

## Réalisation d'un framework autour des jeux de cartes

### 1 Présentation

Le but concret est de réaliser des applications permettant de jouer à plusieurs jeux de cartes. Vous observerez en les étudiant que s'ils ont tous certaines particularités qui les distinguent, ils ont surtout beaucoup de choses assez générales en commun.

L'objectif que nous vous assignons consiste à interpréter ce qui selon vous est généralisable, et ce qui est très particulier. Cette analyse vous permettra de structurer ensuite votre code autour de classes que vous réutiliserez de façon optimale pour réaliser plusieurs de ces jeux.

Le **framework** général que nous vous demandons est une sorte de bibliothèque logicielle dont la caractéristique principale est l'usage important de différentes formes de généricité et qui définit un cadre encourageant certaines architectures logicielles.

Dans l'idéal il sera conçu pour être utilisable pour réaliser n'importe lequel des jeux qu'on vous propose ici. Il devra être possible de mélanger des joueurs humains et des robots. Les premiers interagiront à l'aide du clavier, les second devront pouvoir jouer automatiquement sans pour autant posséder une intelligence impressionnante (ils doivent cependant respecter les règles).

Votre première réalisation est imposée : il s'agit de la Bataille <sup>1</sup>. Si votre framework est correctement architecturé, l'implémentation de ce jeu devra être extrêmement rapide et constituera un premier test de sa qualité.

Ensuite il vous faut choisir impérativement au moins 4 jeux parmi les suivants :

- le 8 américain <sup>2</sup>
- le Uno <sup>2</sup>
- la Belote <sup>2</sup>
- le Tarot <sup>2</sup>
- la Briscola <sup>3</sup>
- la Scopa <sup>4</sup>

Nous pouvons admettre que des simplifications aient été appliquées pourvu qu'elles ne consistent pas à rendre triviale l'implémentation (i.e. on peut admettre de ne pas vérifier certaines règles trop compliquées ou subtiles, mais pas de transformer le jeu de Tarot en pseudo-bataille).

---

1. <https://www.regles-de-jeux.com/regle-de-la-bataille/>

2. <https://www.regles-de-jeux.com/>

3. <https://fr.wikipedia.org/wiki/Briscola>

4. [https://fr.wikipedia.org/wiki/Scopa\\_\(jeu\\_de\\_cartes\)](https://fr.wikipedia.org/wiki/Scopa_(jeu_de_cartes)) contentez-vous de la version française

## 2 La réalisation

Le projet doit être réalisé en c++11 naturellement et en binôme. Un salon discord sera ouvert pour vous permettre de former vos binômes. Le projet doit être proprement conçu, architecturé et documenté. L'utilisation d'outils (comme un éditeur UML, un IDE, etc.), la réalisation de tests, de jeux de démonstrations, etc. est fortement encouragée.

Ne perdez pas de vue que l'objectif est d'avoir des implémentations facilitées par l'analyse qui vous a conduit à concevoir votre framework. Ne vous lancez pas tout de suite dans l'écriture du code ! Prenez le temps de regarder plusieurs règles, de penser un peu à la façon dont ils se déroulent, et d'imaginer quelles sont les briques fonctionnelles qui pourraient être utilisées.

Il n'est pas demandé d'interface graphique (c'est même très fortement déconseillé), un bon affichage texte est largement suffisant. Cela ne signifie nullement qu'il ne faut pas employer un motif conceptuel de type MVC. L'accent doit être mis sur l'architecture logicielle, de sorte que le code soit aisé à lire et modifier ; ce point est très important car il permettra de vérifier vos compétences dans la maîtrise d'un développement orienté objet. Essayez naturellement d'illustrer les choses vues cette année.

Trois choses sont à réaliser et livrer sous forme d'une archive au format `tar` ou `zip` :

1. un répertoire avec vos sources sous la forme d'un code source en langage C++11 compilable *via* un `Makefile`<sup>1</sup>,
2. un diagramme UML décrivant l'architecture de classe de votre framework,
3. un rapport de quelques pages présentant ce qui est important,
4. un README facilitant l'installation, sans supposer que nous utilisons tel ou tel IDE.

Le travail est à réaliser en binôme et à déposer sur Moodle avant le 10 janvier 2021. Les soutenances auront lieu la semaine du 11 janvier.

---

1. Respecter impérativement cette consigne ! Vous pouvez utiliser n'importe quel outil pour développer mais la livraison doit contenir de quoi générer le projet avec la commande `make`. Une bonne façon de procéder est de simuler la livraison auprès de camarades lesquels devraient pouvoir générer les applications par simple appel à `make` (éventuellement en modifiant le nom ou l'emplacement du compilateur et des bibliothèques). Et si cela fonctionne, de livrer le tout comme il sera demandé ensuite...