

# IHDCB131 Programmation 1

## Projet individuel : Marie aux Cartes

Cédric Libert & Gonzague Yernaux

Décembre 2020

### Introduction

Waw ! vous êtes arrivés au bout de la matière. Vous avez donc toutes les cartes en main pour mener à bien un projet de bonne envergure.

Justement, l'entreprise de création de jeux C-drague a entendu parler de vous. Cette entreprise, qui cherche à créer des jeux en C dans le but d'attirer un public féminin en fac info, est à la recherche d'une équipe dynamique qui saura utiliser au mieux ses atouts de programmation pour créer un petit jeu de Bataille Complexe. La Bataille Complexe, c'est un jeu de cartes qui ressemble à une Bataille, mais en plus... complexe.

### Objectifs et livrables

Concrètement, les objectifs de ce projet sont les suivants :

- consolider vos acquis en programmation sur un énoncé complexe. En particulier, il vous est demandé de faire usage d'un maximum de concepts vus au cours dans votre programme ;
- mettre en œuvre un aspect essentiel dans tout projet informatique : l'analyse du problème et la réflexion en amont, qui permettent d'aborder un problème complexe par raffinements successifs;
- être capable, étant donné un temps limité, de réaliser au moins un sous-ensemble fonctionnel d'un projet.

Nous vous demandons de poster sur WebCampus, d'ici la fin de la séance, le résultat de votre travail, sous la forme de deux fichiers :

- le code .c ;
- un bref document (PDF, Word ou équivalent), expliquant :
  - l'auto-évaluation : voir la section auto-évaluation, Merci de détailler votre auto-évaluation, il ne suffit pas de dire "ok" ou "pas ok" à côté de chaque critère !
  - jusqu'où vous êtes allés dans le projet, qu'est-ce que vous avez implémenté exactement;
  - ce qui, selon votre jugement, doit être expliqué à l'assistant. Par exemple : "j'ai respecté les consignes, sauf pour tel point car il me paraissait plus logique de faire autrement". Ou : "aie, je n'ai pas réussi à faire tel point, et donc j'ai fait de la façon suivante : ...".

### Énoncé

Voici la description du jeu de "Bataille Complexe" qu'il vous est demandé de réaliser. Il doit s'agir d'un programme C avec lequel on interagit en ligne de commande. D'abord, nous décrivons le jeu de base ; ensuite, nous exposons les différentes extensions/modes de jeu qui doivent venir le complexifier.

## Jeu de base

Le jeu se base sur le principe d'une Bataille :

- On commence par mélanger le paquet de cartes.
- Le paquet est distribué équitablement entre les deux joueurs.
- Chaque joueur tire la première carte de son paquet.
- C'est le joueur avec la carte la plus haute<sup>1</sup> qui remporte la manche. Il récupère les cartes qui ont été retournées ce tour-ci et les place sous son paquet de cartes.
- Dans le cas où les cartes tirées sont égales, les joueurs retournent à nouveau une carte, et le duel se fait entre les deux nouvelles cartes tirées. Jusqu'à ce qu'il y ait un gagnant pour cette manche... ou que la partie se termine, voir point suivant.
- Le jeu se poursuit de cette manière jusqu'à ce qu'un joueur n'ait plus de cartes ; il est alors perdant de la partie, à moins qu'au même moment, l'autre joueur n'ait plus de cartes non plus, auquel cas la partie se solde en une égalité.

Dans le jeu de base, il n'y a pas besoin d'interagir avec la console : les deux joueurs font leurs actions automatiquement. Autrement dit, lorsque l'on exécute votre programme, une partie complète doit se dérouler immédiatement et le résultat de chaque manche doit clairement être affiché à l'écran. On doit donc pouvoir retracer chaque coup effectué en lisant la console, et savoir quel joueur (joueur 1 ou joueur 2) a remporté la partie.

Ainsi, la console affichera des lignes du type:

```
Coup 4 | J1: as coeur      | J2: 2 pique      | Gagnant de la manche: J1 (+2 cartes)
Coup 5 | J1: 3 carreau     | J2: 3 trefle     | Bataille.
Coup 5 | J1: reine carreau | J2: valet carreau| Gagnant de la manche: J1 (+6 cartes)
Coup 6 | J1: 5 pique       | J2: roi trefle   | Gagnant de la manche: J1 (+2 cartes)
...
Gagnant: J1
```

Vous remarquez que, en cas de bataille, on considère qu'on reste dans le même coup, et le joueur gagnant remporte bien 6 cartes: les deux cartes de même valeur, les deux cartes "face cachée" et les deux nouvelles cartes retournées.

