

Ce TP noté est à réaliser sur les dernières séances de TP.

Le projet est à réaliser en binôme et sous Eclipse, en Java (version 1.8).

Une programmation objet mettant en œuvre les concepts d'héritage, de classe abstraite, d'interface et de polymorphisme est exigée.

Ce projet consiste à programmer un jeu en version console puis en version graphique.

La version graphique devra être codée selon l'architecture MVC.

Un jeu de petits chevaux

I. Le jeu

But du jeu :

Le but du jeu est de faire se déplacer alternativement un cavalier de couleur rouge et un autre cavalier de couleur bleue sur un plateau. Chaque cavalier cherche à dépasser la ligne d'arrivée sur un parcours semé d'obstacles. A chaque tour un dé est roulé. Sa valeur indique le nombre de cases que le cavalier va franchir.

Le plateau

Le plateau a une longueur égale aux nombre maximum de cases sur une ligne.

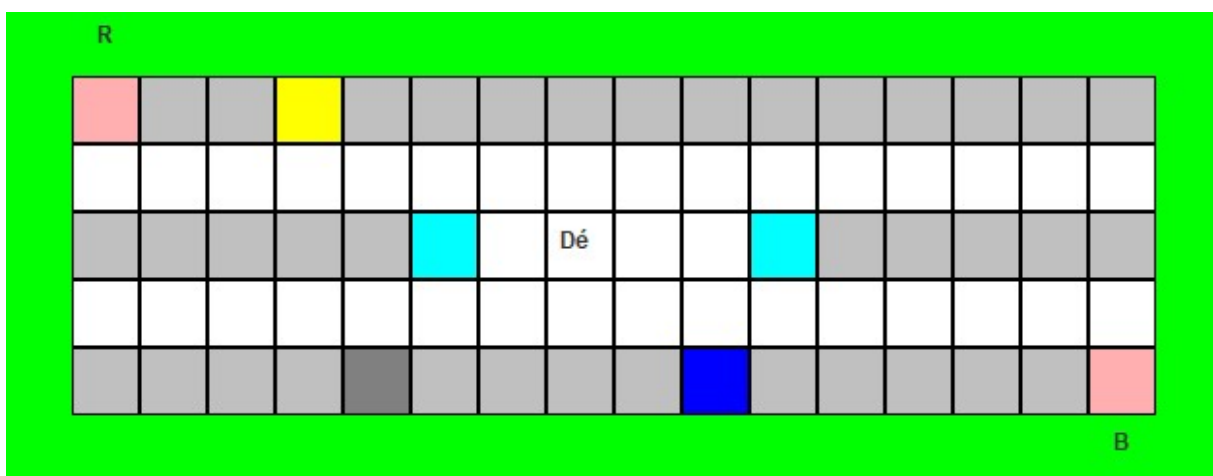


Figure 1. Le plateau

Sur la figure 1, le plateau a une longueur égale à 16. Le cavalier rouge est représenté par la lettre R. le cavalier bleu est représenté par la lettre B.

La case de départ du cavalier rouge est la case rose située en dessous du cavalier rouge. La case de départ du cavalier bleu est la case rose située au-dessus du cavalier bleu.

Ce plateau présente trois obstacles : un trou (case jaune), une rivière (case bleu foncé) et une haie (case gris foncé). Les deux cases en bleu clair représentent les lignes d'arrivée comme indiqué dans les figures 2 et 3 ci-dessous. Les autres cases en gris clair sont des cases dites « neutres » ou « libres ». Les cases blanches ne sont pas concernées par le parcours des cavaliers (voir figures suivantes).

