

LO02 - Automne 2021 - Projet Splendor

Dans ce projet, il s'agit de concevoir et développer une application permettant de jouer au jeu de société **SPLENDOR** créé par Marc André et édité par *SPACE Cowboys*. Pour découvrir ce jeu, vous pouvez consulter les règles du jeu (relativement simples). Il est conseillé de jouer quelques parties pour bien comprendre les différents mécanismes du jeu. Bien que le plaisir de jouer avec des vraies cartes est indéniable, il existe plusieurs versions en ligne gratuites.

1 Fonctionnalités attendues

L'application devra permettre de jouer des parties standards de 2 à 4 joueurs en mode console. Les joueurs pourront être humains ou virtuels (une IA simulant un joueur). En plus du jeu standard, on doit pouvoir aussi jouer à au moins une des extensions officielles du jeu (voir « les Cités de Splendor »).

2 Éléments d'interface

L'application doit permettre de paramétrer une partie :

- nombre de joueurs ;
- paramètres des joueurs (nom, joueur humain/IA, niveau d'adversité d'une IA) ;
- activation d'une extension ;
- activation de règle(s) spécifique(s).

L'application doit permettre de jouer la partie et de déterminer le gagnant.

A tout moment de la partie, il doit être possible d'annuler la dernière action d'un joueur tant que cette action n'a pas été confirmée.

Vous êtes libre d'organiser votre interface en mode console du moment qu'elle permet de piloter facilement l'application. Les visuels doivent être suffisants pour comprendre l'état de chaque joueur. Il n'est pas demandé d'utiliser de développer une interface graphique ou d'utiliser les graphismes originaux du jeu.

3 Sauvegarde du contexte (optionnelle)

Au démarrage de l'application, le précédent état de l'application (et donc son éventuelle partie en cours) au moment de son arrêt pourra être restauré.

4 Evolution de l'application

L'architecture de votre application devra permettre d'intégrer de nouvelles fonctionnalités sans remettre en cause le code existant (ou a minima). Les choix de conception devront donc permettre de rendre l'application évolutive et notamment de garantir la facilité d'ajout des composants suivants (sans impacter le reste du programme) :

- ajout de nouvelles IA joueurs,
- ajout d'une nouvelle règle (et la vérification de son respect par l'application),
- ajout de nouvelles extensions,
- éventuellement l'ajout d'éléments d'IHM de paramétrage liés aux nouveaux ajouts.

Vous démontrerez la pertinence de votre architecture dans votre rapport en expliquant comment ajouter ces éléments. En choisissant une autre extension (différente de celle que vous avez implémenté), à titre d'illustration, vous expliquerez comment faire (sans le faire) pour implémenter cette autre extension (classes à créer, intégration dans l'architecture, éventuelles modifications de codes, ...).

5 Consignes

- Le projet est à effectuer en groupe de 4 ou 5 étudiants (du même groupe de TD).
- En plus des instructions standards du C++/C++11/C++14, vous pouvez utiliser l'ensemble des bibliothèques standards du C++/C++11/C++14.
- Ne faite la partie "optionnelle" qu'après avoir fait ce qui est obligatoire. Si cette partie n'est pas faite, il n'y a aucune pénalité dans la notation.

6 Livrable

Le livrable final est composé des éléments suivants :

- **Code source** : l'ensemble du code source du projet. Attention, **ne pas fournir d'exécutable ou de fichier objet**.
- **Video de présentation avec commentaires audio** : une courte video de présentation dans laquelle vous filmerez et commenterez votre application afin de démontrer le bon fonctionnement de chaque fonctionnalité attendue (max 10 min, 99 Mo).
- **Rapport** : Un rapport en format `.pdf` (max 15-20 pages) composée des parties suivantes :
 - un bref résumé de ce que permet votre application (en précisant parmi les opérations attendues celles qui ont été implémentées et celles qui ne l'ont pas été) ;
 - la description de votre architecture en justifiant vos choix ;
 - une argumentation détaillée où vous montrez que votre architecture permet facilement des évolutions (notamment au travers de l'extension choisie).
 - une description détaillée du planning constaté de votre projet ;
 - une description détaillée de la contribution personnelle de chacun des membres du groupe sur les différents livrables (cette partie sera notamment utilisée pour la notation). Vous évaluez en % la part de contribution de chaque membre sur l'ensemble. Chaque membre devra aussi reporter une évaluation du nombre d'heures de travail qu'il a consacré au projet.

L'ensemble des livrables est à **rendre avant le 7 novembre 23h59 au plus tard (partout dans le monde)**.

7 Évaluation

Le barème de l'évaluation du projet est comme suit :

- **Couverture des fonctionnalités demandées** : 8 points
- **Choix de conception et architecture** : 4 points. En particulier sera évaluée la capacité de l'architecture à s'adapter aux changements.
- **Évaluation des livrables** : 8 points (video, code source, rapport, respect des consignes).

Remarque : une note inférieure ou égale 8/20 au projet est éliminatoire pour l'obtention de l'UV.

8 Conseils

- Plusieurs TDs utilisent certains concepts communs au projet afin de commencer vous familiariser avec les différentes entités de l'application qui est à développer. On ne perdra pas de vue que les questions développées dans ces TDs ne constituent pas une architecture forcément pertinente pour le projet. Celle-ci devra être complètement retravaillée en tenant compte de l'ensemble des entités du projet.
- La partie difficile du projet est la conception de votre architecture : c'est là-dessus qu'il faut concentrer vos efforts et passer le plus de temps au départ.

- Il est conseillé d'étudier au moins les design patterns suivants qui pourraient être utiles pour élaborer l'architecture de votre projet : decorator, factory, abstract factory, builder, bridge, composite, iterator, template method, adapter, visitor, strategy, facade, memento. En plus de vous donner des idées de conception, cette étude vous permettra de vous approprier les principaux modèles de conception. Attention, cela ne signifie pas qu'ils doivent forcément être utilisés.
- Pour la persistance des informations, vous êtes libres d'élaborer vos formats de fichier.
- L'apparence console de l'application ne sera pas prise en compte dans la notation. Soyez avant tout fonctionnels. Ça peut être très moche du moment que c'est jouable.