

Projet tuteuré - Semestre 2 - Programmation Java

Le jeu des grenouilles et des crapauds

Groupe de 4 ou 5 étudiants

Le but de ce projet est de développer un programme permettant de jouer au jeu et au casse-tête des grenouilles et des crapauds.

Description du jeu des grenouilles et des crapauds

Le jeu oppose deux joueurs et 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:

a) Plateau initial

G	G		C	C
1	2	3	4	5

b) La grenouille de la case 2 avance en 3

G		G	C	C
1	2	3	4	5

c) Le crapaud de la case 4 avance en 2

G	C	G		C
1	2	3	4	5

d) La grenouille de la case 3 avance en 4

G	<i>C</i>		G	<i>C</i>
1	2	3	4	5

e) Le crapaud de la case 5 avance en 3

G	<i>C</i>	<i>C</i>	G	
1	2	3	4	5

f) La grenouille de la case 4 avance en 5

G	<i>C</i>	<i>C</i>		G
1	2	3	4	5

g) 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 les déplacements des grenouilles et des crapauds n'alternent pas obligatoirement. 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 :

<i>C</i>	<i>C</i>		G	G
----------	----------	--	----------	----------

Exemple de disposition perdante :

G	<i>C</i>	<i>C</i>		G
----------	----------	----------	--	----------

Application à développer

A son lancement l'application proposera à l'utilisateur d'afficher une aide pour son utilisation.

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. Si ce n'est pas le cas, la configuration sera considérée comme invalide.

Exemple de configuration initiale :

	G			<i>C</i>		<i>C</i>
	<i>C</i>	G		G	G	
G	G			<i>C</i>	<i>C</i>	
	G	<i>C</i>	G		<i>C</i>	G
	G	<i>C</i>		<i>C</i>	<i>C</i>	

Les configurations préétablies devront être stockées en mémoire permanente afin d'être retrouvées lors du prochain lancement de l'application.

Remarque

L'application fonctionnera en mode console ou bien sera dotée d'une interface graphique, au choix du groupe réalisant le projet.

TRAVAIL A REALISER

- Etablir un cahier des charges précis
- Réaliser un diagramme de cas d'utilisation UML
- Faire une description écrite des cas d'utilisation
- Réaliser la conception de l'application en utilisant notamment un diagramme de classes UML
- Développer l'application dans le langage Java, ainsi que des programmes de tests unitaires les plus complets possible
- Réaliser des tests d'intégration
- **Remarque :** Un travail supplémentaire sera demandé aux groupes de 5 étudiants.

Documents à rendre

A l'issue du projet, un dossier technique devra être rendu. Il comportera :

- le cahier des charges, les documents d'analyse, les spécifications détaillées
- un document de conception intégrant le diagramme de classes et des explications pour les éléments les plus complexes
- la répartition des tâches au sein de l'équipe
- la liste des objectifs atteints et non atteints
- un dossier de tests comportant
 - le code source des programmes écrits pour réaliser les tests unitaires
 - le résultat de l'exécution de ces programmes : des copies d'écran et des explications
 - les scénarios de tests (avec les résultats) effectués pour tester globalement l'application
- des exemples de **code source** bien écrit et commenté, **avec des explications rédigées en plus des commentaires présents dans le code**. Il ne faut pas insérer dans le dossier la totalité du code source, mais seulement une vingtaine de pages. Il vous appartient de sélectionner le code source le plus pertinent.
- un manuel utilisateur
- un bilan

La qualité de la rédaction (clarté, précision, mise en forme, orthographe) sera prise en compte dans l'évaluation du projet.

Documents numériques à déposer sur le serveur

- tout le code source du projet, y compris les programmes de test
- un exécutable du jeu
- un fichier au format *pdf* correspondant au dossier décrit dans la section précédente.