

INF2-1-B, 1^{er} mars 2019

Labo 3 : Jeu des 7 familles

But

Ce projet vise à construire un système pouvant jouer au jeu des 7 familles. Le système sera implémenté sous la forme d'une classe « Joueur », accompagnée d'une classe « Partie » qui contrôle le jeu. On souhaite faire jouer plusieurs joueurs entre eux et observer le déroulement de la partie. On *ne* souhaite *pas* qu'un humain puisse jouer contre le système, et on *ne* cherche *pas* les meilleures stratégies.



Source : <https://dessinemoiunehistoire.net/>

Règles du jeu

Le jeu des 7 familles utilise un jeu de cartes constitué de groupes de cartes appelés familles. Le but de chaque joueur est d'avoir à la fin d'une partie un maximum de familles.

Les objets représentés sur les cartes varient d'un jeu à l'autre. À l'origine, une famille était constituée de six cartes (aïeule, aïeul, mère, père, fille, fils), et chaque famille était distinguée par une couleur.

Dans ce projet, on représente les familles par des nombres de 1 à NOMBRE_FAMILLES et chaque membre par une lettre : A, B, C, ... avec un nombre paramétrable de CARTES_PAR_FAMILLE. Deux autres paramètres sont le NOMBRE_JOUEURS et la donne initiale, en CARTES_PAR_JOUEUR.

Par exemple, on peut avoir un jeu avec 9 familles de 4 cartes chacune, auquel jouent 4 joueurs, et initialement on distribue 5 cartes à chacun. Les cartes non distribuées sont posées sur la table et forment la pioche (16 cartes restantes dans cet exemple).

Déroulement :

- Tour-à-tour, chaque joueur choisit librement un autre joueur, et lui demande un membre précis d'une famille, p.ex. : « Dans la famille 3, je demande le B ». Celui qui demande doit avoir en main au moins un membre de cette famille pour pouvoir demander un autre.
- Si celui à qui on demande possède la carte demandée, il la donne au premier, qui peut refaire une demande (nécessairement au même joueur). S'il ne l'a pas, le premier prend une carte dans la pioche (s'il en reste), et le tour passe au joueur suivant.
- Lorsqu'un joueur a tous les membres d'une famille, il pose la famille devant lui et la conserve.
- Le jeu finit lorsque plus personne n'a de cartes en main et que la pioche est vide.
- À la fin, chaque joueur gagne autant de points que de familles posées devant lui.
- Après plusieurs parties, le gagnant est celui qui a le plus grand total de points.

Travail demandé

Le projet simule plusieurs joueurs (instances de la classe `Joueur`) qui s'affrontent dans une partie (instance de `Partie`). Le système distribue les cartes (instances de la classe `Carte`) au hasard, puis tour-à-tour chaque joueur demande des cartes à un autre, selon les règles données mais sans stratégie intelligente. À chaque fois qu'un joueur reçoit une carte, il vérifie s'il a complété une famille, auquel cas il la pose devant lui. On affichera les actions selon le modèle fourni en annexe, qui doit être respecté.

À la fin de la partie, chaque joueur reçoit autant de points qu'il a de familles. Le système (dans le `main()`) fera jouer 100 parties avec les paramètres ci-dessus, et indiquera le total obtenu par chaque joueur, divisé par le `NOMBRE_FAMILLES` (pour normaliser le score en %).

Bonus. Copier la classe `Joueur` sous le nom de `MeilleurJoueur` (ou faire une classe dérivée si on peut) puis améliorer la stratégie du `MeilleurJoueur`. P.ex., un tel joueur demande d'abord les membres des familles dont il a déjà un maximum de cartes. Refaire jouer 100 parties avec un joueur de cette classe et trois de l'autre, et observer si son score final est meilleur que celui des autres.

Projet à rendre

Le projet est à effectuer par groupes de trois. Il doit être rendu avant le **vendredi 15 mars à 23h59**.

Merci de remettre sur Cyberlearn un fichier ZIP appelé `Labo3_Nom1_Nom2_Nom3.zip` contenant :

- Le dossier du projet sous NetBeans (mais sans les dossiers 'build' et 'dist'). Le code source doit être commenté selon les indications données au début d'INF2.
- Un rapport (PDF, 1-2 pages) présentant les choix d'implémentation et les difficultés résolues.
- Un exemple d'exécution (fichier TXT) avec les énoncés des actions pour une (seule) partie, ainsi que les scores finaux des joueurs après 100 parties (avec les paramètres de la page 1).

