sur la figure 2 on voit que Luke obtient un 3 avec le dé et décide d'aller vers la droite.

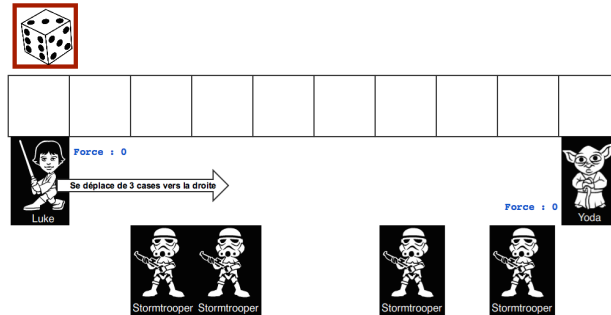


FIGURE 2 – Luke obtient un 3 avec le dé, il décide d'aller vers la droite.

Après avoir effectué son déplacement le *jedi* sécurise la zone (c'est-à-dire la case sur laquelle il vient d'arriver). L'ennemi disparaît de celle-ci. Via cette action la force du *jedi* est modifiée, si un ennemi se trouve sur la zone il gagne un point de force.

Sur la figure 3, Luke après être arrivé sur la case numéro 4 sécurise cette zone. Un ennemi se trouve sur cette case, il gagne un point de force et l'ennemi disparaît.

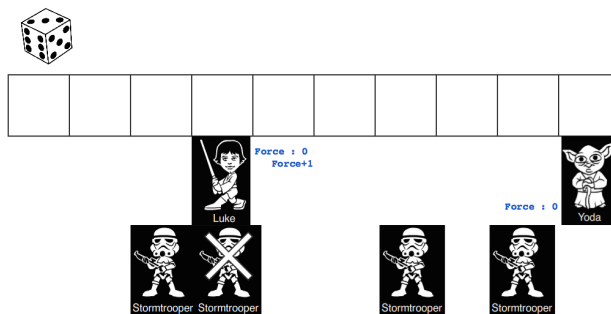


FIGURE 3 – Luke obtient 1 point de force supplémentaire en sécurisant sa zone occupée par un ennemi.

Aucune action particulière n'est à prévoir si l'autre *jedi* occupait cette case.

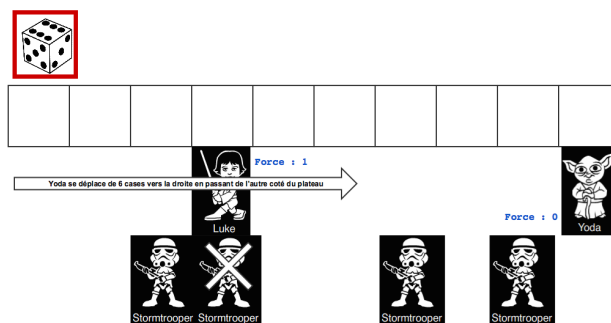


FIGURE 4 – Yoda obtient un 6 avec le dé et décide de se déplacer vers la droite, il fait le tour du plateau de jeu.

Si son déplacement le fait traverser un des bords du plateau, le *jedi* apparaît à l'autre bout du plateau. Le plateau est refermé sur lui même. Sur la figure 4 on observe Yoda qui obtient un 6 après avoir lancé le dé.

Il décide de partir vers la droite et vu que le plateau est fermé sur lui-même, il se retrouve en case 6. Sur la figure 5 Yoda sécurise la case numéro 6, sa force reste

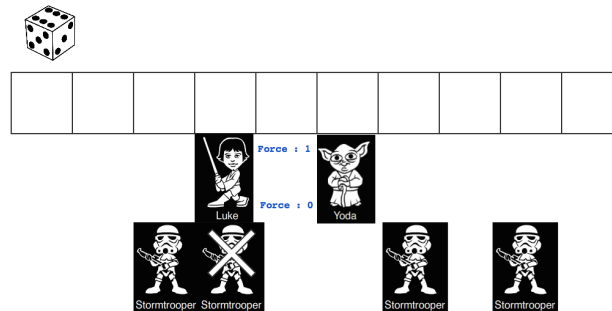


FIGURE 5 – Yoda sécurise la zone, sa force n'est pas modifiée vu qu'aucun ennemi n'est présent.

inchangée vu qu'aucun ennemi ne s'y trouve.

Le jeu se termine lorsqu'il n'y a plus d'ennemis. Le vainqueur est le *jedi* dont la force est la plus grande.

Durant la conception de ce jeu on utilisera des tableaux contenant la position des personnages du jeu :

- un tableau pour la position des deux *jedis*. Dans notre exemple en début de partie ce tableau vaut $\{1, 10\}$.
- un tableau pour la position des ennemis. Dans notre exemple en début de partie ce tableau vaut $\{3, 4, 7, 9\}$.

Le nombre d'ennemis sera compris entre 1 et la moitié de la taille du plateau de jeu et sera choisi aléatoirement en début de partie. Notez que tous ces tableaux seront indicés en commençant à 0.

