

Le travail doit être fait en équipe de 3 (+/- 1) personnes. Idéalement, les équipes devraient rester les mêmes durant toute la session. Aucun retard ne sera accepté.

## Échecs

Vous devez concevoir un logiciel pour jouer aux échecs.

### Règles du jeu

Pour les règles du jeu : <https://fr.wikipedia.org/wiki/%C3%89checs>

En plus des déplacements des pièces et du jeu usuel, il faut implémenter les règles particulières suivantes :

- lorsqu'un pion traverse le jeu au complet, il devient une reine;
- s'il ne reste que les deux rois sur le jeu, la partie est nulle;
- lors du premier déplacement d'un pion, il peut avancer de 2 cases au lieu d'une;

### Démarrer une partie

Lors du démarrage d'une partie, le logiciel demande à l'utilisateur s'il désire jouer contre un humain ou contre un joueur artificiel.

S'il choisit de jouer contre un humain, l'application se connecte à un serveur qui lui affectera un autre joueur par Internet.

S'il choisit de jouer contre un joueur artificiel, il devra ensuite choisir un niveau de difficulté.

### Niveau de difficulté

Le logiciel offrira 2 niveaux de difficulté. Chaque niveau de difficulté correspond à un algorithme d'intelligence artificiel différent.

#### 1<sup>er</sup> algorithme : Niveau débutant

Le joueur artificiel effectue des coups au hasard.

#### 2<sup>ème</sup> algorithme : Niveau avancé

Le joueur artificiel effectue des coups intelligents en fonction de ses chances de gagner.

### Visualiser une partie

Lorsqu'une partie est terminée, il est possible de visualiser le déroulement de la partie en rejouant les tours.

### Sauvegarde et chargement

Pendant une partie contre un joueur artificiel, le jeu peut être sauvegardé dans un document XML. Une partie sauvegardée peut être chargée également.

## **Meilleurs temps**

Lorsqu'un utilisateur joue une partie contre un joueur artificiel, le logiciel doit conserver le meilleur temps pour chaque niveau de difficulté. Ces données sont conservées par chaque installation du logiciel. Autrement dit, les meilleurs temps ne sont jamais transmis sur le réseau.

Lorsqu'un utilisateur bat un record de temps, le logiciel doit lui demander son nom et le conserver avec son temps et son niveau de difficulté. Il doit être possible de consulter les meilleurs temps dans l'application.

### ***TP1 – 12 octobre***

Vous devez faire la conception architecturale du jeu. Vous devez remettre un rapport détaillant l'architecture projetée du système. Le rapport doit contenir :

- le(s) diagramme(s) de cas d'utilisation modélisant les fonctionnalités du système;
- le(s) diagramme(s) de classes du système (un modèle du domaine);
- les diagrammes de séquence ou de communication modélisant les différentes fonctionnalités du système;
- le diagramme de packages du système;
- le diagramme de composants;
- le diagramme de déploiement du système.

Le rapport doit contenir une page de présentation et une table des matières. Le rapport doit contenir le texte nécessaire pour introduire la modélisation et ce qui est présenté dans chaque section du document. Le rapport doit être remis en format PDF par Moodle. Un seul membre de l'équipe doit faire la remise.

Le rapport doit être remise avant le 12 octobre 2017 à 9h30.

### ***TP2 – 9 novembre***

Vous devez faire la conception détaillée du jeu. Vous devez remettre un rapport détaillant la conception projetée du système. Le rapport doit contenir :

- le(s) diagramme(s) de classes illustrant la conception détaillée;
- les diagrammes de séquence montrant les relations entre les classes pour chaque cas d'utilisation;
- le diagramme de packages du système (en version détaillée, c'est-à-dire en indiquant les classes contenues dans les packages);
- chaque responsabilité/action des classes doit être justifiée par un pattern GRASP; cette justification doit être documentée (une justification par méthode publique; une justification pour la création des instances d'un classe).

Le rapport doit contenir une page de présentation et une table des matières. Le rapport doit contenir le texte nécessaire pour introduire la modélisation et ce qui est présenté dans chaque section du document. Le rapport doit être remis en format PDF par Moodle. Un seul membre de l'équipe doit faire la remise.

Le rapport doit être remise avant le 9 novembre 2017 à 9h30.

### **TP3 – 21 décembre**

Vous devez modifier votre conception détaillée afin d'y introduire des patrons de conception GoF. Le nombre de patrons à introduire correspond au nombre de membres dans l'équipe (ex. 3 patrons pour une équipe de 3 personnes). Pour chaque patron, vous devez fournir un diagramme de classe et un (ou plusieurs, si nécessaire) diagramme de séquence illustrant les relations entre les classes du patron et les autres classes du projet.

Vous devez également développer l'application dans le but de valider votre conception. Toutes les fonctionnalités du jeu doivent être développées. Toute différence entre l'implémentation et la conception doit être documentée dans le rapport que vous allez me remettre. Pour chaque problème de conception que vous allez corriger, vous devrez décrire comment vous avez détecté le problème de conception et comment vous l'avez corrigé.

Le projet doit être développé en Java et toutes les bibliothèques doivent être gérées avec Maven. Vous devez me remettre le code source du projet et le rapport (en format PDF). La remise doit être faite par Moodle au plus tard le 21 décembre 2017 à 23h55. Un seul membre de l'équipe doit faire la remise. N'oubliez pas d'inclure une documentation (readme) indiquant comment exécuter votre logiciel.