

LO21 - 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. 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 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 les graphismes originaux du jeu qui ne sont pas libres de droit.

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 Bibliothèque des joueurs humains - Historique des scores (optionnelle)

Vous pouvez implémenter une bibliothèque des joueurs humains (à utiliser lors du paramétrage) afin de conserver un historique de leurs scores et éventuellement d'autres statistiques de votre choix.

5 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, ...).

6 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.
- L'interface graphique est à réaliser en utilisant le framework Qt.
- Ne faites les parties "optionnelles" qu'après avoir fait ce qui est obligatoire. Si ces parties ne sont pas faites, il n'y a aucune pénalité dans la notation.

7 Livrables attendus

7.1 Rapports intermédiaires

Pour mener ce projet, vous devrez vous organiser de manière efficace. Il sera donc important de tenter de réfléchir aux tâches que vous allez devoir réaliser en vous posant pour chacune d'elle les questions suivantes :

- Cette tâche peut-elle être encore découpée en sous-tâches ?
- Quelle est la complexité de la tâche ?
- Quelle est la durée estimée pour réaliser cette tâche ?
- Quelles sont les relations de dépendance entre cette tâche et les autres ?

Vous devrez ensuite prioriser vos tâches (indispensable, importante, utile, bonus) et les répartir entre les différents membres du groupe de projet.

Au cours du projet, il vous est demandé de rendre compte de votre organisation. vous devrez ainsi rendre 3 courts comptes-rendus (2 à 5 pages maximum) lors des semaines suivantes :

1. la semaine du 15 novembre ;
2. la semaine du 6 décembre ;
3. la semaine du 20 décembre.

Chaque rapport comportera :

- La liste des tâches mises jour (tâches a priori pour le rapport 1) avec l'estimation de leur durée en mettant en évidence les nouvelles tâches par rapport au précédent compte-rendu (rapports 2 et 3).
- Une répartition a priori des tâches qui restent à faire entre les différents membres du groupe de projet.
- L'état d'avancement par rapport au précédent compte-rendu (pour le compte rendu 2 et 3) : la répartition entre les membres du groupe et la durée a posteriori des tâches déjà effectuées ou un pourcentage d'état d'avancement pour les tâches en cours avec la durée de travail déjà fournie.

Bien que la liste des tâches s'affinera au cours du temps, vous devrez d'être précis sur les tâches à court et moyen-terme par rapport au rendu du rapport.

Les rapports devront être déposés sur moodle dans la partie prévue à cet effet avant la fin de la semaine mentionnée (avant samedi 23h59 partout dans le monde).

Il sera tenu compte de la qualité de ces comptes-rendus dans la notation (le manque de précision dans les éléments demandés sera pénalisé)

7.2 Livrable final

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**.
- **Documentation** : une documentation complète en `html` générée avec `Doxygen`.
- **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 évalueriez 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 3 janvier 23h59 au plus tard (partout dans le monde)**. Les éléments du livrable devront être déposés sur moodle dans la partie prévue à cet effet.

8 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, documentation doxygen, rapport, compte rendus intermédiaires, respect des consignes).

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

9 Conseils

- Il est fortement recommandé d'utiliser un logiciel de gestion de versions afin de faciliter le travail collaboratif.
- 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. Il est tout de même conseillé d'utiliser XML et d'utiliser les outils XML de Qt.
- Au lieu d'utiliser des fichiers, vous pouvez utiliser un SGBD comme SQLite.
- L'apparence de l'application ne sera pas prise en compte dans la notation. Soyez avant tout fonctionnels. Ca peut être très moche du moment que c'est jouable. En particulier le design des différents éléments visuels n'ont pas à être élaborés : il faut simplement que l'on puisse jouer.