Le parcours de chaque cavalier :

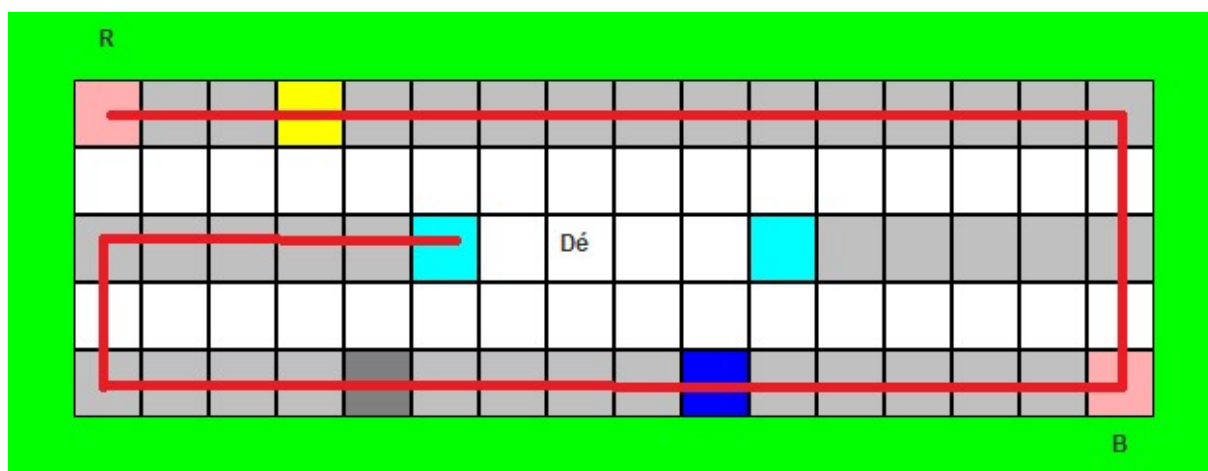


Figure 2. Parcours du cavalier rouge de sa case départ à sa case d'arrivée

Le parcours des cavaliers de leur case de départ à leur case d'arrivée est donné dans la figure 2 pour le cavalier rouge, et dans la figure 3 pour le cavalier bleu. Des obstacles peuvent les retarder dans leur course ou encore les faire revenir à leur case départ, comme détaillé dans la partie II : « Guide pour le codage du jeu ». Un cavalier gagne la partie lorsqu'il dépasse sa case d'arrivée.

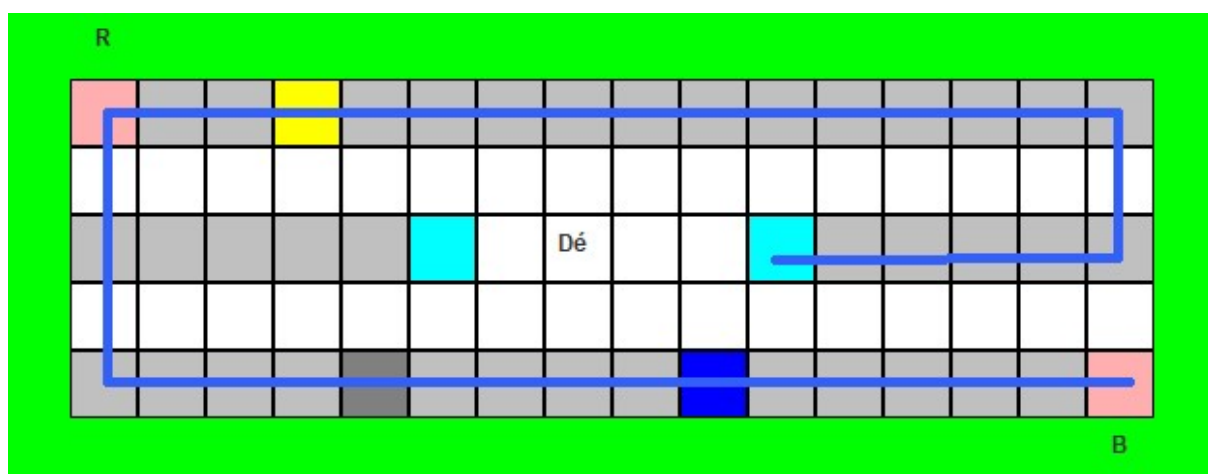


Figure 3. Parcours du cavalier bleu de sa case départ à sa case d'arrivée

Pour rappel, les cases d'arrivée sont de cette couleur : sur le plateau.

