DEVOIR DE CONTROLE CONTINU

UNIVERSITÉ CAEN NORMANDIE

A RENDRE POUR LE

VENDREDI 25 NOVEMBRE 2022 (LE SOIR AVANT MINUIT)

1. Organisation

Ce devoir de Contrôle Continu est à réaliser en groupes de 4 étudiants. En cas de nécessité, des groupes de 3 étudiants pourront être acceptés, mais en aucun cas des groupes de plus de 4.

Une partie du temps de travail de TP pourra y être consacrée, selon les indications de vos enseignants respectifs, mais du travail hors des heures de TP sera indispensable pour mener à bien ce devoir.

Un dépôt devra être créé sur la forge pour chaque groupe (obligatoirement comme un sous-projet du projet "TP Génie Logiciel L3"). Les noms court et long du dépôt devront comporter les noms des quatre membres du groupe, selon l'exemple suivant :

- nom court "durand-dupont-smith-doe"
- nom long "Génie Logiciel Durand, Dupont, Smith, Doe".

D'autre part, en plus des membres du groupe, Amal Mahboubi, Christophe Charrier, Yohann Jacquier et Yann Mathet devront être ajoutés comme **managers** du dépôt. Si ces points ne sont pas respectés, le devoir ne sera pas corrigé (note = 0).
Attention, lors de la création du sous-projet, ne cochez pas la case faisant « hériter » du projet parent.

Au lendemain du 25 novembre, le code de chaque groupe sera extrait de son dépôt pour correction : un répertoire nommé *livraison* devra être présent à la racine et contenir, pour chacune des deux parties du devoir (voir plus loin) :

- un répertoire *src* contenant le code commenté
- un répertoire *doc* contenant la Javadoc
- un répertoire *dist* contenant l'exécutable ou la librairie (un fichier .jar, et d'éventuelles ressources nécessaires à l'exécution)
- un fichier *build.xml* permettant de re-générer le contenu de dist via ANT.
- parallèlement, un répertoire *rapport* contenant un mini rapport au format PDF contenant toute information utile à la compréhension de votre conception logi-

cielle. Par exemple quelques diagrammes de classes précisant votre mise en œuvre de certains design patterns, le principe de vos algorithmes non triviaux, etc. Ce rapport pourra faire aux environs de 3 à 7 pages bien illustrées.

2. Critères d'évaluation

En premier lieu, nous prendrons en compte la qualité de l'architecture logicielle et de votre code. Nous regarderons notamment si votre conception est :

- facile à comprendre (répartition en packages et en classes cohérente, noms de classes et de méthodes éloquents, commentaires dans le code lorsque c'est utile)
- facile à maintenir et à faire évoluer (pas de code spaghetti, pas d'inter-dépendance entre packages, pas de redondance, mise en œuvre de patterns permettant l'intégration de nouveaux éléments ou algorithmes, etc.)
- robuste (tests effectués)

Bien sûr, une bonne ergonomie, un design agréable, des options supplémentaires seront appréciés mais ne constitueront pas l'essentiel de la note. En particulier, une application qui ferait ce qui est demandé dans le sujet mais qui ne répondrait pas aux critères d'évaluation exposés ci-dessus n'obtiendrait assurément pas la moyenne.

L'utilisation de patterns (en plus de MVC) est aussi un critère important d'évaluation. Des suggestions seront faites au fil des cours et TP.

3. Sujet : Jeux de cartes, Blackjack

On souhaite réaliser des développements autour de jeux de cartes, menant en particulier à la création d'un jeu de « blackjack ».
Une particularité importante de ce devoir est qu'il comportera 2 volets :

- D'une part, une libraire « cartes » permettant de créer et manipuler des cartes, pouvant être utilisée par d'autres applications.
- D'autre part, un jeu de « blackjack » s'appuyant sur cette librairie

Il y aura donc une dépendance de « blackjack » vers « cartes ».
La librairie et le jeu s'appuieront sur MVC, c'est-à-dire proposeront notamment un découpage clair entre modèle et vue(s).

Concernant la librairie « cartes », une version minimale consistera à créer un jeu de 52 cartes classique permettant de jouer par exemple au blackjack. Vous pourrez néanmoins essayer de créer une librairie plus polyvalente, capable de créer différents types de jeux de cartes, donc le jeu de 52 cartes ne serait qu'un cas particulier.
Cette librairie intègrera en plus de la notion de carte, celle de paquet de carte (pouvant représenter la « main » du joueur, etc.), et de manipuler ces derniers (ajout, retrait, etc.). Plusieurs vues pourront être proposées (cartes faces cachées, cartes visibles, en éventail, etc.) pour s'adapter aux différents impératifs des jeux de cartes.

Concernant le jeu « blackjack », vous pourrez en proposer une version intégrant les règles minimales, ou une version plus complexe intégrant les splits et les assurances.
Il s'agira d'une version monoposte où le joueur humain se mesure à des « robots » (des I.A. programmées par vos soins).