



Rapport de Projet

INFORMATIQUE - Projet Ascenseur

L'ASCENSEUR

Gauthier CUSSONNEAU
Axel ROYER
Nicolas TIREL

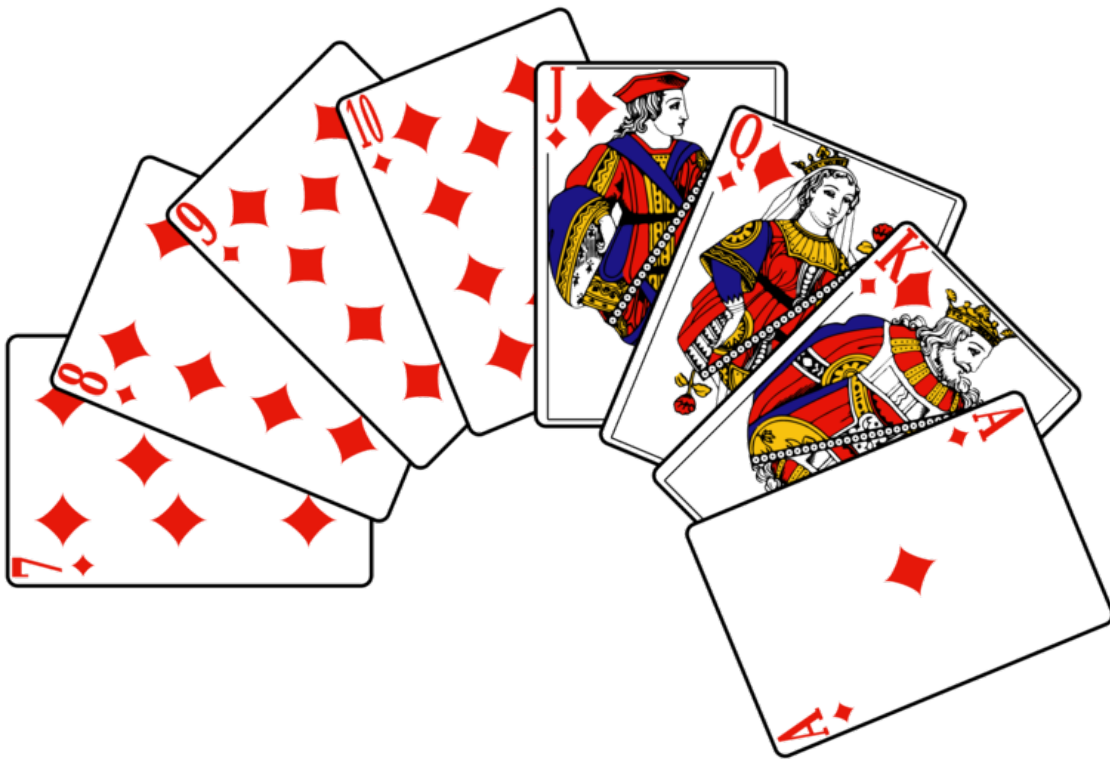
2017 — 2018

Table des matières

1	Récapitulatif du problème demandé	2
2	Description	3
2.1	Du programme	3
2.2	De ses fonctionnalités	3
2.3	De son utilisation	4
3	Choix durant l'implémentation	5
4	Les avantages et les limites	5
4.1	Les avantages	5
4.2	Les limites	6
5	Contribution de chaque élève au projet	6
6	Bilan de chaque élève à propos du projet	6
6.1	Nicolas	6
6.2	Axel	6
6.3	Gauthier	6