## Extensions

La Bataille, c'est chouette, mais ça manque de stratégie... et puis, regarder des ordinateurs jouer l'un contre l'autre, c'est un peu limité, non ? Pimentons tout cela ! Voici quelques extensions/améliorations du jeu qui permettront de le rendre plus intéressant.

1. Permettez que l'on puisse relancer une partie dès qu'une partie est finie, pour ne pas devoir relancer le programme.
2. Sélectionnez au hasard une couleur dominante à chaque carte tirée. Cette couleur donne un bonus de 1 point à toute carte de cette même couleur.
3. Ajoutez un mode de jeu "multicartes" dans lequel les joueurs ne tirent pas une, mais deux (si possible) cartes, et choisissent laquelle des deux jouer. Il ne s'agit donc plus d'un jeu automatique entre ordinateurs : il faut que les joueurs interagissent avec la console car ils doivent choisir une des deux cartes. La carte qu'ils ne choisissent pas est défaussée. Pour le choix des cartes, on commence par le joueur ayant perdu la dernière manche ou, à défaut, par le joueur 1.
4. Ajoutez un mode de jeu "atouts" qui peut être sélectionné au début d'une partie, et qui fonctionne comme suit :
  - chaque joueur, au départ, choisit une "couleur" (pique/trèfle/carreau/cœur) ;
  - lorsqu'un joueur remporte une manche avec une carte de sa couleur, il gagne un point ;

---

<sup>1</sup>L'as est considéré comme la plus forte de toutes !

- lorsqu'un joueur perd une manche alors qu'il avait joué une carte de sa couleur, il perd un point ;
  - en fin de partie, le joueur n'est plus le premier à avoir écoulé son tas de cartes... mais celui qui a le plus de points !
5. Ajoutez un *cheat code* au mode "multicartes" : lorsqu'un joueur doit choisir une carte, il peut entrer un code secret qui lui permet de mélanger son paquet de cartes et d'en tirer deux nouvelles.
  6. À chaque nouvelle partie, demandez à l'utilisateur combien de joueurs vont s'affronter. Clarifiez vous-mêmes les règles pour la Bataille à plus de deux joueurs (comment répartir les cartes ? combien de joueurs maximum ?), et documentez vos choix dans votre rapport.
  7. D'autres idées d'amélioration ? Allez-y !

Les extensions peuvent être implémentées de différentes manières, évidemment, et avec quelques nuances de fonctionnement. Dans le rapport, **détaillez vos choix** et expliquez comment vous avez levé les ambiguïtés éventuelles de l'énoncé.

## Auto-évaluation

Prenez connaissance des critères de cette auto-évaluation avant la réalisation du projet : ils vous guideront. Pour chacun des critères, indiquez dans votre rapport s'il est rempli ou non. Le cas échéant, expliquez pourquoi le critère n'est pas rempli.

- Votre programme compile sans avertissement (warning).
- Votre programme s'exécute puis s'arrête (aucune boucle infinie).
- Votre code est lisible et correctement indenté.
- Votre programme fait usage de sélections, itérations, tableaux, structures de données.
- Votre programme est organisé en différentes fonctions.
- Vos fonctions sont spécifiées (préconditions/postconditions).
- Les identifiants de variables et de fonctions sont explicites.
- Le programme mélange bien le paquet de cartes avant chaque partie.
- Une manche du mode "de base" se déroule bien selon les règles de la Bataille.
- Le programme affiche des informations qui permettent de bien s'y retrouver dans le jeu.
- Le programme implémente correctement l'extension 1.
- Le programme implémente correctement l'extension 2.
- Le programme implémente correctement l'extension 3.
- Le programme implémente correctement l'extension 4.
- Le programme implémente correctement l'extension 5.
- Vous avez testé, avec succès, votre programme, sur plusieurs cas de test.

## Une dernière remarque

Hmmm... Le projet est sympa, mais vous n'en voyez pas le bout. Vous avez peur de ne pas arriver à tout faire dans le temps imparti ? **Ce n'est pas grave : allez le plus loin possible !** Vous ne serez pas simplement évalués sur le résultat final, mais aussi (et surtout) sur vos talents de programmeurs (et d'équipe !). Si vous n'arrivez pas au bout de projet, arrangez-vous tout de même pour que votre code soit exécutable, c'est-à-dire qu'il compile et affiche un résultat, même s'il s'agit d'un résultat intermédiaire. Expliquez alors dans le rapport pourquoi, à votre avis, le temps vous a manqué.

Bonne chance !