Une partie est une suite de tours qui se termine lorsqu'un des deux cavaliers a gagné la partie. Faire évoluer la partie d'un tour consistera à déplacer un cavalier sur son parcours après avoir fait rouler un dé à 6 faces. Pour qu'un cavalier commence à courir sur son parcours, il doit d'abord venir sur sa case départ. Il pourra venir sur sa case départ si c'est son tour de jouer et si la valeur du dé est égale à 6. Ensuite chaque fois que ce sera à son tour de jouer, il avancera du nombre de cases égales à la valeur du dé. Il gagnera s'il est le premier à dépasser sa case d'arrivée. La partie sera alors terminée.

II. Guide pour le codage du jeu :

Dans ce paragraphe quand il est dit qu'une classe comprend (ou inclut) des attributs, ce n'est pas exhaustif. La classe peut éventuellement comprendre d'autres attributs supplémentaires.

1. La classe **Plateau** (Board en anglais) comprend une collection de cases, la liste des deux cavaliers et sa longueur.

Attention : on doit pouvoir modifier la longueur du plateau (un paramètre) et retrouver une application correcte. Cette longueur doit être strictement supérieure à 10 et rester dans une limite raisonnable par rapport aux affichages. Le nombre de cases sur la ligne qui précède la case d'arrivée peut varier selon la longueur du plateau.

On ne modifie pas la hauteur du plateau.

2. Les **cases** (Cell en anglais) sont organisées par héritage. Coder une classe abstraite (*Cell*), les classes modélisant des cases hériteront de celles-ci.

Quand un cavalier joue un tour, la valeur du dé permet de déterminer sa case cible. Le cavalier se déplace alors pour occuper la case cible. S'il rencontre un obstacle sur son déplacement alors il saute cet obstacle (le trou, la haie ou la rivière). Par contre, lorsque le cavalier arrive sur une case (case cible), celle-ci doit réagir à cette sollicitation. Ainsi, chaque case devra implémenter l'interface Questionnable ci-dessous :

```
public interface Questionnable {  
    public String process (Rider r);  
    // le cavalier r est éventuellement modifié par la case  
    // est retourné : un message qui explicite l'effet de la case  
    // sur le cavalier r.  
    // en version console : ce message est affiché dans la console  
    // en version graphique ce message est affiché dans la fenêtre  
    // graphique  
}
```

Chaque case comprend un symbole qui sera affiché (en mode console) pour représenter cette case à l'écran.

La description du *process* de chaque case est décrit ci-dessous

2.1. **Case libre** (CellFree, de symbole '.')

Une case libre (en gris clair sur la figure 1) non occupée par un cavalier reçoit le cavalier sur sa case sans modifier le cavalier.

2.2. **Case occupée par un cavalier.**

Lorsqu'un cavalier r1 arrive sur une case déjà occupée par un cavalier r2 alors le cavalier r1 renvoie la cavalier r2 à sa case départ. Autrement dit, le cavalier r2 recommence son parcours.

2.3. **Case trou** (CellHole, de symbole '@')

Lorsqu'un cavalier c1 arrive sur une case trou (sa case cible est un trou), il tombe dans le trou. Il doit alors passer trois tours avant de continuer à avancer, sauf si l'autre cavalier c2 arrive sur le même trou, auquel cas le cavalier c1 revient à sa case d'épart et le cavalier c2 tombe dans le trou.

