

Mini-projet C++

ITII - 2014

Consignes générales

Ce projet est à réaliser de manière individuelle. Toute proximité trop importante dans le code entre 2 projets sera sanctionnée.

Le projet devra être rendu sous la forme d'une archive zippée que vous enverrez à l'adresse email suivante: alexis.cuglietta@gmail.com. Vous mettrez dans le titre du message le tag [UNO]. La date limite de rendu est fixée au 31 mai 2014 à 23:59. Les projets rendus hors délais ne seront pas acceptés.

Vous pourrez joindre à votre archive un fichier readme.pdf dans lequel vous ajouterez les précisions et commentaires que vous estimerez nécessaires. La qualité du code de votre projet sera évaluée et il vous est vivement recommandé de le commenter. Faites en sorte que les fonctions que vous implémentez soient courtes et lisibles.

Votre projet devra au moins compiler sous Linux avec GCC. Il vous est conseillé de réaliser un petit Makefile¹ ainsi que d'éventuels tests-unitaires pour prouver le bon fonctionnement de votre programme.

Un petit jeu de UNO

On se propose au cours de ce mini-projet de réaliser un jeu de UNO en mode console. Si vous ne le connaissez pas, vous trouverez une description du jeu et les principales règles sur la page Wikipédia associée: <http://fr.wikipedia.org/wiki/Uno>

Il vous est conseillé de réaliser ce projet en procédant de manière incrémentale, en respectant les étapes suivantes :

1. Création d'une classe Carte

Vous pourrez commencer par créer une classe qui représentera une carte de UNO. N'oubliez pas qu'il existe 2 types de cartes : les cartes normales (qui ont un simple numéro et une couleur) et les cartes spéciales qui ont un rôle particulier dans le jeu.

Vous réfléchirez à la manière la plus appropriée de représenter ces cartes en termes d'attributs de classe. Implémentez les différents constructeurs et une fonction permettant l'affichage comme `toString()` ou `operator<<()`².

2. Création d'une classe TasCarte

Nous allons également créer une classe représentant un tas de cartes. Cette classe permettra de gérer une liste de « Carte ». Vous implémenterez les méthodes permettant de :

- trier le tas de cartes
- mélanger le tas de cartes
- récupérer et supprimer la première carte
- ajouter une carte dans le tas
- dire si le tas contient encore des cartes
- afficher le contenu du tas de cartes
- vider le tas de cartes

¹ Tutoriel sur les Makefiles : <http://gl.developpez.com/tutoriel/outil/makefile/>

² Surcharge d'opérateurs de flux : <http://fr.openclassrooms.com/informatique/cours/programmez-avec-le-langage-c/les-operateurs-de-flux>

Indice : les méthodes `random_shuffle()` et `sort()` fournies par la librairie standard vous faciliteront le travail, mais ne fonctionnent pas avec tous les containers. La fonction `sort()` nécessite un `operator<` ou une fonction de comparaison.

3. Création d'une classe Joueur

Vous allez à présent créer une classe intitulée Joueur. Chaque joueur possède un nom et une liste de cartes³. Faites en sorte que l'on puisse afficher de manière lisible les cartes de chaque joueur.

4. Création d'une classe Jeu

Votre classe Jeu pourra contenir une liste de joueurs (le nombre de joueur varie entre 2 et 10 et sera demandé lors du début de partie). Cette classe contiendra également 2 tas de cartes : la pioche et le tas des cartes défaussées.

A la fin de la partie, votre jeu affichera le nom du gagnant et demandera si l'on souhaite jouer une autre partie. Dans un premier temps, il vous est conseillé de n'implémenter que les comportements des cartes normales. Quand ceci sera effectué vous pourrez ajouter les cartes spéciales du UNO (joker, +2, +4, changement de sens, etc.)

5. Bonus éventuels

Si vous arrivez à finir les points précédents et que votre jeu fonctionne de manière satisfaisante, vous pourrez y ajouter des fonctionnalités qui vous rapporteront des points supplémentaires.

Voici quelques suggestions:

- Création d'une intelligence artificielle permettant de jouer contre l'ordinateur
- Création d'une interface utilisateur graphique (avec une librairie comme GTK par exemple)
- Création d'un mode multi-joueurs en réseau

³ La librairie standard C++ propose une gestion des listes: <http://www.commentcamarche.net/faq/11255-introduction-a-la-stl-en-c-standard-template-library>