

---

## USRS45 - TP

*Réalisation d'une version C# du jeu du Black Jack*

Octobre 2020

---

### Énoncé :

Le but de ce TP est de développer étape par étape **une version C# du jeu du Black Jack**. Prenez bien le temps de lire l'ensemble du sujet avant de commencer. Toutes les étapes du jeu de base sont guidées. Une fois celui-ci fonctionnel, des améliorations non guidées vous sont demandées.

### Le jeu de base :

1. Créer un nouveau projet et sauvegarder-le.

#### Initialisation :

2. Créer une variable **valeurCartes** qui contient un dictionnaire de cartes possibles associées à leur nombre de points.

Indication :

```
Dictionary<"string","int"> dict = new Dictionary<"string","int">();  
dict.Add("1",1);  
...  
dict.Add("10",10);  
dict.Add("V",10);  
dict.Add("D",10);  
dict.Add("R",10);
```

3. Créer une variable **joueurH** contenant une liste vide ainsi qu'une variable **joueurO** contenant également une liste vide.

*Ces variables contiendront les cartes piochées par l'humain (H) et par l'ordinateur (O).*

#### Générer le paquet de cartes :

4. Créer une variable **paquet** initialisée grâce aux clés du dictionnaire **valeurCartes**. La variable **paquet** correspond à une liste qui contient 2 paquets de cartes complets, c'est-à-dire de l'as au roi dans chaque couleur (4 couleurs), deux fois.

5. Mélanger les valeurs de la variable **paquet**.

Indication :

```
var ListShuffled = myList.OrderBy(x => Guid.NewGuid()).ToList();
```

#### Distribution des cartes :

6. Faire une **boucle** qui se répète deux fois et qui contient les instructions suivantes :
  - a. Ajouter à la fin de **joueurH** la dernière carte de **paquet** et retirer cette carte de **paquet**.
  - b. Ajouter à la fin de **joueurO** la dernière carte de **paquet** et retirer cette carte de **paquet**.

#### Affichage du jeu :

7. **Afficher** "Humain : X" avec X les cartes piochées par l'humain.
8. **Afficher** "Ordinateur : ? X" avec X la deuxième carte piochée par l'ordinateur, la première restant secrète.

#### Déroulement d'un tour :

9. Créer une variable **stopJoueur** initialisée à **false**, qui indique si le joueur souhaite arrêter de piocher des cartes.
10. Créer une variable **stopOrdinateur** initialisée à **false**, qui indique si l'ordinateur souhaite arrêter de piocher des cartes.
11. Créer une variable **finPartie** initialisée à **false**, qui indique si la partie est terminée.

#### Décision du joueur :

12. **Si stopJoueur est à false alors**
  - a. **Afficher** "Voulez-vous piocher une nouvelle carte?"
  - b. **Afficher** "o - Oui"
  - c. **Afficher** "n - Non"
  - d. **Demander** à l'utilisateur de répondre et **enregistrer** sa réponse dans une variable **choixJoueur**.
  - e. **Si choixJoueur est égal à "o" alors**
    1. **Afficher** "Humain : Je pioche."
    2. Ajouter la dernière carte de **paquet** à la liste **joueurH**.
  - f. **Sinon**
    1. **Afficher** "Humain : Je m'arrête là."
    2. **Assigner true** à la variable **stopJoueur**.

#### Décision de l'ordinateur :

13. **Si** `stopOrdinateur` est à `false` **alors**
  - a. **Si** le score actuel donné par la somme des cartes de l'ordinateur est inférieur ou égal à 15 **alors**
    1. **Afficher** "Ordinateur : Je pioche."
    2. Ajouter la dernière carte de `paquet` à la liste `joueurO`.
  - b. **Sinon**
    1. **Afficher** "Ordinateur : Je m'arrête là."
    2. Assigner `true` à la variable `stopOrdinateur`.

#### Affichage du jeu :

14. **Afficher** de nouveau le jeu suivant les indications des questions 7 et 8.

#### Fin de partie :

15. **Si** la variable `stopOrdinateur` et la variable `stopJoueur` sont égales à `true` en même temps **alors assigner** `true` à la variable `finPartie`.
16. **Si** le score du joueur Humain est supérieur ou égal à 21 **ou** que le score de l'ordinateur est supérieur ou égal à 21 **alors assigner** `true` à la variable `finPartie`.

#### Mise en place d'une boucle :

17. Pour permettre de jouer une partie entière, il faut pouvoir jouer plusieurs tours. Trouver le bon emplacement pour mettre en place une **boucle**, qui continue **tant que** la variable `finPartie` est égale à `false`.

### Améliorations :

Les étapes suivantes ne sont plus guidées, elles ne doivent être abordées qu'une fois le jeu de base entièrement fonctionnel.

1. Permettre à l'utilisateur de choisir le nombre de paquets de cartes.
2. Permettre à l'utilisateur de saisir son prénom et remplacer "Humain" par son prénom dans les affichages.
3. Mettre en place une fonction "Score" qui retourne le score d'un joueur, humain ou ordinateur, en fonction des cartes qu'il a pioché. Modifier les affichages du jeu pour voir "(Y pts) Humain : X" avec X toujours les cartes piochées par l'humain mais en rajoutant Y la somme des points pour les cartes déjà piochées.
4. Mettre en place un affichage spécifique en fin de partie qui, en fonction des cas : félicite le joueur lorsqu'il a gagné, lorsqu'il a un black jack, qui lui annonce lorsqu'il a perdu parce qu'il a dépassé 21 ou que l'ordinateur a eu un black jack...
5. Ajouter la gestion des cartes As qui peuvent valoir soit 1 soit 11.