1 Récapitulatif du problème demandé

Lors de ce premier projet informatique de l'année il nous a été demandé de réaliser le **Jeu de l'Ascenseur**. Afin de mieux comprendre le projet nous allons donc d'abord expliquer brièvement en quoi consiste l'Ascenseur. Le jeu se joue de 2 à 5 joueurs et avec un paquet de 52 cartes. Il est composé d'une phase ascendante et d'une phase descendante. Au début du jeu, le paquet de cartes est divisé par le nombre de joueurs afin de déterminer s'il y aura ou non un atout et quel sera le nombre maximal de carte que les joueurs pourront avoir en main. Ensuite chaque joueur annonce combien de plis il espère remporter durant la manche, que l'on appelle le contrat (un pli remporté s'apparente à une victoire du joueur durant le tour, la somme totale des plis annoncés par les joueurs ne devant être égale au nombre de cartes distribuées). Débute ensuite la partie, tour à tour chaque joueur doit jouer une carte de la même couleur que la carte jouée par le premier joueur et si possible supérieure à celles posées par les autres joueurs avant afin de remporter le pli. Il faut bien évidemment prendre en compte l'existence de l'atout, un joueur qui possède une carte de la couleur de l'atout (défini au début de chaque manche) peut la jouer et ainsi surpasser toutes les autres cartes (exceptées les cartes atout de valeur supérieure) et donc remporter le pli. Chaque tour se termine donc quand chaque joueur a joué une carte et chaque pli remporté octroie un point, et chaque gagne ou perd des points en fonction du fait qu'ils aient rempli ou non leur contrat. Le jeu se compose donc de deux phases, décomposée elles-mêmes en manches elles-mêmes décomposées en tours : par exemple lors de la première manche il n'y a qu'un seul tour car chaque joueur n'a qu'une seule carte en main, mais lors de la seconde manche il y aura deux tours car un tour est terminé lorsque chaque joueur a joué une seule carte. La phase ascendante se termine lorsque toutes les cartes auront été distribuées et les joueurs commencent alors la phase descendante dans le sens inverse c'est à dire par distribuer le plus grand nombre de cartes pour arriver à une carte par joueur au dernier tour. A chaque manche les points sont comptabilisés, le joueur ayant le plus de points à la fin de la partie remporte la victoire. Il nous est donc demandé de coder entièrement ce jeu et de réaliser une IA qui serait alors capable de jouer contre un joueur physique ou virtuel à ce jeu.



2 Description

2.1 Du programme

Le programme est composé d'un premier menu appelé au démarrage avec un sous-menu "Jouer". Lorsque l'utilisateur entre dans le menu "Jouer", quatre options lui sont proposées. La première, "Nouvelle partie" permet de démarrer une partie à zéro en prenant en compte les options si elles ont été définies auparavant. La deuxième, "Continuer la partie" permet de poursuivre une partie sauvegardée pendant le jeu ou une partie initialisée par un fichier de sauvegarde qui doit être préférablement placé dans le même dossier que le jeu, dans le cas contraire l'utilisateur devra indiquer son chemin d'accès. La troisième permet de configurer les options du jeu et enfin la quatrième option permet de quitter le jeu. Les différentes options de jeu configurables sont : le nombre de points gagnés pour les joueurs ayant rempli leur contrat, le nombre de points par pli gagné dans le cas où le joueur a rempli son contrat, le nombre de point perdu pour les joueurs n'ayant pas rempli leur contrat et enfin le nombre de points par pli perdu. Durant le jeu, l'utilisateur peut à tout moment sauvegarder la partie dès que le programme le lui permet. Nous avons décidé de coder le programme de telle sorte qu'il demande régulièrement à l'utilisateur s'il souhaite sauvegarder sa progression en temps réel. En indiquant "oui" au programme, l'utilisateur aura alors à donner le nom du fichier dans lequel la partie sera sauvegardée en sachant que si le fichier existe déjà, il sera écrasé. Afin de faciliter l'évolution du jeu, si l'utilisateur indique n'importe quoi d'autre que "oui", le jeu continue sans sauvegarder.



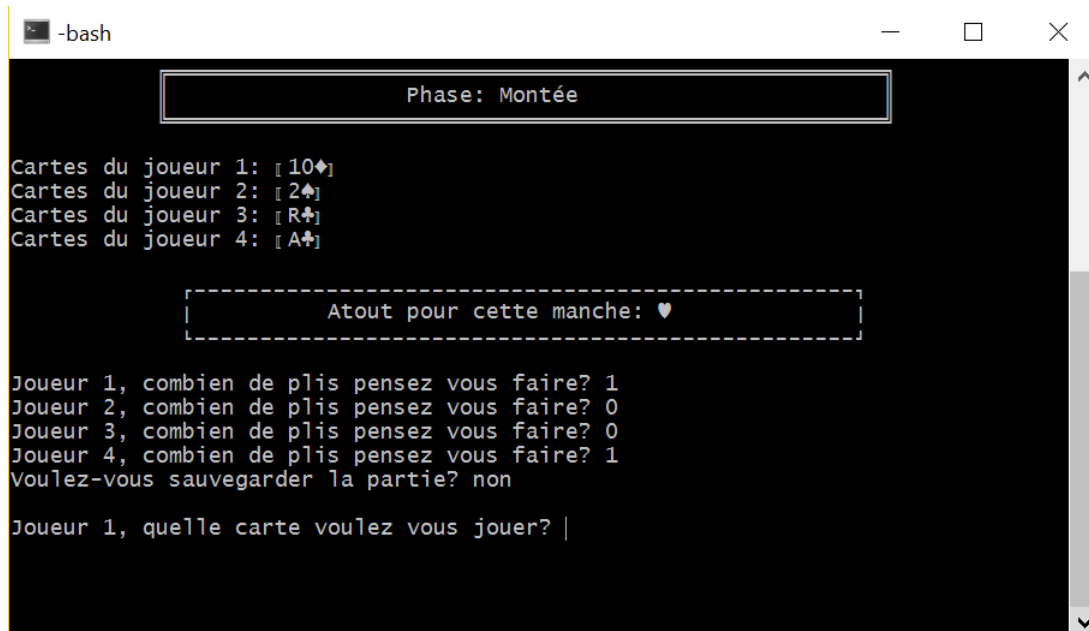
2.2 De ses fonctionnalités

En plus de permettre à l'utilisateur de jouer au jeu de l'Ascenseur, notre programme offre plusieurs fonctionnalités supplémentaires :

