

Optimisation de stratégies pour une variante du tarot mexicain

Introduction

- Principe du jeu:

- Cartes: atouts du tarot
- But: prendre le nombre de plis préalablement annoncé
- Un écart avec l'annonce se convertit en un point

- But du TIPE:

- modélisation informatique de parties
- élaborer une stratégie optimisée

Première étape

- stratégie de base en ce qui concerne les annonces et le jeu
- Simuler des parties de tarot africain entre 4 ordinateurs
- Créer un rendu graphique des résultats

Première étape

Fonctions utilisées:

- **Annonce(main)**

Fonction qui renvoie l'annonce d'un joueur pour une main donnée.

- **gagnantdupli(cartesjoueescetour,gagnantdupliprecedent)**

Fonction qui renvoie le numéro du joueur qui remporte le pli.

- **maxdespluspetits(main,cartesjoueescetour)**

Fonction qui renvoie la carte la plus basse de la main, étant plus haute que toutes celles déjà jouées par les autres joueurs. Si le joueur n'a pas de telle carte, la fonction renvoie la carte la plus basse de de la main.

Première étape

- **mindesplusgrands(main,cartesjoueescetour)**

Fonction qui renvoie la carte la plus haute de la main, étant plus basse que toutes celles déjà jouées par les autres joueurs. Si le joueur n'a pas de telle carte, la fonction renvoie la carte la plus haute de de la main.

- **choixcarte(main,cartesjoueescetour,n)**

Fonction qui renvoie la carte qui doit être posée par un joueur dans une situation donnée.

Première étape

```
def annonce(main):  
    k=0  
    e=0  
    for i in main:  
        if (22>i and i>17):  
            k+=1  
        if i==22:  
            e+=1  
        if(e==1 and k<=2):  
            k+=1  
    return(k)
```

**On ajuste l'algorithme
d'annonce en simulant un
grand nombre de parties**

Limite placée en 14 : moyenne de 2.54

Limite placée en 15 : moyenne de 2.00

Limite placée en 16 : moyenne de 1.68

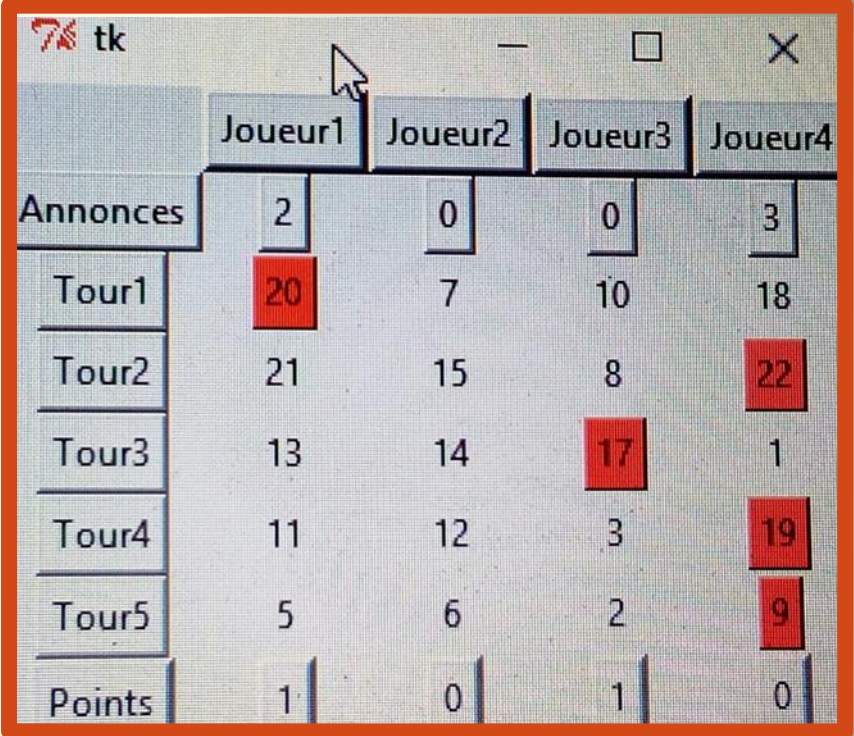
Limite placée en 17 : moyenne de 1.66

Limite placée en 18 : moyenne de 1.95

Première étape

Ici, simulation d'une partie:

- **Joueur1** annonce 2 plis et n'en prend qu'un: il gagne un point
- **Joueur2** annonce 0 et fait 0: il ne gagne pas de point
- **Joueur3** annonce 0 et prend un pli: il gagne un point
- **Joueur4** annonce 3 et prend 3 plis: il ne gagne pas de point



	Joueur1	Joueur2	Joueur3	Joueur4
Annonces	2	0	0	3
Tour1	20	7	10	18
Tour2	21	15	8	22
Tour3	13	14	17	1
Tour4	11	12	3	19
Tour5	5	6	2	9
Points	1	0	1	0

Deuxième étape

- Stratégie avancée en ce qui concerne les annonces et le jeu
- Simuler des parties entre les ordinateurs
- Observer les résultats et ajuster l'algorithme

Deuxième étape

Fonctions utilisées:

- Les fonctions utilisées en première étape
- **plusgrand(carte,cartesjouees,main)**

Fonction qui renvoie toutes les cartes supérieures à une carte encore en jeu

- **maxinf(main)**

Fonction qui renvoie la carte la plus haute en dessous de 18 dans la main du joueur.

Deuxième étape

- **Choixcarteplus(annonces,main,plis,cartesjouees,g)**

Fonction qui renvoie la carte qui doit être posée par le joueur dans une situation donnée.

Améliorations:

- Séparation +18/-18
- Prend en compte les annonces
- Prend en compte la position dans le round

Conclusion

Stratégie optimisée

- Vision globale du jeu
- Anticipation
- Optimisé uniquement contre une stratégie précise

Simulation informatique

- Parties entières
- Représentation graphique sous forme de tableau