2.4. **Case haie** (CellHedge, de symbole '|')

Lorsqu'un cavalier c1 arrive sur une case haie (sa case cible est une haie), il doit franchir la haie. Il doit alors obtenir une valeur de dé impair aux tours suivants, pour la franchir. Si le deuxième cavalier arrive sur la même haie alors le premier cavalier retourne à sa case départ.

2.5. Case rivière (CellRiver, de symbole '~')

Lorsqu'un cavalier c1 arrive sur une case rivière (sa case cible est une rivière), il doit franchir la rivière. Il doit alors obtenir une valeur de dé pair aux tours suivants, pour la franchir. Si le deuxième cavalier arrive sur la même rivière alors le premier cavalier retourne à sa case départ.

Remarque : dans cette version du jeu, une haie ou une rivière comprend une seule case.

2.6. Case coté (CellSide, de symbole '+'), c'est une case située sur les bords du plateau (en vert sur la figure 1).

Le cavalier ne va jamais sur une case coté, sauf avant de commencer à jouer. En effet, chaque cavalier est sur une case coté (le cavalier est à l'écurie) avant de rejoindre sa case départ.

2.7. Case blanche (CellWhite, de symbole '') : le cavalier ne va jamais dessus. Ces cases servent à aérer la disposition des cases sur le plateau.

III. Travail demandé

Les programmes doivent être codés en Java (version 1.8) avec l'IDE Eclipse.

Partie 1 : jeu en mode console

Analyse : Vous rendrez d'abord l'analyse de votre projet pour la version en mode console. Cette analyse prendra la forme d'un diagramme de classes complet de tous les objets que vous envisagez de développer, et décrivant les choix principaux d'implantation retenus pour coder le projet. Cette analyse est à déposer sur Moodle, deux jours avant **vosre séance de TP de la semaine du 26 avril**.

Si votre séance TP est le jeudi 29 avril, vous déposez l'analyse au plus tard le lundi 26 avril à minuit. Si votre séance TP est vendredi 30 avril, vous déposez l'analyse au plus tard le mardi 27 avril à minuit.

Développement :

Le jeu en mode console initialisera le damier selon l'exemple ci-dessous (Figure 4), qui représente un plateau de longueur 16.

```
+ R + + + + + + + + + + + + + + + +
+ # . . @ . . . . . . . . . . . +
+
+ . . . . . * . . . . . +
+
+ . . . . | . . . . ~ . . . . # +
+ + + + + + + + + + + + + + B +
```

Figure 4. Le plateau avec les deux cavaliers R et B à l'écurie.

Les cases « départ » sont représentées avec le symbole #.

Les cases « arrivée » sont représentées avec le symbole *.

A l'issue de chaque tour une trace est affichée à l'écran comme le montre les exemples ci-dessous

Exemples :

Le premier tour en mode console

Le cavalier de couleur ROUGE joue

Valeur du dé : 6

** Le cavalier ROUGE est sur la case départ **

```
+ + + + + + + + + + + + + + + +
+ R . . @ . . . . . . . . . . +
+
+ . . . . . *          * . . . . . +
+
+ . . . . | . . . . ~ . . . . . # +
+ + + + + + + + + + + + + + + B +
```

R ROUGE Case départ B BLEU dans l'écurie

Le tour numéro 2 en mode console

Le cavalier de couleur BLEU joue

Valeur du dé : 1

```
+ + + + + + + + + + + + + + + +
+ R . . @ . . . . . . . . . . +
+
+ . . . . . *          * . . . . . +
+
+ . . . . | . . . . ~ . . . . . # +
+ + + + + + + + + + + + + + + B +
```

R ROUGE case départ B BLEU dans l'écurie

Le tour numéro 3 en mode console

Le cavalier de couleur ROUGE joue

Valeur du dé : 4

** Le cavalier ROUGE est sur une case neutre **

```
+ + + + + + + + + + + + + + + +
+ # . . @ R . . . . . . . . . . +
+
+ . . . . . *          * . . . . . +
+
+ . . . . | . . . . ~ . . . . . # +
+ + + + + + + + + + + + + + + B +
```

