Algorithme et programmation procedurale, Semestre 2. Compte rendu du projet ascenseur

Chléo CHEVRIER
Fathima MOHAMED-FARHAN

21 juin 2019

Table des matières

I Introduction

Le projet consiste à programmer un jeu, l'ascenseur. Dans un premier temps nous expliquerons les règles du jeu. Puis nous dirons comment nous avons l'intention de procéder pour coder. Puis nous analyserons au fur et à mesure. Puis nous finirons par une conclusion. L'objectif est d'implémenter le jeu en pascal pour 2 à 5 joueurs en deux versions : une première qui est 'humaine' et une seconde qui est constituée de joueurs qui sont des ordinateurs.

II Règles du jeu (prises dans l'énoncé du projet)

L'Ascenseur est un jeu de cartes qui nécessite 2 à 5 joueurs, ainsi qu'un paquet de 52 cartes « normales » sans joker (pique, cœur, carreau, trèfle, et pour chaque couleur, dans l'ordre croissant de valeur : numéros 2 à 10, puis valet, dame, roi et as). A chaque tour, le premier joueur (ou celui qui a gagné au tour précédent) expose une carte de son jeu, puis les autres joueurs doivent à leur tour exposer une carte de la même couleur, l'un après l'autre (ceux qui n'ont pas une carte de cette couleur peuvent en exposer une autre de leur jeu, n'importe laquelle). Celui qui a joué la plus forte carte remporte le pli, ou la « levée », et joue le premier au tour suivant. Au bout d'un certain nombre de tours, les joueurs n'ont plus de cartes à exposer, et la manche se termine. La particularité de l'Ascenseur est que, contrairement à un « jeu de levées » classique, celui-ci fonctionne par contrat. En effet, au début de chaque manche, les joueurs doivent tous parier, en annonçant le nombre de plis qu'ils comptent remporter. S'ils n'ont pas le nombre exact à la fin de la manche, ils perdent des points pour chaque pli manquant, ou au contraire pour chaque pli en trop s'ils en ont remporté plus que prévu. Ceux qui ont « rempli leur contrat » gagnent des points pour chaque pli remporté. Il est tout à fait possible d'annoncer qu'on n'en fera aucun. Notez toutefois que le nombre total de paris doit obligatoirement être égal au nombre de cartes donné à chaque joueur, plus un. Et comme celui qui a distribué les cartes est le dernier à parier, il est fréquent qu'il soit obligé de parier bien plus que ce qu'il a prévu. Une partie complète se déroule en deux phases. Dans la phase ascendante, les joueurs reçoivent chacun une seule carte à la première manche, 2 cartes à la deuxième, 3 à la troisième... et ainsi de suite jusqu'au nombre

maximal de cartes distribuables à chaque joueur (10 cartes pour 5 joueurs, 13 pour 4 joueurs...). Dans la phase descendante, les joueurs reçoivent chacun de nombre maximal à la première manche, puis une carte de moins à la suivante, et ainsi de suite jusqu'à l'ultime manche, qui se joue avec 1 seule carte chacun (d'où le nom de « jeu de l'Ascenseur »). Le décompte des points se fait à la fin de la partie, le gagnant étant celui qui possède le score total le plus élevé (il est possible de faire également un décompte en milieu de partie). Une autre particularité de l'Ascenseur est que c'est un jeu avec atout. En effet, à chaque manche, quand les cartes ont été distribuées à tous les joueurs, la première carte du tas non distribué doit être retournée, et sa couleur (pique, cœur, carreau ou trèfle) sera la « couleur de l'atout » (pour cette manche uniquement). Ce qui veut dire que pendant un tour, si un joueur n'a pas la couleur demandée, il peut « couper » avec une carte de la couleur de l'atout (notez qu'il n'y est pas obligé : il peut jouer une carte d'une autre couleur, qui sera alors sans valeur). Celui qui remporte le pli est donc, soit celui qui a la carte la plus forte de la couleur demandée (si personne n'a coupé), soit celui qui a coupé avec la carte la plus forte de la couleur de l'atout. Dans certaines variantes du jeu de l'ascenseur, on n'utilise l'atout qu'en phase descendante. Dans d'autres variantes, on l'utilise lors des deux phases. En ce qui concerne le décompte des points, il existe plusieurs variantes. Dans l'une d'entre elles, à chaque manche, les joueurs qui remplissent leur contrat (en empochant le nombre exact de plis qu'ils avaient prévu initialement) gagnent 5 points par pli gagné. Ceux qui ont donc parié qu'ils ne feraient aucun pli ne gagnent rien du tout. Quant à ceux qui n'ont pas rempli leur contrat, ils perdent 5 points par pli en moins ou en trop. Dans d'autres variantes, le nombre de points perdus à chaque manche est fixe, peu importe le nombre de plis qu'on a en moins ou en trop. Dans d'autres variantes encore, chaque joueur remplissant son contrat remporte 10 points par défaut + 2 points par pli remporté.

