

Compte Rendu : Informatique Projet n°1

Anabelle, Pierre-Louis



Abstract

Voici le compte rendu du projet d'informatique n°1. Vous trouverez dans ce document le cahier des charges, une description du projet, les difficultés rencontrées et les limitations de notre script actuel. Ce projet consiste en la programmation d'un jeu de cartes, "Le Trio", pouvant accueillir entre 3 et 6 joueurs. Ce jeu met en avant une logique de stratégie et de mémoire, où chaque joueur doit tenter de révéler les trois occurrences d'un même numéro pour marquer un point. Le projet comprend plusieurs versions améliorées du jeu, allant d'une version basique à une version avec une intelligence artificielle adaptative, offrant des défis de stratégie plus intéressants. L'objectif principal est de garantir une expérience utilisateur simple et efficace tout en maintenant un code clair et optimisé.

Table of Contents

Cahier des Charges	1
Description du Projet	1
Versions	1
Version A+B	1
Version C	1
Version D	2
Accès au Code	2
Difficultés et Solutions Trouvées	2
Limites et Perspectives	3
Annexe	4

Cahier des Charges

Le projet est de coder et programmer un jeu de cartes, plus précisément le jeu du Trio.

Le Trio est un jeu de cartes pouvant se jouer de 3 à 6 joueurs. Il est composé de cartes numérotées de 1 à 12, avec 3 occurrences de chaque numéro, soit un total de 36 cartes. Au début de la partie, les cartes sont distribuées face cachée entre les joueurs, et certaines cartes restent totalement inconnues et sont placées au centre.

L'objectif de chaque joueur est de révéler, à son tour, les 3 occurrences d'un même numéro. Pour ce faire, lorsqu'il est de son tour, un joueur peut choisir entre : révéler la carte minimum d'un joueur (y compris lui-même), révéler la carte maximum d'un joueur (y compris lui-même) ou révéler une des cartes situées au centre. Il répète cette opération deux fois si les deux premières cartes révélées sont différentes, ou trois fois si elles sont identiques. Si les trois cartes révélées comportent le même numéro, elles sont retirées de la partie et le joueur gagne un point. C'est ensuite au tour du joueur suivant.

La partie se termine lorsqu'un joueur atteint trois points ou lorsqu'un joueur révèle le trio de 7.

L'objectif du projet est de programmer le jeu de cartes du Trio de sorte à ce qu'il soit le plus ergonomique possible pour l'utilisateur, tout en optimisant le code et en le rendant compréhensible pour autrui.

Description du Projet

Versions

Version A+B

Dans la version A+B, nous avons programmé une version du jeu du Trio qui prend en compte le nombre de joueurs choisi entre 3 et 6. Dans cette version, l'utilisateur doit jouer tous les joueurs à tour de rôle. Attention, cette version est intolérante aux fautes et aux variations de syntaxe.

Version C

Dans la version C, le code est conçu pour 3 à 6 joueurs. L'utilisateur joue le joueur 1 tandis que l'ordinateur s'occupe de jouer pour tous les autres joueurs. Les actions des joueurs gérés par l'ordinateur sont complètement aléatoires. Dans cette version, nous avons prévu une gestion complète des erreurs,

permettant à l'utilisateur de continuer la partie même en cas d'erreur de saisie, et de reprendre son action.

Version D

Dans la version D, on retrouve les mêmes fonctionnalités que dans les versions précédentes. De plus, l'ordinateur réalise des choix avec un minimum d'intelligence lorsqu'il joue à la place d'un joueur. Lorsqu'un joueur géré par l'ordinateur doit jouer, ses actions dépendent d'une base de connaissances contenant les cartes du joueur ainsi que les cartes déjà révélées durant la partie. Si cette base de données contient trois cartes avec le même numéro, elles deviennent la cible du joueur et sont révélées en priorité. Si ce n'est pas le cas, la cible du joueur est soit sa carte minimum soit sa carte maximum. En fonction de la première carte retournée, la cible est modifiée. Enfin, les actions du joueur dépendent de la valeur de sa cible : si la cible est inférieure ou égale à 6, le joueur demande un minimum ; si cette valeur est supérieure à 6, le joueur demande un maximum.

Accès au Code

Pour accéder au code, deux méthodes s'offrent à vous :

- **Replit** : La méthode la plus simple est d'aller sur [Replit](#) via le lien d'invitation suivant. Un compte vous sera demandé, que vous pourrez créer rapidement et gratuitement. Il vous suffira ensuite de cliquer sur "run" (ou ctrl+enter) et d'entrer le numéro associé à la version que vous souhaitez exécuter (le numéro associé (1, 2 ou 3) et non la lettre correspondant à la version).
- **GitHub** : La deuxième façon est de suivre les instructions présentes dans le README du répertoire GitHub suivant pour installer les dépendances nécessaires et lancer le programme : [Projet-Info sur GitHub](#).

Difficultés et Solutions Trouvées

L'une des principales difficultés rencontrées concernait la gestion des cartes au centre ainsi que la stratégie. Avec un jeu de cartes physique, les "cartes au centre" sont placées face cachée et leur position reste constante tout au long de la partie, permettant aux joueurs de les retourner progressivement et de mémoriser leur valeur ainsi que leur emplacement. Cependant, en Python, les cartes ne sont pas concrètes, ce qui a introduit des complications dans la représentation et la gestion de la mémoire des positions.