R ROUGE case neutre B BLEU dans l'écurie

Le tour numéro 4 en mode console

Le cavalier de couleur BLEU joue

Valeur du dé : 6

** Le cavalier BLEU est sur la case départ **

```
+ + + + + + + + + + + + + + + +
+ # . . @ R . . . . . . . . . . +
+
+ . . . . . *          * . . . . . +
+
+ . . . . | . . . . ~ . . . . . B +
+ + + + + + + + + + + + + + + +
```

R ROUGE case neutre B BLEU case départ

Le jeu en mode console est à rendre pour le **vendredi 7 mai avant minuit (dépôt sur MOODLE)**.

Partie 2 : jeu en mode graphique

Coder une version graphique du jeu.

Le modèle sera une reprise du jeu en mode console, la vue comprendra une fenêtre graphique principale. Un bouton « Tour suivant » permettra de faire du pas à pas.

Le jeu en mode graphique est à rendre pour le **mardi 25 mai avant minuit (dépôt sur MOODLE)**.

Exemple d’affichage de la fenêtre principale au démarrage du jeu :

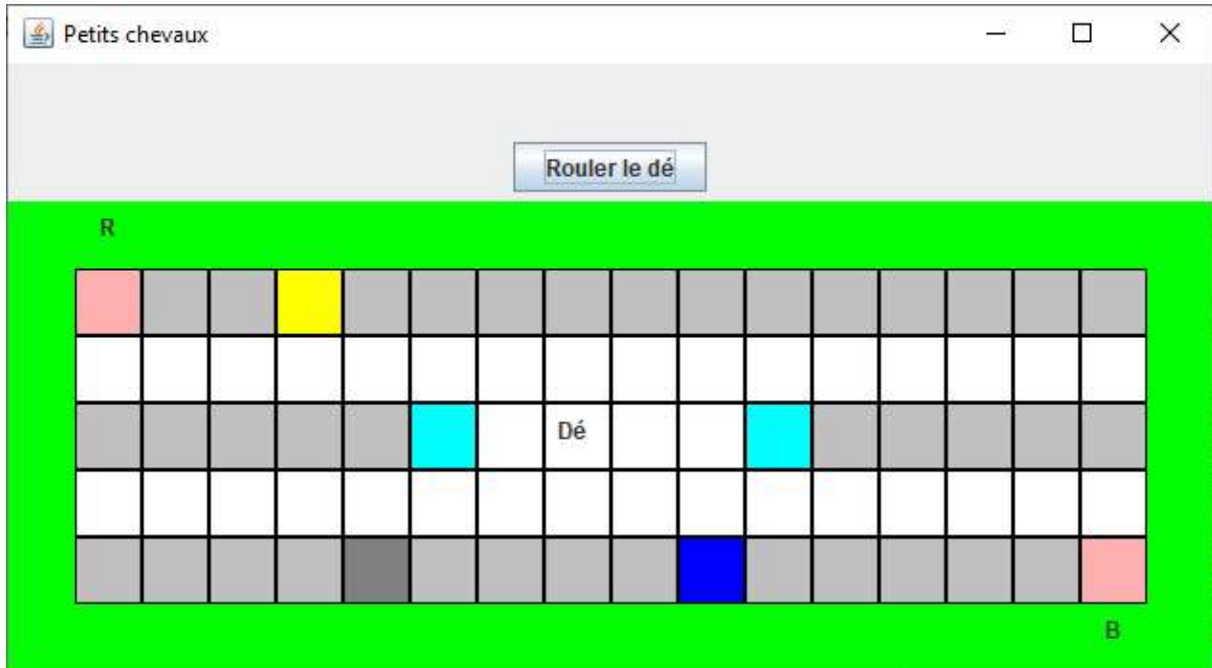


Figure 5. Le jeu au démarrage d’une partie.