- La sauvegarde : l'utilisateur peut, après avoir donné les contrats de tous les joueurs, sauvegarder la partie dans un fichier externe afin de pouvoir la relancer plus tard via le menu principal.
- L'interface graphique : tout au long du jeu, l'utilisateur sera accompagné par une interface soignée au mieux afin de donner une esthétique travaillée au jeu et rendre l'expérience belle et agréable pour l'utilisateur.

2.3 De son utilisation

Lors du lancement du programme, l'utilisateur est accueilli par un écran de démarrage présentant l'option de jouer au jeu. L'utilisateur doit ensuite appuyer sur la touche "Entrée" afin de passer au menu principal. Afin de se diriger dans les différentes options du menu, l'utilisateur doit taper le numéro correspondant au choix de l'option et valider sa saisie par la touche "Entrée". Il peut ainsi lancer une nouvelle partie, lancer une partie depuis un point de sauvegarde, modifier les options du jeu pour une nouvelle partie ou tout simplement quitter le jeu. Si l'utilisateur souhaite continuer une partie qu'il a enregistré auparavant, il doit alors indiquer le nom du fichier de sauvegarde et veiller à indiquer si besoin le chemin d'accès du fichier, ne pas oublier l'extension du fichier et enfin lancer un fichier de sauvegarde respectant la norme indiquée dans le cahier des charges, sans quoi le programme risque de ne pas fonctionner correctement. Si l'utilisateur souhaite lancer une nouvelle partie, il lui est conseillé de modifier, si il le souhaite, au préalable les options du jeu afin qu'elles soient prises en compte dans la partie. Lorsque l'utilisateur lance la partie, il doit indiquer le nombre de joueurs et la partie peut alors commencer.



```
-bash
Phase: Montée

Cartes du joueur 1: [ 10♦ ]
Cartes du joueur 2: [ 2♦ ]
Cartes du joueur 3: [ R♣ ]
Cartes du joueur 4: [ A♣ ]

Atout pour cette manche: ♥

Joueur 1, combien de plis pensez vous faire? 1
Joueur 2, combien de plis pensez vous faire? 0
Joueur 3, combien de plis pensez vous faire? 0
Joueur 4, combien de plis pensez vous faire? 1
Voulez-vous sauvegarder la partie? non

Joueur 1, quelle carte voulez vous jouer? |
```

Une partie se déroule en deux phases : une phase montante et descendante. Durant les deux phases chaque joueur reçoit un nombre de cartes différent selon la manche, croissant ou décroissant selon la phase du jeu. Les cartes de chaque joueur sont alors affichées à l'écran et chaque joueur est invité à donner son contrat, à savoir le nombre de plis qu'il pense gagner au cours de la manche. Afin d'éviter de rendre l'interface du jeu trop complexe, toutes les cartes sont indiquées en même temps pour tous les joueurs. Tous les joueurs indiquent ensuite, un par un, leur contrat sans oublier que la somme des contrats doit obligatoirement être différente du nombre de cartes distribuées à chaque joueur, obligeant le dernier joueur à changer son contrat si celui-ci ne respecte pas cette règle. Une fois les contrats indiqués par tous les joueurs, le premier joueur ou le joueur ayant gagné le pli précédent, indique la carte qu'il souhaite jouer en premier. Pour cela, il indique le numéro de positionnement de sa carte dans son jeu : c'est à dire en écrivant "1" puis en validant son choix avec la touche "Entrée" si il souhaite jouer sa première carte, "2" sa deuxième carte, etc... Si le joueur ne rentre pas un nombre correct, le programme l'invite à changer son choix en indiquant que la carte demandée n'est pas dans son jeu. Chaque joueur joue ainsi sa carte à tour de rôle puis le gagnant est indiqué par la console et le pli se termine. Le programme comptabilise les points pour chaque joueur et lance le pli suivant, et ainsi de suite jusqu'à ce que la partie s'achève. Une fois terminée, les scores sont affichés et l'utilisateur est redirigé vers le menu de démarrage après avoir validé en appuyant sur la touche "Entrée".