III Attribution des points

Pour les modalités d'attribution du nombre de point nous décidons d'attribuer au gagnant du tour 10 points, plus 2 points par plis remportés. Et nous décidons d'attribuer -2 points par écart entre le nombre de plis parié et le nombre de plis faits.

C'est la procedure *Calcule-Point* qui calculeras les points. Elle prend en

entre le tableau des contrats (au est enregidtrée les contrat de chaqu'un) et un autre tableau ou est enregistrée les plis utilisés il compare les 2 tableaux et complète un troisième tableau qui enregistre les points.

IV Nombre de phase

Nous avons calculé le nombre de cartes qu'il fallait distribuer au joueur en fonction du nombre de joueur au total en n'utilisant pas d'atout (ce qui correspond a la phase ascendante) :

Nombre de joueur	Nombre de phase et nombre de carte dans la dernière étape de la phase ascendante et minimale de la phase descendante.	Nombre de carte restante après avoir distribué les cartes à la dernière étape de la phase ascendante ou à la première étape de la phase descendante.
2	26	0
3	17	1
4	13	0
5	10	2

TABLE 1 – tableau du nombre de carte distribuer et restante en fonction du nombre de joueur

Nous avons, de même, calculé le nombre de cartes qu'il fallait distribuer au joueur en fonction du nombre de joueur au total tout en utilisant un atout (ce qui correspond la phase descendante) :

Nombre de joueur	Nombre de phase et nombre de carte dans la dernière étape de la phase ascendante et minimale de la phase descendante.	Nombre de carte restante après avoir distribué les cartes à la dernière étape de la phase ascendante ou à la première étape de la phase descendante. et après avoir mis l'atout.
2	25	1
3	17	0
4	12	3
5	10	1

TABLE 2 – tableau du nombre de carte distribuer et restante en fonction du nombre de joueur avec atout

Dans le programme principal, une fonction va calculer à chaque manche le nombre de carte à distribuer et fair ces calcul.

Dans notre programme le nombre de manche seras calculer par la fonction *nb-de-joueurs-calc-Manches* qui le nombre total de manche.

V Les Types et variables

1 Les types

On a crée plusieur type tout d'abord un type pour les cartes qui contient sont numéraux de (2 à 14), sa couleur, un boolean (initialiser a TRUE et utiliser pour la distribution, l'interface graphique, et lors du tour de table. On as rajouter un second string qui associeras à la carte son image. Puis un type pour les Atout qui contient la couleur de l'atout et un boolean pour savoir si il est activer ou pas. On a aussi crée un noeud et un pointeur qui point vers des noeuds. La valeur du pointeur seras des tableau de carte.

On a aussi crée plusieur Tableau statique au dynamique :

— tableau2d : un tableau statique de carte en 2D serviras, pour le melange
— tableau : un tableau dynamique des cartes en 1D serviras, pour la distribution et pour le tour de table ;
— tabplateau tableau de nombre dynamique, serviras pour les paris et le calcule des plis gagner et le clcule de point.

2 Les variables principales

Lors de notre projet nous avons initialiser de nombreuse variables. Nous citerons, ici, les variables les plus importante :

- numdeManche ; le numéro de la manche (a ne pas confondre avec M qi est le nombre de carte dans la main des joueurs)
- nbdeJoueur ; le nombre de joueur qui seras initialisée au début
- J1,J2,J3,J4,J5 ; Les paquets des joueurs
- tabC ; les cartes posés sur la table
- je ; le numéro de chaque joueur (variables très importante car elle seras initialiser au début des paris. Elle nous permetras de savoir qui commence à poser sa carte. Puis elle changeras a chaque tour en fonction de qui gagne le tour se qui nous permetras de savoir qui commence au prochain tour. De plus dans les tableaux TabC / P / Preal / TabPoint il seront remplie en fonction de cette variable. En effet onreserve la première case de ces tableau au joueur 1, la deuxième au joueur 2 ... Il est dont important de savoir qui a ganer, perdue, quelle est sa carte, son score, ses paris ...
- P, Preal, TabPoint ; qui sont des tableau d'entier. dns le primier on enregistre les contrats de chaque joueur, dans le second on enregistre les plis réalisés, et dans le troisième on calcule les point des joueurs. Les deux premiers tableaux seront réinisialiser a la fin de la manche.
- at ; qui correspond a l'atout. Il seras activé uniquement à la second phase
- Ptr-add ; Notre liste chainée circulaire qui relieras les paquets de tous les joueurs. Il permetras une meilleur fluiditée quand on passe d'un joueur à un autre durant les paris et les tours de table
- tabcarte ; Il corespond au paquet de carte au tous debut on l'initialise avec toutes les cartes et on le mélange.
- tabcarte2 ; le tableau des cartes en 1D, le tableau seras initialisé une fois le tableau 2D melangé. Il servira à la dristribution.

