Université de Paris
Master 1 Informatique

Génie Logiciel Avancé
Année 2020–2021

Examen

6 mai 2021

Instructions Justifiez vos réponses. Les documents sont interdits. Tout dispositif électronique et/ou de communication est interdit. La durée de l'examen est de 2 heures.

Exercice 1 [Testing (théorie), 4 points] Expliquez les différences entre les méthodes de sélection de tests en boîte blanche et en boîte noire. Donnez au moins un exemple de cas de test pour chaque méthode.

Exercice 2 [Conception (théorie), 4 points] Donnez une description du patron de conception « proxy ». Présentez son architecture à objet à l'aide d'un diagramme UML. Expliquez son utilité et donnez au moins un exemple pratique de conception logicielle où il serait utile.

Exercice 3 [Conception (exercice), 7 points] On souhaite réaliser un logiciel de jeu d'échec en ligne nommé WeCheck.

WeCheck devra permettre aux utilisateurs de jouer aux échecs dans un navigateur web en solo contre une intelligence artificielle et à deux à travers un serveur de jeu à distance fourni par les réalisateurs de WeCheck.

Pour les jeux à deux, un mécanisme de liste d'attente devra être fourni pour pouvoir créer des parties et joindre les parties d'un joueur en attente. Un classement des joueurs (sur la base des parties gagnées/perdus par chaque joueur) doit être maintenu et affiché sur le site web de WeCheck.

Des interfaces utilisateur alternatives doivent aussi être supportées, comme notamment des applications pour smartphone. Comme le marché des jeux pour smartphone est très dynamique, WeCheck doit être développé le plus indépendamment possible des plateformes spécifiques et être ensuite porté vers les gammes de smartphones les plus populaires, comme Android et iOS. L'interface de jeu doit être simple et ergonomique, sur toutes les plateformes supportées.

WeCheck doit supporter différentes cadences de jeu, p.ex. blitz, avec temps de réflexion maximum total sur la partie de 15 minutes par jouer, voire encore plus rapides (comme 30 seconds et 1 minute, assez populaires dans le jeu en ligne), jusqu'aux parties rapides de 30 minutes, et parties « par correspondance » (sur plusieurs semaines, avec envoie des coupes par email dans un format standardisé).

Il doit aussi être possible dans le futur de changer les règles du jeu pour supporter des variantes populaires du jeu d'échec comme l'ajout des nouveaux types de pièces, la visibilité restreinte de l'échiquier, le jeu à 3 joueurs, le jeu sur un échiquier 3D, etc.

a) Proposez une architecture logicielle de haut niveau pour WeCheck sous la forme d'une liste de modules. Pour chaque module, donnez son nom et une description de ses responsabilités. Illustrez votre architecture en utilisant un diagramme d'architecture de votre choix (p.ex., un diagramme des paquets UML ou un diagramme des composants C4) en mettent en évidence les dépendances de chaque module.

b) En utilisant un diagramme de classe de votre choix (p.ex., un diagramme des classes UML ou un diagramme de code C4) détaillez la conception à objet d'au moins deux des modules qui composent votre architecture.

Veillez à bien prendre en compte la modularité, la maintenabilité, la réutilisation des patrons de conception, et la possibilité des extensions futures dans vos choix d'architecture.

Exercice 4 [Tests, 5 points]

a) Écrivez en Java l'interface minimale d'un type de données abstrait *dèque* (file à double entrée) d'entiers, une file/pile dans laquelle on peut ajouter et retirer des éléments soit au début soit à la fin.

b) Proposez au moins 10 tests unitaires pour ce type de données, en donnant une courte description (= 1 phrase) de chaque test. Par exemple : « si on ajoute au début d'une dèque vide l'entier 42 et ensuite on retire un élément depuis sa fin, on obtient 42 ».

c) Donnez l'implémentation, avec la bibliothèque de test jUnit, d'au moins 5 tests parmi les tests unitaires proposés.