Pour pallier cela, nous avons initialement pensé à placer les cartes du centre dans une liste, en utilisant l'indice de chaque élément pour représenter sa position. Toutefois, ce choix a engendré un problème supplémentaire : lorsque l'une des cartes du centre est retirée suite à un trio trouvé, les indices de la liste sont réévalués, modifiant ainsi la position perçue par les joueurs. Afin de conserver la mémoire des cartes et leur emplacement, nous avons décidé d'utiliser un dictionnaire dans lequel chaque carte du centre est associée à une clé unique. Cette clé joue le rôle de position fixe tout au long de la partie, permettant ainsi aux joueurs de retenir les cartes sans que leur emplacement ne change.

Une autre difficulté rencontrée concernait l'intégration d'une intelligence artificielle rudimentaire pour les joueurs contrôlés par ordinateur. Bien que nous ayons implémenté une mémoire globale qui permet à chaque joueur de suivre les cartes connues, son impact reste faible sur la stratégie de jeu. Les joueurs contrôlés par ordinateur se concentrent principalement sur les cartes qu'ils possèdent eux-mêmes, sans analyser de manière approfondie les informations connues par les autres joueurs. Ainsi, malgré l'accumulation des cartes connues au fil de la partie, leur stratégie n'évolue pas vraiment.

Limites et Perspectives

Malgré les améliorations apportées à l'intelligence artificielle des joueurs contrôlés par ordinateur, certaines limites persistent, laissant place à de futures améliorations.

L'une des limitations principales est le côté très basique de la stratégie actuelle des joueurs gérés par l'ordinateur. En effet, ces joueurs choisissent leurs cartes cibles en se basant surtout sur leurs propres cartes, accordant peu d'attention à la mémoire globale. Pour rendre cette intelligence artificielle plus complète et plus proche du comportement humain, il serait pertinent de développer un système d'analyse plus complexe qui prend en compte de manière optimale l'ensemble des informations disponibles, permettant ainsi des décisions stratégiques plus cohérentes et adaptatives.

Une autre limite concerne la nature uniforme de la mémoire des joueurs contrôlés par ordinateur. Contrairement aux humains qui, d'un joueur à l'autre, peuvent retenir différentes cartes et en oublier d'autres, les joueurs virtuels de notre script disposent d'une mémoire identique. Cela réduit la diversité des stratégies et les rend prévisibles. Une piste d'amélioration consisterait à introduire une forme d'évolution ou de différenciation dans la mémoire des joueurs, leur permettant de simuler la variabilité et la créativité humaines. En permettant à chaque joueur contrôlé par ordinateur de développer une stratégie individuelle, nous pourrions rendre le jeu plus complexe et intéressant.

Ces améliorations potentielles pourraient non seulement rendre le jeu plus dynamique et imprévisible, mais également améliorer l'expérience des joueurs humains, qui se retrouveraient face à des adversaires plus évolués et variés.

Annexe

Voici quelques captures d'écran illustrant les situations clés de notre script.

```
C'est au joueur 1 de jouer.  
Vos cartes sont : [3, 4, 7, 9, 12]  
  
Saisissez "c" pour révéler une carte au centre, "max" pour révéler la carte  
la plus grande d'un joueur, "min" pour révéler la carte la plus petite d'un  
joueur : min  
Saisissez le numéro du joueur dont vous voulez révéler la carte (vous pouvez  
choisir votre propre numéro) : 4  
La carte révélée du joueur 4 est : 1  
Saisissez "c" pour révéler une carte au centre, "max" pour révéler la carte  
la plus grande d'un joueur, "min" pour révéler la carte la plus petite d'un  
joueur : min  
Saisissez le numéro du joueur dont vous voulez révéler la carte (vous pouvez  
choisir votre propre numéro) : 4  
La carte révélée du joueur 4 est : 1  
Saisissez "c" pour révéler une carte au centre, "max" pour révéler la carte  
la plus grande d'un joueur, "min" pour révéler la carte la plus petite d'un  
joueur : min  
Saisissez le numéro du joueur dont vous voulez révéler la carte (vous pouvez  
choisir votre propre numéro) : 5  
La carte révélée du joueur 5 est : 1  
Trio gagnant ! Un point pour le joueur 1  
Tour terminé, c'est au joueur suivant.
```

Figure 1: Le joueur humain gagne un trio

```
Trio gagnant ! Un point pour le joueur 1  
Tour terminé, c'est au joueur suivant.  
  
Le joueur 1 est le gagnant !
```

Figure 2: Un joueur gagne la partie

```
C'est au joueur 5 de jouer.  
Ce joueur est g  r   par l'ordinateur.  
Cartes de l'ordinateur : [1, 6, 6, 7, 12]  
  
Le joueur 5 cherche    compl  ter le trio de 12  
Le joueur 5 choisit : max  
Le joueur 5 choisit de r  v  ler la carte maximale du joueur 5  
La carte r  v  l  e du joueur 5 est : 12  
Le joueur 5 choisit : max  
Le joueur 5 choisit de r  v  ler la carte maximale du joueur 2  
La carte r  v  l  e du joueur 2 est : 12  
Le joueur 5 choisit : max  
Le joueur 5 choisit de r  v  ler la carte maximale du joueur 1  
La carte r  v  l  e du joueur 1 est : 12  
Trio gagnant ! Un point pour le joueur 5  
Tour termin  , c'est au joueur suivant.
```

Figure 3: Un joueur g  r   par l'ordinateur gagne un trio

```
C'est au joueur 1 de jouer.  
Le joueur 1 n'a plus de cartes.
```

Figure 4: Un joueur n'a plus de carte

```
Cartes disponibles au centre : [0, 1, 2, 3, 4, 5]  
Scores actuels : [0, 0, 0, 0, 1, 0]
```

Figure 5: Affichage du score