

## INFO-H-100 : Introduction à la programmation Projet 1 : Rami simplifié

Le but du projet est d'implémenter en *Python 2.7* une version simplifiée, à un joueur, du Rami<sup>1</sup>. Le jeu se déroule de la manière suivante :

- Le moteur de jeu mélange deux jeux de cartes classiques de 52 cartes (donc sans Joker) pour créer la pile appelée *pioche*.
- Le joueur tire 10 cartes dans la pioche.
- A chaque tour, tant que le joueur a des cartes en main :
  1. Le joueur pioche une carte.
  2. Le joueur décide quelles cartes il veut poser pour former une (et une seule) combinaison. Il peut choisir de ne pas déposer de cartes.
  3. S'il lui reste des cartes en main, le joueur doit se défausser d'une carte de son choix.
- Quand le joueur n'a plus de cartes en main, le programme annonce que le jeu est fini, puis se termine.

Les combinaisons que le joueur a le droit de poser sont :

**Les paires** Une paire est une combinaison de deux cartes de même valeur, quelle que soit leur couleur<sup>2</sup>.

**Les brelans** Un brelan est une combinaison de trois cartes de même valeur, quelle que soit leur couleur.

**Les carrés** Un carré est une combinaison de quatre cartes de même valeur, quelle que soit leur couleur.

**Les tierces** Une tierce est une combinaison de trois cartes de la même couleur dont les valeurs se suivent.

Pour les tierces, on considère que l'as est avant le 2, et que les valet, dame et roi sont après le 10, dans cet ordre-là.

Si le joueur se trompe (c'est à dire qu'il dépose des cartes qui ne forment pas une combinaison), l'ordinateur lui propose de recommencer.

```
Roi Carreau  Roi Coeur  7 Carreau  9 Pique  9 Trefle  10 Trefle  2 Coeur  5 Pique  7 Trefle  6 Pique
  1           2           3           4           5           6           7           8           9           10
```

```
Numéro des cartes à poser (1->10)
Choisissez une carte (-1 pour terminer) 1
Choisissez une carte (-1 pour terminer) 2
Choisissez une carte (-1 pour terminer) -1
```

```
Vous déposez:  Roi Coeur  Roi Carreau
```

```
 7 Carreau  9 Pique  9 Trefle  10 Trefle  2 Coeur  5 Pique  7 Trefle  6 Pique
  1           2           3           4           5           6           7           8
```

```
Numéro de la carte à rejeter (1->8): 4
```

```
 7 Carreau  9 Pique  9 Trefle  2 Coeur  5 Pique  7 Trefle  6 Pique  8 Pique
  1           2           3           4           5           6           7           8
```

```
Numéro des cartes à poser (1->8)
Choisissez une carte (-1 pour terminer) 2
Choisissez une carte (-1 pour terminer) 3
Choisissez une carte (-1 pour terminer) -1
```

```
Vous déposez:  9 Trefle  9 Pique
```

```
 7 Carreau  2 Coeur  5 Pique  7 Trefle  6 Pique  8 Pique
  1           2           3           4           5           6
```

```
Numéro de la carte à rejeter (1->6): 2
```

```
 7 Carreau  5 Pique  7 Trefle  6 Pique  8 Pique  9 Carreau
  1           2           3           4           5           6
```

```
Numéro des cartes à poser (1->6)
Choisissez une carte (-1 pour terminer) 1
Choisissez une carte (-1 pour terminer) 3
Choisissez une carte (-1 pour terminer) -1
```

```
Vous déposez:  7 Trefle  7 Carreau
```

```
 5 Pique  6 Pique  8 Pique  9 Carreau
  1           2           3           4
```

```
Numéro de la carte à rejeter (1->4): 4
```

```
 5 Pique  6 Pique  8 Pique  10 Carreau
  1           2           3           4
```

1. Voir <http://fr.wikipedia.org/wiki/Rami> pour les règles complètes, *qu'on ne vous demande pas d'implémenter*.  
2. Les couleurs (ou *enseignes*) sont coeur, pique, carreau ou trèfle.

On vous demande de découper votre programme en fonctions dont voici quelques prototypes à implémenter obligatoirement :

- `rami()` qui lance le jeu et effectue une partie.
- `creer_jeu()` qui crée un jeu de 52 cartes et le renvoie.
- `melange(pioche)` qui mélange la pioche.
- `tire(pioche)` qui tire une carte dans la pioche et la renvoie.
- `distribue(pioche)` qui tire 10 cartes et renvoie la main de l'utilisateur.
- `defausse(main)` qui demande à l'utilisateur de se défausser d'une carte et l'enlève de sa main.
- `choix_depose(main)` qui demande à l'utilisateur son choix de cartes.
- `paire(combinaison)` qui vérifie si la combinaison est une paire.
- `brelan(combinaison)` qui vérifie si la combinaison est un brelan.
- `carre(combinaison)` qui vérifie si la combinaison est un carre.
- `tierce(combinaison)` qui vérifie si la combinaison est une tierce.
- `valide(combinaison)` qui vérifie si une combinaison de cartes est valide.

Vous pouvez également définir d'autres fonctions si vous estimez que cela simplifie ou clarifie votre code.

## Consignes

Le projet est à réaliser **seul** et à remettre **au plus tard le 12 décembre 2011 à 12h30** au UB4.131 (à côté de la salle informatique Socrate). Nous vous demandons de vous limiter **uniquement** à la matière vue au cours des 7 premiers TPs.

Vous devez rendre un rapport dactylographié de l'ordre de 5 ou 6 pages comprenant :

1. une introduction présentant succinctement les différentes fonctions de votre programme,
2. une description de la façon dont vous représentez respectivement les cartes et les jeux de cartes ainsi qu'une description des opérations sur les jeux de cartes,
3. une explication de l'algorithme utilisé pour mélanger les cartes,
4. une capture d'écran présentant une exécution de votre programme et
5. votre code complet intelligemment commenté.

Vous devez également remettre votre code de façon électronique sur un serveur Mercurial se trouvant à l'adresse <http://informa2.ulb.ac.be>. L'utilisation de ce système sera détaillée sur la page web des projets.

Les critères d'évaluation sont les suivants :

- Le code : **résout exactement le problème demandé sans ajout ni apport personnel**, qualité de l'algorithme, qualité et pertinence de la découpe en fonction, syntaxe correcte, parfaitement indenté, les variables bien nommées, le minimum de commentaires utiles, respect des conventions et des règles de bonne pratique vues au cours, aux séances d'exercices et résumées sur la page web des projets<sup>3</sup>.
- Le rapport : qualité du contenu, présentation, français correct.
- Le respect des consignes.

Les élèves assistants sont disponibles à 12h30 dans la salle Socrate afin de répondre à vos questions. Consultez leur horaire sur la page web des TPs.

---

3. <http://cs.ulb.ac.be/public/teaching/infoh100/projets>