Le damier est initialisé de la même façon que celui présenté figure 4 en mode console.

La fenêtre principale comprend trois panels principaux :

- Panel en haut pour les commandes. Avec le bouton « Rouler le dé » et un label qui indiquera quel est le joueur qui joue (voir figure suivante). Chaque tour commence par un roulé de dé !
- Panel au centre : le plateau, les bords du plateau sont en vert. Les cavaliers R et B sont sur le bord près de leur case départ (ils sont à l’écurie en début de partie).
- Panel en bas qui indiquera des informations sur la situation de chaque cavalier (voir figures suivantes).

Le dé est placé au centre du plateau, la valeur du dé y sera affichée (voir figures suivantes).

Puis l'utilisateur appuie sur le bouton « Rouler le dé ». Et la fenêtre principale montre par exemple :

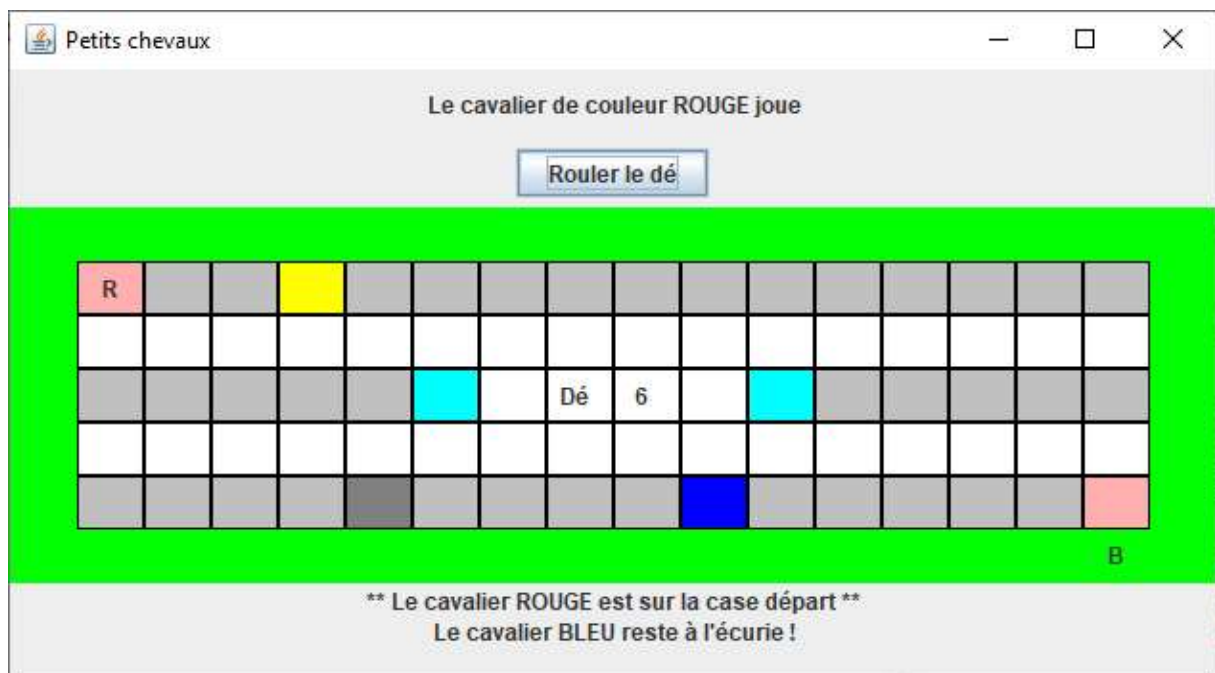


Figure 6. Le premier tour du jeu

A l'issue des tours la partie peut se terminer ainsi :