Le contenu d'une case d'un de ces tableaux est un nombre compris entre 1 et 10 qui représente la position du personnage. Si le personnage disparaît la case prend la valeur 0. Dans notre exemple au moment de la figure 5 le tableau de la position des ennemis vaut $\{3, 0, 7, 9\}$

Un tableau supplémentaire de deux cases contiendra la valeur de la force de chaque *jedi*. En début de partie la force des deux *jedis* vaut 0.

I

Algorithmique

Consignes

Pour la partie algorithmique,

- Vous ne pouvez pas utiliser de notes.
- Vos réponses se feront au bic bleu ou noir sur la feuille de réponses.
- Sauf spécification du contraire, les données lues ou reçues ne comportent pas d'erreurs.
- Les noms en gras (variables et types) doivent être respectés.
- Veillez à travailler de manière modulaire.

Algorithmes disponibles

Dans le cadre de cet examen vous pouvez utiliser si nécessaire les algorithmes ci-dessous sans les écrire :

- algorithme **hasard**($n \downarrow$: entier) \rightarrow entier qui renvoie un entier au hasard entre 1 et la valeur n ;
- algorithme **estLibre**($ennemis \downarrow$: tableau de **nbEnnemis** entiers, $position \downarrow$: entier) \rightarrow booléen qui reçoit un tableau de **nbEnnemis** cases et une position en paramètre et qui renvoie un booléen indiquant si la position est occupée par un ennemi du tableau passé en paramètre ;
- algorithme **securiseZone**($ennemis \downarrow \uparrow$: tableau de **nbEnnemis** entiers, $position \downarrow$: entier) \rightarrow booléen qui retourne 0 si aucun ennemi ne se trouvait à la position donnée en paramètre et 1 sinon. Cet algorithme met également à jour le tableau **ennemis** en mettant à 0 la position de l'ennemi occupant la position donnée en paramètre, cette action fera disparaître l'éventuel ennemi qui s'y trouve. Si on donne le tableau d'ennemis $\{3, 4, 7, 9\}$ et la position 4 en paramètre de cet algorithme, il renverra la valeur 1 et le tableau ennemis vaudra $\{3, 0, 7, 9\}$;
- algorithme **affichePlateau**($jedi \downarrow$: tableau de 2 entiers, $force \downarrow$: tableau de 2 entiers, $ennemis \downarrow$: tableau de **nbEnnemis** entiers, $taille \downarrow$: entier, $joueur \downarrow$: entier) qui affiche le plateau de jeu sur base des différentes données de la partie (position des deux *jedis*, force des deux *jedis*, position des ennemis, taille du plateau de jeu et joueur courant).

1

Compte des ennemis

(5 points)

Écrivez un algorithme **compteEnnemis** qui reçoit un tableau **ennemis** de **nbEnnemis** entiers représentant les positions des ennemis et qui compte le nombre d'ennemis non éliminés. Au début du jeu tous les ennemis ont une position (un nombre de 2 à **taille du plateau-1**, les positions 1 et **taille du plateau** étant occupée par les *jedis*). Après une élimination la position de l'ennemi passe à 0.

2

Déplacement des jedis

(5 points)

Écrivez un algorithme **calculePosition** qui :

- reçoit **jedi** un entier supérieur ou égal à 1 correspondant à la position du *jedi* à déplacer ;
- reçoit **taille** un entier correspondant à la taille du plateau de jeu ;
- lance un dé à 6 faces ;
- demande au joueur s'il souhaite se déplacer vers la droite, dans le cas contraire il se déplace vers la gauche ;
- calcule et retourne la nouvelle position du *jedi* en prenant garde aux conditions aux bords du plateau de jeu (rappelez-vous le déplacement de Yoda à la figure 4).

3

Initialiser le tableau des ennemis

(5 points)

Écrivez un algorithme **placerEnnemis** qui :

- reçoit **nbEnnemis** le nombre d'ennemis à générer ;
- reçoit **taille** la taille du plateau du jeu ;
- retourne un tableau de **nbEnnemis** d'entiers contenant les positions des différents ennemis.

Ces positions sont choisies **aléatoirement** et ne peuvent se répéter. Si un ennemi occupe la position numéro 4, les ennemis suivants devront occuper une position différente. La première et la dernière position du plateau de jeu sont interdites vu qu'elles sont réservées à Luke et Yoda.

Pour écrire cet algorithme n'hésitez pas à utiliser l'algorithme **estLibre**(ennemis : tableau de nbEnnemis entiers, position : entier) → booléen. Pour ce faire n'oubliez pas d'initialiser votre tableau de **nbEnnemis** d'entiers à 0 en début d'algorithme.

Si on demande à l'algorithme de générer un tableau de 4 ennemis pour un plateau de taille 10, il retournera par exemple le tableau {3,4,7,9}, qui ne contient ni la position 1 ni la position 10.

