I.U.T. de Rodez Année 2018 - 2019

DUT Informatique – Semestre 2

*Sujet proposé par C. Servières*

**Projet tuteuré - Semestre 2 - Programmation Java**

**Cahier des charges**

**du jeu des grenouilles et des crapauds**

**Groupe de 4 étudiants**

# Description du jeu des grenouilles et des crapauds

Le jeu oppose deux joueurs ou un joueur et une IA, et il consiste à déplacer des grenouilles et des crapauds sur un plateau quadrillé de N lignes et M colonnes. En début de partie, les grenouilles sont attribuées à l'un des joueurs, les crapauds à l'autre. Le détenteur des grenouilles commence, et ensuite les joueurs jouent à tour de rôle. Le premier qui ne peut plus avancer ses batraciens a perdu.

Les grenouilles avancent horizontalement de la gauche vers la droite, et les crapauds de la droite vers la gauche.

Si la case adjacente dans le sens de déplacement est libre, la grenouille (respectivement le crapaud) déplacée vient occuper cette case. Si, par contre, elle est occupée par un crapaud (respectivement une grenouille) et que la case suivante est libre, le batracien vient occuper cette case libre. Dans tous les autres cas, le batracien ne peut être déplacé.

**Exemple avec un plateau de 1 \* 5:**

1. Plateau initial

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** | **G** |  | *C* | *C* |

1 2 3 4 5

1. La grenouille de la case 2 avance en 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** |  | **G** | *C* | *C* |

1. 2 3 4 5

1. Le crapaud de la case 4 avance en 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** | *C* | **G** |  | *C* |

1 2 3 4 5

1. La grenouille de la case 3 avance en 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** | *C* |  | **G** | *C* |

1. 2 3 4 5

1. Le crapaud de la case 5 avance en 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** | *C* | *C* | **G** |  |

1. 2 3 4 5

1. La grenouille de la case 4 avance en 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** | *C* | *C* |  | **G** |

1. 2 3 4 5

1. Le joueur qui détient les crapauds a perdu puisque ceux-ci ne peuvent plus avancer.

# Description du casse-tête des grenouilles et des crapauds

Les règles de déplacement sont identiques à celles du jeu. Cependant le joueur n’est pas contraint dans le choix des déplacements des grenouilles et des crapauds, il peut alors déplacer plusieurs fois le même pion. Le but de l'unique joueur est d'amener toutes les grenouilles à droite et tous les crapauds à gauche du plateau. Si cette disposition est atteinte, la partie de casse-tête est gagnée.

Exemple de disposition gagnante :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *C* | *C* |  | **G** | **G** |
|  |  | | |  |
| **G** | *C* | *C* |  | **G** |

Exemple de disposition perdante :

Un chronomètre pourra être afficher pour montrer combien de temps le joueur va passer sur le casse-tête à la fin le temps pourra être sauvegarder dans un fichier texte pour être réafficher lors de la prochaine partie comme temps à battre.

# Application à développer

L’application sera dotée d’une interface graphique qui permettra à son lancement de proposer à l’utilisateur d’afficher une aide pour son utilisation.

Mais l’application devra permettre :

* À deux joueurs de s’affronter
* À un unique joueur de jouer contre l’ordinateur
* À un unique joueur d’utiliser le casse-tête
* De gérer des configurations initiales

Dans le cas d’une partie entre deux joueurs, ceux-ci devront saisir leur nom ce qui permettra ensuite à l’application de bien spécifier lequel des deux doit jouer. Les grenouilles et les crapauds seront attribués aux deux joueurs de manière aléatoire. En fin de partie, le nom du gagnant sera clairement affiché.

Dans le cas d’une partie contre l’ordinateur, ou bien lors de l’utilisation du casse-tête, l’application proposera à l’unique joueur d’entrer son nom. S’il n’en donne aucun, il sera désigné par un « joueur humain ». En fin de partie, un message indiquera le résultat de celle-ci et spécifiera le nom de l’unique joueur.

Après une partie, l’application affichera à nouveau le menu principal, ce qui permettra à l’unique joueur ou aux deux joueurs de lancer, éventuellement, une nouvelle partie.

# Configurations initiales

Une **configuration** est la description d'un plateau dans un état précis. Elle indique donc la taille du plateau (nombre de lignes et de colonnes) et la disposition des grenouilles et des crapauds présents sur le plateau. La taille du plateau aura une limite supérieure à fixer (liée aux contraintes d’affichage).

Pour spécifier la configuration initiale d’une partie, l'utilisateur choisira l'une des deux alternatives suivantes :

* Entrer au clavier cette configuration.

* Sélectionner une configuration préétablie. Par conséquent, le programme devra permettre de gérer un ensemble de configurations préétablies que l'utilisateur pourra modifier à son gré par insertion ou retrait d'une configuration. En particulier, lorsque l'utilisateur aura saisi une configuration initiale, le programme lui demandera s'il souhaite l'inclure dans l'ensemble des configurations préétablies.

Le placement des batraciens sur le plateau est libre. Toutefois, l’application vérifiera qu’il y a autant de batraciens de chaque catégorie et que la configuration entrée n’est pas gagnante. Si ce n’est pas le cas, la configuration sera considérée comme invalide.

Il faut qu’il y a forcément une grenouille et un crapaud sur une ligne sinon la configuration est invalide

Exemple de configuration initiale :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **G** |  |  | *C* |  | *C* |
|  | *C* | **G** |  | **G** | **G** |  |
| **G** | **G** |  |  | *C* | *C* |  |
|  | **G** | *C* | **G** |  | *C* | ***G*** |
|  | **G** | *C* |  | *C* | *C* |  |

Les configurations préétablies devront être stockées en mémoire permanente (On utilisera un fichier texte pour stocker les configurations préétablies) afin d’être retrouvées lors du prochain lancement de l’application.

# L’IA

Un joueur pourra affronter une IA en 1 contre 1, cette IA devra réaliser des taches les plus simple :

* Déterminer où se situe ses pions
* Déterminer les cases vide autour d’un pion
* Avancer le pion sur une case vide

Plusieurs difficultés d’IA pourront être ajouter :

* Facile : IA réalisant les taches simple vue au-dessus de façons aléatoires.
* Moyenne : IA réalisant les taches simple vue au-dessus de façon moins aléatoires.
* Difficile : IA réalisant les taches simples en essayant de bloquer l’adversaire.