

Rapport projet UAA9

Projet BlackJack

Contents

Introduction :.....	2
Présentation du projet	2
Difficultés :.....	2
1 ^{ière} difficulté	2
2 ^{ème} difficulté	3
3 ^{ème} difficulté	3
4 ^{ème} difficulté	3

Introduction :

Bienvenue dans ce rapport de projet ayant pour sujet la création d'un jeu de blackjack en langage de programmation C#.

Dans ce rapport, je ferai une brève présentation du projet. J'expliquerai les difficultés que j'ai rencontré à travers ce parcours,

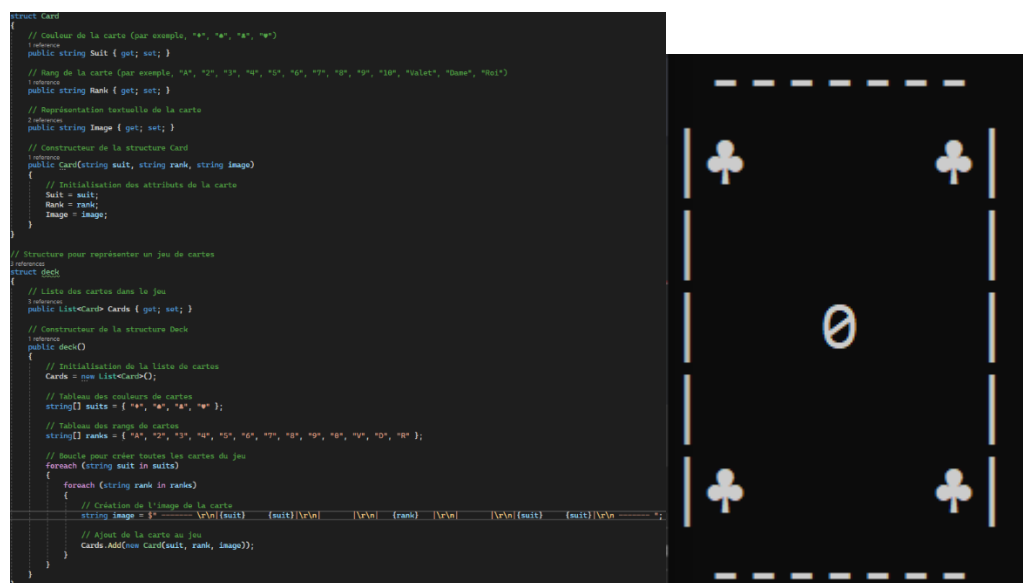
Présentation du projet

Le but de ce projet était de faire une reproduction d'une expérience de casino à une table de blackJack. Des joueurs peuvent se mettre à une table avec une mise de départ et jouer contre le croupier.

Difficultés :

1^{ière} difficulté

Je me suis mis pour objectif de faire en sorte que l'on puisse avoir une image des cartes dans la console et ce, d'une manière où je ne devais pas concaténer à chaque fois l'image de ma carte. Il a donc fallu que je trouve un moyen de générer automatiquement l'image de toutes les cartes d'un jeu de carte classique. J'ai ensuite procédé à faire en sorte que l'on puisse faire afficher ces cartes en choisissant l'option de test au lancement du programme. Le résultat fut cette structure et qui donne cette image :



2^{ème} difficulté

La deuxième difficulté que j'ai rencontré fut de mettre plusieurs cartes côte à côte. Pour ça, j'ai fait usage de la fonction `split()` qui permet de séparer une chaîne de caractère à chaque fois qu'il y a la présence du paramètre spécifié. Grâce à cette fonction, j'ai pu séparer des images chaque lignes à imprimer et fait en sorte que les cartes soient affichées collées entres elles



3^{ème} difficulté

Il se peut que lors de la création du projet, j'ai oublié que le BlackJack se jouait à plusieurs personnes. Le problème fut que j'ai réalisé ça quand j'avais déjà une version 1 joueur du jeu. Cela m'a forcé à devoir changer entièrement ma logique de jeu. Il a fallut que je fasse un nouveau système pour garder en mémoire les mains des joueurs. Après ça, il a fallut que je convertisse mon code pour supporter les différentes mains et j'ai finis par pouvoir faire en sorte que chaque joueurs puisses jouer au moment de leur tour dédié.

4^{ème} difficulté

Il faut savoir que dans mon code, je compte les points de chaque joueurs en début de tours pour savoir si ils peuvent jouer ou non en fonction des règles du jeu. Au moment où j'ai modifier le code pour accepter plusieurs joueurs, je me suis retrouvé fasse à un problème qui m'a pris plusieurs jours à comprendre. Les points étaient parfois mal calculé. Il se trouvait que le problème fut lié à la fonction qui déterminait si un Ace valait 1 point ou 11. Il y avait juste un mélange de pointeurs de boucles `for()` qui permettaient de naviguer dans mes tableaux

Références bibliographiques :

ChatGPT/Aria

J'ai eu une grande aide de la part de l'intelligence artificielle notamment pour me réexpliquer comment créer une structure et pour la découverte de la fonction `split()`.

Pour le reste, j'avoue que j'ai réussi à m'en sortir seul à l'aide de mes connaissances sur la pratique du `Console.ReadKey()` pour la vérification de mes entrées ou diverses pratiques qui me sont propres.

Annexe :

Github : https://github.com/Solune-Petit/2024-2025/tree/main/projets_fin_annee/CSharp/5T24_PetitSolune_BlackJack

GithubProject : <https://github.com/users/Solune-Petit/projects/7>