4

Jouer une partie

(5 points)

Écrivez un algorithme **jouePartie** qui

- reçoit la taille du plateau de jeu² ;
- crée et initialise le tableau des positions des deux *jedis* Luke et Yoda ;
- crée et initialise le tableau des forces des deux *jedis* ;
- crée et initialise le tableau des positions des ennemis via l'algorithme **placerEnnemis**, le nombre d'ennemis est un nombre aléatoire compris entre 1 et la moitié de la taille du plateau de jeu³ ;
- tant que sur le plateau de jeu se trouve des ennemis faire jouer les deux *jedis* chacun à leur tour :
 - afficher le plateau de jeu ;
 - déplacer le *jedi* via l'algorithme **calculePosition** ;
 - mettre à jour la force du *jedi* après l'appel à l'algorithme **securiseZone**, un *jedi* gagnant un point de force si la zone était occupée par un ennemi ;
 - changer le joueur courant.
- affiche le résultat de la partie, c'est-à-dire le nom du vainqueur (Luke ou Yoda) ou la mention "Match nul".

2. Fixée à 10 dans notre exemple elle peut prendre d'autres valeurs entières. La valeur minimale étant 3.

3. Si le plateau comporte un nombre impair de cases, le résultat est tronqué.

II

Java et laboratoire

Consignes

Pour la partie java,

- 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.
- 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 package, classe, méthodes demandés dans l'énoncé.
- Vous remplacerez bien sûr `g12345` par votre numéro d'étudiant.

5

Sine qua non

(0 point)

Créez un répertoire `evaluations/janvier`. Changez les droits sur votre répertoire `janvier` 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'examen tant que cette question n'a pas été validée par votre professeur.

6

Travailler dans un package

(2 points)

Dans la suite de l'examen, votre classe fera partie du package `evaluations.janvier`.

Votre programme s'appellera `StarWars.java` et sera situé dans

`/home/g12345/evaluations/janvier/sources/`⁴

et `StarWars.class` dans

`/home/g12345/evaluations/janvier/classes/evaluations/janvier`.

Écrivez ici :

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

4. Vous remplacerez évidemment `g12345` par **votre** identifiant

- la commande (complète et précise) que vous allez utiliser pour **compiler** votre classe ;

- la commande (complète et précise) que vous allez utiliser pour **exécuter** votre classe ;

- le contenu minimal de votre variable d'environnement *CLASSPATH*.

7

StarWarsUtils

(2 points)

Nous avons écrit pour vous une classe *StarWarsUtils* qui contient les méthodes :

- `public static boolean estLibre(int[] ennemis, int position)` permettant de savoir si la position `position` est occupée par un personnage du tableau `ennemis` ;
- `public static int securiseZone(int[] ennemis, int position)` permettant de mettre à jour le tableau `ennemis` si un ennemi se trouve à la position `position`. Cette méthode retourne 0 si aucun ennemi ne se trouvait à la position `position` et 1 sinon ;
- `public static void affichePlateau(int[] jedi, int[] force, int[] ennemis, int taille, int joueur)` permettant d'afficher le plateau de jeu.

Nous savons que la classe *StarWarsUtils* se trouve sur la machine *linux1* dans un des sous-répertoires de */eCours*.

La javadoc de la classe est à votre disposition dans un répertoire `doc` situé à côté de la classe.

- Quelle commande allez-vous utiliser pour retrouver le fichier *StarWarsUtils.class* ?

- Que devrez-vous ajouter au contenu de la variable d'environnement *CLASSPATH* pour pouvoir exécuter la classe *StarWarsUtils* ?

- Que s'affiche-t-il à l'écran lors de l'exécution de la classe *StarWarsUtils* ?

8

StarWars

(14 points)

Lors du td10, vous avez écrit pour vous une classe *Util* qui contient des méthodes utilitaires. N'hésitez pas à les utiliser dans le cadre de l'examen sans oublier de copier le fichier *Util.java* et ses dépendances dans le dossier `/home/g12345/evaluations/janvier/sources/`.

Dans votre classe *StarWars*, écrivez les méthodes :

- `public static int compteEnnemis(int[] ennemis)` ;
- `public static int calculePosition(int jedi, int taille)`, comme pour le td10 les lectures d'entrées au clavier seront robustes ;
- `public static int[] placerEnnemis(int nbEnnemis, int taille)` qui lancera une `IllegalArgumentException` si **nbEnnemis** n'est pas compris entre 1 et **taille/2** ;
- `public static int[] jouePartie(int taille)` .

9

Javadoc

(3 points)

Écrivez la javadoc pour la méthode

`public static int[] placerEnnemis(int nbEnnemis, int taille)`.

Générez la javadoc dans le répertoire `myDoc`. Ecrivez la commande complète que vous avez utilisée pour générer cette documentation.

10

JUnit

(4 points)

Écrivez une série de 4 tests **JUnit** afin de valider la méthode `public static int compteEnnemis(int[] ennemis)`.

Écrivez ici :

- la commande permettant d'afficher le résultat des tests JUnit ;

- le contenu de votre variable d'environnement `CLASSPATH` pour pouvoir exécuter ces tests.