VI Programme principal

Dans notre programme principal on demande a quelle version le joueur veux jouer :
La version humaine ou la version Ordinateur Puis en fonction de se qu'a tapé

l'utilisateur, il active une procedure.

VII La version humaine

Pour la version humaine nous allons créer un programme destiné à une utilisation comprise entre 2 et 5 personnes. Le jeu est constitué de deux phases. La première est la phase montante et la deuxième est la phase descendante. Lors de la phase montante, on n'utilisera pas l'atout. Par contre, nous l'utiliserons dans la phase descendante.

On commence par demander le nombre de joueur n. Puis on initialise quelques variables. On créé une boucle repeat qui correspond aux manches. On la fera répéter m fois ce qui correspond au nombre de manche au total (phase ascendante + descendante). Puis on mélange les cartes, on les distribues et on forme un paquet par joueur. On utilise la méthode expliquée plus bas. Les paquets de joueurs sont des tableau relier par une liste chainer.

Pour savoir qui commence les paris, la Procedure appelle une autre Procedure qui active l'interface graphique. On ouvre la table des paris (appeler P) puis on retourne les paris dans un tableau qui seras garder en mémoire jusqu'à la fin de la manche.

Puis dans la structure principale, on initialise le nombre de carte dans la main du joueur (au debut il est egale au numéro de la manche). On créé une boucle repeat pour chaque tour. Le procedure *association* active les passages.Une fois le tour de table finie le programme active. La procédure *PremiereCarte* qui retourne qui remporte le plie. Puis le programme remplie le tableau compte les plis de joueur (appeler Preal) .

On relance la procédure tour jusqu'au tour m-1 avec m qui correspond au nombre de cartes.

A la fin de la première boucle on Calcule les points et le nombre de manche prend +1. Sauf si le numero de la manche depasse 52/nbdeJoueur. Dans se cas on active les atout et le numéraux de la manche prend -1.

1 La version ordinateur

La version ordinateur se comporte des même phase que la version humaine.

On commence par demander le nombre de joueur humain et le nombre de joueur fictif. On fait la somme des deux pour avoir le nombre de joueur au totale.

Puis on enchaine avec le mélange des cartes et la distribution. Chaque joueur vas ensuite donner ses contrats et la procédure *PARI-IA* vas ensuite retourner le tableau des contrats (comme durant la version humaine).

VIII Création du paquet de carte

Le paquet de carte prend la forme d'un tableau en deux dimension de 4x13 appeler (tabcarte). Il est construit à partir de deux tableau d'une dimension. Le premier (tabcouleur) est un tableau de chaine de caratère, et le deuxième (tabcouleur) est un tableau d'entier qui contiendras tous les entier de 2 a 14. Les nombre correspondant chaqu'un a une carte (2 =2 , 11 = valet , 12 = dame , 13 = roi , 14 = as)et la chaîne de caractère sera la couleur de la carte (t r è f l e , c a r r e a u , c o e u r , p i q u e). Grace à la fonction *Remplissage-du-tabl-ranger* , on obtient un tableau de deux dimension dont les cases contiennent un entier, une chaine de caractere et un booléen. Le booléen nous sera utile pour le mélange des cartes.

IX Mélange du paquet de carte

Pour mélanger le paquet de cartes, on utilise la procédure *Mélange-des-cartes1* et *Mélange-des-cartes1*. Le principe est le suivant : on choisit aléatoirement deux cases qui correspondent à deux cartes du tableau, et on échange leur place. Elle sont ensuite marqué false.La procédure commence par prendre la valeur de la première case et s'arrête à la dernière case de la première ligne. La première procédure prend en compte toute les cartes, celle qui sont marquées en TRUE et en FALSE. La deuxième prend uniquement celle marquer en TRUE (donc celle qui n'ont jamais été melangées). CE qui garantir un meilleur mélange.

X Distribution du paquet (mélangé)

On aura donc un premier tableau de 4 colonnes et de 13 lignes. Pour avoir un paquet mélangé, on va créer un deuxième tableau d'une ligne et de 52 colonnes qui sera rempli aléatoirement à partir du premier. Pour différencier les cartes qui ont déjà été distribuée, on ajoute dans le premier tableau un booléen qui indiquera false si la carte a déjà été distribuée.