3 Choix durant l'implémentation

Pour ce projet, nous nous sommes donné comme objectif de pouvoir jouer en priorité, c'est à dire sans prendre en compte une interface graphique travaillée, comme nous avait indiqué notre professeur. Une fois le jeu rendu fonctionnel, nous avons embelli l'aspect visuel pour rendre le jeu plus agréable aux yeux de l'utilisateur. Durant la programmation du jeu, nous nous sommes orientés vers différents choix :

- Pour tirer une carte, nous avons décidé de tirer aléatoirement sur les 52 cartes puis ensuite de vérifier si cette carte n'a pas déjà été tirée. Si cette carte est déjà dans le jeu, on retire une nouvelle carte jusqu'à ce que l'on tombe sur une nouvelle carte
- Pour comparer deux cartes afin d'en déterminer le gagnant, nous avons utilisé la comparaison directe d'OCaml en renvoyant `true` si la première carte l'emporte sur l'autre et `false` dans le cas contraire.
- Pour lire le fichier, nous avons procédé ligne par ligne en extrayant chaque élément important dans des types grâce à une fonction de séparation qui permet de ne transformer une chaîne en une liste contenant à chaque élément une partie de la chaîne, entre deux caractères précisés en entrée
- Nous n'avons malheureusement pas pu implémenter une IA dans le code car nous n'avons pas réussi à la faire fonctionner sûrement dû à notre manque d'expérience dans le domaine et nous avons préféré laisser un jeu fonctionnant sans problème uniquement avec des humains qu'un jeu ne fonctionnant pas correctement

4.2 Les limites

La fonction pour tirer les cartes n'est pas forcément la meilleure solution car elle implique dans le cas où la quasi totalité des cartes sont distribués de tomber aléatoirement sur une carte qui n'a pas déjà été tirée. Nous nous sommes aussi rendu compte que la fonction pour séparer une chaîne de caractères n'est peut être pas autant optimisée que les fonctions implémentées directement dans les modules d'OCaml.

5 Contribution de chaque élève au projet

Nous avons décidé de répartir le projet selon les intérêts de chacun à savoir Nicolas se concentrant sur l'évolution de l'utilisateur dans les menus et au cours de la partie, Gauthier sur toute la partie technique du jeu, à savoir distribuer les cartes à chaque joueur, définir le gagnant parmi les joueurs, ajouter les points selon les options définies auparavant ... et enfin Axel sur la gestion de l'écriture et de la lecture du fichier de sauvegarde.

6 Bilan de chaque élève à propos du projet

6.1 Nicolas

Pour ce projet j'ai essayé de m'y mettre le plus tôt possible afin d'exploiter au maximum le temps donné mais j'ai eu du mal à bien utiliser ce temps. J'ai trouvé ce projet très intéressant car il propose de coder un jeu complet qui est peu connu sur internet ce qui augmente la difficulté car nous n'avons pas d'exemple pour comprendre le fonctionnement du jeu. Cependant, je pense rendre un programme respectant bien le cahier des charges et je suis très fier que nous y soyons parvenu.

6.2 Axel

Ce projet m'a paru pas mal compliqué dans un premier temps, car pour moi c'était complètement nouveau, nous venons d'apprendre à utiliser OCaml et nous devons réaliser un jeu de cartes entier avec une réflexion tout par tour, des stratégies, un choix dans la structure visuelle et technique du programme. Je me suis senti un peu désemparé face à la complexité du travail à fournir mais grâce à l'aide de mes camarades j'ai pu apprendre à raisonner méthodiquement et à élaborer des fonctions concrètes permettant au programme de proposer un panel d'options et de dialogues avec l'utilisateur. Ainsi j'ai pu participer pleinement à l'élaboration d'un programme fonctionnel plutôt complexe et je m'en réjouis.

6.3 Gauthier

Pour ce premier projet de l'année de CPI2 nous avons à réaliser un jeu de cartes s'appelant le Jeu de l'Ascenseur. Premièrement je ne savais pas ce qu'était ce jeu et comment on y jouait, j'ai donc cherché sur internet les règles afin de mieux comprendre le principe de celui-ci. J'ai trouvé ce projet très intéressant car nous devons chercher toutes les possibilités de jeu et voir lorsque cela va aller bloquer, nous devons même parfois jouer nous-même pour nous rendre compte et savoir si nous avons la bonne solution. J'ai parfois eu certaines difficultés pour réaliser certaines fonctions mais comme nous avons un bon esprit de groupe, les deux autres personnes étaient là pour m'aider et tout rentrer dans l'ordre. Dans l'ensemble j'ai donc trouvé ce projet intéressant dans l'ensemble, il m'a permis de mieux m'habituer au langage OCaml.