Contraintes et suggestions

- Utiliser trois classes : `Partie`, `Joueur`, `Carte` et séparer les fichiers `.h` et `.cpp`
- Une `Carte` est définie par la famille et le membre, qui sont à afficher comme un nombre suivi d'une lettre, mais peuvent être codés tous deux par des entiers (`unsigned short`)
- Un `Joueur` a un nom et deux listes de cartes (p.ex. `std::vector<...>`) : les `cartesEnMain` et les `famillesSurTable`
- Un `Joueur` doit pouvoir décider quelle carte demander (fonction membre `demanderCarte`) en fonction des cartes qu'il a en main. Stratégie basique : compléter une famille au hasard ; ou, pour la question bonus, la famille dont on a déjà le plus de membres. *On ne demande pas de mémoriser les échanges précédents.*
- Un `Joueur` a une fonction membre `detecterFamille` qui cherche dans les `cartesEnMain` si une famille est complète, et la pose dans les `famillesSurTable`
- La `Partie` gère la pioche de cartes et organise le jeu. Elle donne le tour à chaque joueur, et tire au hasard le joueur à qui on demande des cartes (à condition qu'il en ait en main). Elle détecte aussi la fin de la partie, quand aucun joueur n'a plus de cartes en main et la pioche est vide.
- On contrôle depuis le `main()` le lancement de plusieurs parties successives et la gestion des scores totaux des joueurs, retournés par l'instance de `Partie` après chaque partie.

Exemple d'affichage d'une partie

Début de la partie de 7 familles

*** Tour 1 ***

Alice : 2C 9A 1D 4D 7B []
Bobby : 2D 3D 4C 7A 8C []
Carol : 4B 1B 9D 8A 5C []
Danny : 1A 2B 3B 2A 5A []
Pioche: 5D 6A 7C 6B 9B 6D 1C 3A 8D 4A 9C 3C
6C 5B 8B 7D

Alice demande à Bobby la carte 2A
mais Bobby ne l'a pas
Alice prend une carte dans la pioche (7D)
Bobby demande à Alice la carte 2A
mais Alice ne l'a pas
Bobby prend une carte dans la pioche (8B)
Carol demande à Alice la carte 4A

mais Alice ne l'a pas
Carol prend une carte dans la pioche (5B)
Danny demande à Bobby la carte 1B
mais Bobby ne l'a pas
Danny prend une carte dans la pioche (6C)

*** Tour 2 ***

Alice : 2C 9A 1D 4D 7B 7D []
Bobby : 2D 3D 4C 7A 8C 8B []
Carol : 4B 1B 9D 8A 5C 5B []
Danny : 1A 2B 3B 2A 5A 6C []
Pioche: 5D 6A 7C 6B 9B 6D 1C 3A 8D 4A 9C 3C

Alice demande à Bobby la carte 2A
mais Bobby ne l'a pas
Alice prend une carte dans la pioche (3C)
Bobby demande à Alice la carte 2A
mais Alice ne l'a pas
Bobby prend une carte dans la pioche (9C)
Carol demande à Bobby la carte 4A
mais Bobby ne l'a pas
Carol prend une carte dans la pioche (4A)
Danny demande à Bobby la carte 1B
mais Bobby ne l'a pas
Danny prend une carte dans la pioche (8D)

*** Tour 3 ***

Alice : 2C 9A 1D 4D 7B 7D 3C []
Bobby : 2D 3D 4C 7A 8C 8B 9C []
Carol : 4B 1B 9D 8A 5C 5B 4A []
Danny : 1A 2B 3B 2A 5A 6C 8D []
Pioche: 5D 6A 7C 6B 9B 6D 1C 3A
Alice demande à Bobby la carte 2A
mais Bobby ne l'a pas
Alice prend une carte dans la pioche (3A)
Bobby demande à Danny la carte 2A
et Danny donne la carte à Bobby
Bobby demande à Danny la carte 2B
et Danny donne la carte à Bobby
Bobby demande à Danny la carte 2C
mais Danny ne l'a pas
Bobby prend une carte dans la pioche (1C)
Carol demande à Danny la carte 4C
mais Danny ne l'a pas
Carol prend une carte dans la pioche (6D)

Danny demande à Carol la carte 1B
et Carol donne la carte à Danny
Danny demande à Carol la carte 1C
mais Carol ne l'a pas
Danny prend une carte dans la pioche (9B)
*** Tour 4 ***
Alice : 2C 9A 1D 4D 7B 7D 3C 3A []
Bobby : 2D 3D 4C 7A 8C 8B 9C 2A 2B 1C []
Carol : 4B 9D 8A 5C 5B 4A 6D []
Danny : 1A 3B 5A 6C 8D 1B 9B []
Pioche: 5D 6A 7C 6B

...

*** Tour 15 ***

Alice : [2C.2A.2B.2D.]
Bobby : 9C
[3D.3A.3B.3C.7A.7C.7B.7D.8C.8B.8A.8D.]
Carol : 9D 9A 9B [4B.4A.4C.4D.]
Danny :
[1A.1B.1C.1D.5A.5D.5B.5C.6C.6A.6B.6D.]
Pioche:
Bobby demande à Carol la carte 9A
et Carol donne la carte à Bobby
Bobby demande à Carol la carte 9B
et Carol donne la carte à Bobby
Bobby demande à Carol la carte 9D
et Carol donne la carte à Bobby
Bobby demande à Carol la carte 9A
mais Carol ne l'a pas
Alice : [2C.2A.2B.2D.]
Bobby : [3D.3A.3B.3C.7A.7C.7B.7D.
8C.8B.8A.8D.9C.9A.9B.9D.]
Carol : [4B.4A.4C.4D.]
Danny : [1A.1B.1C.1D.5A.5D.5B.5C.
6C.6A.6B.6D.]
Pioche:

La partie est finie !

Nombre de tours : 15

RUN RÉUSSI (temps total: 156ms)