

# MORPION

Le Morpion est la version généralisée du Tic-tac-toe, dans le sens où le Tic-tac-toe est un Morpion de dimension  $n \times n$  avec  $n = 3$ , où  $n$  est un entier naturel. C'est un jeu célèbre pour sa simplicité et son accessibilité à tous mais néanmoins, il est possible d'en faire un jeu imbattable. Etant jouable à deux, l'option de jouer soit avec un autre utilisateur soit avec l'ordinateur est viable. Ce programme s'intéresse à l'implémentation d'une intelligence artificielle dans le jeu et n'offre ainsi comme possibilité que la deuxième option. On a fait usage de l'algorithme MiniMax pour programmer l'intelligence artificielle.

## But du jeu

Le but du Morpion est pour un joueur d'aligner, en rajoutant dans le plateau de jeu de dimension  $n \times n$ ,  $n = 5$  de ses pions successivement dans le cas général ( $n \geq 5$ ) et aligner  $n$  elle même pions successifs pour  $n$  étant 3 ou 4. Il n'importe pas si l'alignement est fait verticalement, horizontalement ou obliquement. Dans notre cas,  $n = 3$ .

## Règles du jeu

### 1. Placement des pions:

Le choix du premier à remplir une case est laissé à l'utilisateur. Après que la première case aie été remplie les joueurs s'alterneront par tour à remplir les cases pour essayer de gagner, ayant la possibilité de ne remplir qu'une seule case par tour, jusqu'à ce que soit l'un d'entre eux remporte la Victoire ou le plateau soit rempli et ainsi le jeu se termine en égalité.

### 2. Aligner les pions:

Pour gagner il suffit d'aligner ces pions successivement sur 3 cases, que l'alignement soit faite horizontalement, verticalement ou obliquement, avant son adversaire

## Développement du jeu

Le programme a été écrit en C++, utilisant la programmation orientée objet. Fait uniquement en Console, elle comporte les options suivantes:

- Classe Menu:

- Fonction initMenu:

*pour initialiser les composantes du menu*

*c'est-à-dire, les attributs de la classe*

- Fonction getChoice:

*pour implementer la fonctionnalité de permettre*

*à l'utilisateur d'interagir avec le menu et faire son choix*

- Fonction manageChoice:

*pour faire en sorte que chaque choix mène*

*correctement à l'étape que le menu indique*

- Fonction digit:

*servant à assurer que les entrées aux claviers se déroule*

*correctement*

- Classe Jeu:

- Fonction setprof et getprof:

*Getter et setter pour l'attribut prof (profondeur)*

- Fonction initiliaser, showBoard et showInstructions:

*Servant à initiliaser et montrer la progression du plateau de jeu*

- Fonction start et play:

*Servant à lancer le jeu et définir son déroulement*

- Fonction rowCrossed, columnCrossed, diagonalCrossed, gameOver et declareWinner:

*Servant à verifier si le jeu est terminé et determiner qui a gagné*

- Fonction bestMove et minimax:

*Servant à faire les calculs pour le fonctionnement de l'IA*

## Librairies utilisé

- `#include <iostream>`:  
Bibliothèque standard gérant les Input/Output
- `#include <vector>`:  
Afin de créer un tableau dynamique servant à contenir les options du menu principal
- `#include <limits>`:  
Afin de faire usage de `std::numeric_limits` en tant qu'argument de `cin.ignore()` avec `std::streamsize`, pour pouvoir extraire sans limite les caractères contenue dans le buffer de `cin` jusqu'à ce que le délimiteur spécifié soit trouvé.

## Conclusion

L'implémentation d'une intelligence artificielle dans le jeu, via l'algorithme MiniMax fu la partie la plus dure, couplé ensuite par l'ajout du menu pour la difficulté. Etant intéressé et fasciné par les jeux , l'intelligence artificielle et la programmation, ce projet m'a permis de m'épanouir sur ces sujets et m'a appris mes limites. J'envisage de rajouter une interface graphique au jeu en utilisant SDL et puis QT ainsi que d'optimiser l'algorithme de l'intelligence artificielle de MiniMax à l'Alpha-Beta. Une fonctionnalité permettant de choisir la dimension du morpion et une autre ajoutant une option d'annuler ces actions et revenir d'un pas en arrière ont été des aspects prévue mais non atteinte.

## Références

- <https://www.learncpp.com/cpp-tutorial/stdcin-and-handling-invalid-input>
- <https://fearyourself.developpez.com/tutoriel/sdl/morpion/part6/>