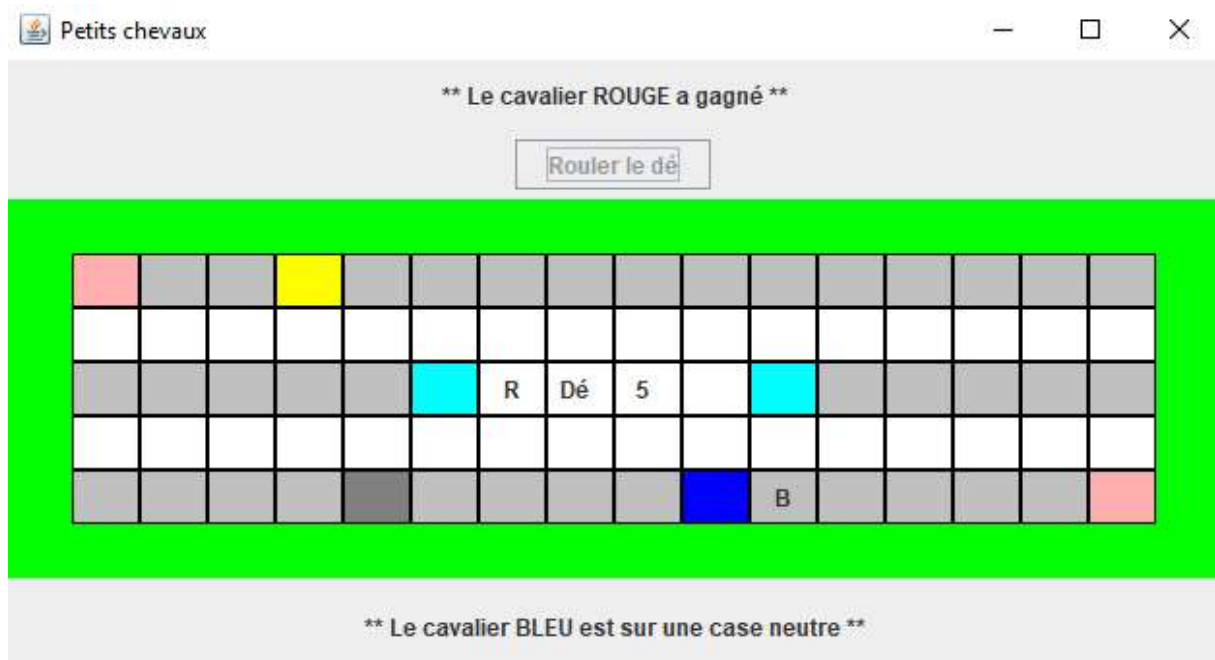


Figure 7. La fin de la partie

Votre application devra donner des affichages proches de figures ci-dessus.

Bonus sur la version graphique : (2 points maximum)

Voici quelques idées de bonus pour la version graphique :

- Tirer au hasard les positions des obstacles.
- Initialisation du plateau : ajouter une fenêtre qui permet par le jeu d'interactions de placer les obstacles du jeu (trou, haie, rivière) avant de commencer une partie.
- Par la bais d'un menu, faire choisir des versions du jeu plus ou moins difficiles en faisant varier la longueur du plateau.
- Pouvoir faire varier le nombre d'obstacles.
- Inventer d'autres obstacles.

Si vous avez des idées de bonus, *faites-en part à votre chargé de TP pour en discuter.*

Calendrier des rendus :

1. Analyse à déposer sur Moodle, deux jours avant **votre séance de TP de la semaine du 26 avril.**
Si votre séance TP est le jeudi 29 avril, vous déposez l'analyse au plus tard le lundi 26 avril à minuit.
Si votre séance TP est le vendredi 30 avril, vous déposez l'analyse au plus tard le mardi 27 avril à minuit.
2. Le jeu en mode console est à rendre pour le **vendredi 7 mai avant minuit (dépôt sur MOODLE).**
3. Le jeu en mode graphique est à rendre pour le **mardi 25 mai avant minuit (dépôt sur MOODLE).**