XI Contrats

1 Version humaine

On vas calculer l'ordre dans lequelle les paris se font en fonction du nombre de joueur et du numero de la manche avec les modulos. puis on demande à chaque joueur leur contrats tout en affichant leur jeu. Le contrats du dernier seront imposée par la procedure *Pari-dernier*. La procedure vas retourner le tableau des paris et le programme principal dle garderas en memoire.

2 Version ordinateur

Au debut, on regarde si le joueur est humain, si il est humain alors on appelle les même fonction que pour la première version. Si il un ordinateur alors on active la procedure *calcule-pari-IA* et *calcule-pari-IA-ATOUT* qui vas calculer les paris du robot.
il calcule le nombre de carte qu'il a dans la main qui correspont au contition des 'if', en effet en fonction de si ont joue avec un atout ou pas les condition vont changer. puis il regarde si sont premier paris est pasq trop haut en fonction des autre paris dans se que il va reduir sont nombre de plis. Comme avec l'autres versions il retourne le tableau des plis.

les contrats

Pour les contrats nous avons utilisé les procedures : Ordre-Pari qui utilise la fonction affiche jeu qui à la fois affiche le jeux et demande quelle carte ils vont jouer ainsi que le nombre de paris. Le nombre de paris est choisi en fonction du nombre de carte suoérieur à 10. Si l'ordinateur n' est pas le premier et subit le nombre de paris i devra respecter ce contrat en jouant.

remplir le contrat

Pour remplir les contrats on a crée deux procedures, Perdre et Gagner. Nous allons utiliser la fonction gagner jusqu'a ce que l'ordinateur remplisse son contrat pour les paris.

On a initialiser deux variables au début : taille à 0 et nb à 1. Nous avons aussi crée un booléen (Premier) qui sert à savoir si l'ordinateur est le premier à jouer (Premier = False) ou bien si il n'est pas la premier à jouer(Premier = Vrai) Nous commencaons par quand l'ordinateur n'est pas le premier à jouer. Pour cela, nous nous intéressons au différent cas possible que l'ordinateur puisse rencontrer afin de mieux coder. Le premier cas possible est si parmis les cartes jouées,la carte gagnante est de la couleur de l'atout. Dans ce cas la si il ne joue pas une carte de la couleur de l'atout supérieur à la carte gagnante il faur qu'ilpose une carte qui lui est inférieur (respectivement supérieur pour gagner) dans la coulaur ou bien n'importe qu'elle carte d'une autre couleur (dans le meilleurs des cas les cartes les plus grandes, respectivement l'inverse our gagner). Ensuite il y a le cas où la carte gagante n'est pas de la couleur de l'atout. Il vas aloors mettre dans un tableau qu'il crée toute les carte de la meme couleur et inférieur (respectivement supérieur) et grace qu tri à bulle il va classer par ordre croissant les cartes et choisir la carte maximum (respectivement maximum).

XII Les tours de table

Les tours de tables est faites grace àdes listes chainées afin de savoir qui commence à chaque manche. à chaque tour. Elle marche bienpour la version humaine mais malheurement nom pour la version avec les ordinateurs.

1 Version humaine

l'interface vas s'afficher puis chaque joueur entre dans la console le numero de la carte qu'il veut jouer

2 Version ordinateur

La version ordinateur malheureusement ne marche plus dès que l'ordinater est le premier à jouer dans une manche et aussi quand on change de manche.

XIII Le calcul des points

Pour le Calcul des points nous n'avons pas utilisé les régles variantes. On a utilisée celle expliquer dans le sujet.Pour cela on a utiliser les fonction : *premiereCarte* et *premiereCarteAtout* pour savoir qui a remporter le plis et *Calcul-Points* qui calcul les points et se charge de remplir le tableau des points.

XIV Interface Graphique

Pour l'interface graphique, on utilise le dossier disponible sur Arel qui nous permet de définir les formes et les couleurs utiles à l'interface graphique.

XV Les améliorations à apporter

Les améliorations à aportées : - améliorer l'Interface graphique - Nommer les ordinateurs qui jouent.

XVI Bilan Personnelle

Ce projet nous aura permis de mieux manipuler en profondeurs les Types, les tableaux et surtout les pointeures et les listes chainés.On avait pas fait beaucoup d'application dessus. Ce projet nous aussi permis de métriser l'intérface graphique juste à l'aide d'un diapo dessus , sans cours... Cela nous confirme que l'on est pas obliger d'attendre d'avoir un cour pour apprendre une notion en Informatique.

XVII Conclusion

Pour conclure nous avons réussi à programmer la majorité du jeu et malheuresement même après avoir détecter la zone où il y a le problème nous ne parvenons pas à trouver l'erreur malheureusement.