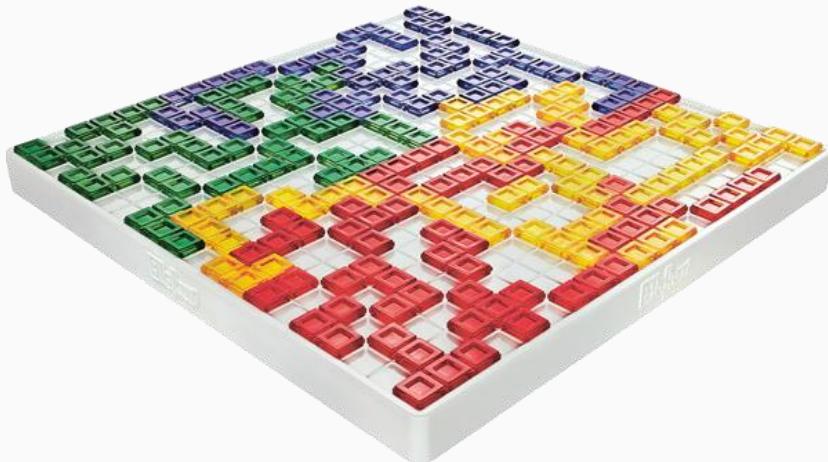


Projet Blokus



↳ [Lien Github](#) ↳

Sommaire

- ◆ [Objectifs atteints](#)
- ◆ [Fonctionnalités réalisées par le soft](#)
- ◆ [Mode d'emploi IHM](#)
- ◆ [Explication de l'interface de placement des pièces en suivant les règles](#)
- ◆ [Répartition des tâches au sein du binôme](#)
- ◆ [Bugs subsistants](#)
- ◆ [Améliorations possibles](#)
- ◆ [Copies écran de réalisation des fonctions essentielles](#)

OBJECTIFS ATTEINTS

	Atteint	Partiellement atteint	Non atteint
Affichage du plateau	☺(° ˘ ° ☺)		
Mode d'emploi de l'IHM	☺(° ˘ ° ☺)		
Choix des pièces (indiquer le nombre...)	☺(° ˘ ° ☺)		
Distinction des pièces pour les 4 joueurs	☺(° ˘ ° ☺)		
Placement des pieces	☺(° ˘ ° ☺)		
Placement des pièces avec rotation	☺(° ˘ ° ☺)		
Placement des pièces avec miroir	☺(° ˘ ° ☺)		
Tour à 2 joueurs	☺(° ˘ ° ☺)		
Compter le nombre de tours	☺(° ˘ ° ☺)		
Score des joueurs	☺(° ˘ ° ☺)		
2 joueurs en reseau			> ^ <
Affichage du plateau pour les 2 joueurs réseau			> ^ <
Interopérabilité win, mac, linux			> ^ <
Tour à 4 joueurs	☺(° ˘ ° ☺)		
4 joueurs en reseau			> ^ <
Affichage du plateau pour les 4 joueurs réseau			> ^ <
Code commenté	☺(° ˘ ° ☺)		

FONCTIONNALITÉS réalisées par le SOFT

Le soft réalise différentes fonctions permettant à notre programme de reproduire au mieux le déroulement du jeu BLOKUS

- ◆ Tout d'abord, notre code initialise plusieurs informations :

Des listes : des joueurs, des pièces, des pièces restantes par joueurs.

Des variables : le nombre de tour, le joueur actuel.

- ◆ Ensuite intervient l'ensemble de nos fonctions :

init_plateau(): permet de créer notre plateau avec les lignes et colonnes numérotées et le plateau au centre (cases vides)

affiche_plateau(): pour afficher le plateau

coin_départ(): propose aux joueurs les quatre coins où placer les pièces, si le coin choisi est disponible, initie le placement de la pièce, sinon redemande le coin

placer_piece_init(): suite de *coin_départ()*, demander la pièce à joueur avec les options possibles et place la pièce si aucun problème, sinon redemande la pièce [explication détaillée dans la fonction *placer_piece()*]

choix_piece(): affiche au joueur actuel l'ensemble des pièces qui lui reste dans la console, et retire de la liste des pièces restantes du joueur la pièce que ce dernier vient de sélectionner pour jouer

affiche_options_piece(): permet d'afficher la pièce choisie sous tous ses angles : les 4 rotations à 90° et les 4 rotations avec le mode miroir, le miroir étant la symétrie de la pièce

placer_piece(): une fois la pièce, avec options, choisie, le programme va essayer de placer la pièce en respectant plusieurs règles : vérifier si la pièce ne déborde pas ou ne se superpose pas, si la pièce est bien dans le coin d'une des pièces du joueur actuel, si la pièce ne touche pas le coté d'une pièce du joueur actuel. Si une des règles n'est pas validée, recommencer le processus de placement avec nouvelle demande

- ◆ Enfin, le cœur du programme :

Le code demande le nombre de joueurs, soit 2, soit 4. À la suite de ce choix, et dans les deux cas, le jeu se lance et commence à demander pour le placement des 4 coins. Après cela commence des tours infinis jusqu'à la fin du jeu, dans notre cas, le jeu se termine quand tous les joueurs ont abandonné : à chaque début de tour est demandé au joueur s'il veut abandonner.

Si tous les joueurs ont abandonné, le programme annonce le nombre de tours, (à raison de 0.25 tour par joueur) et les scores de chacun (-1 point par

carré des pièces restantes), si le joueur n'a plus de pièces, il gagne 15 points en plus. Parmi les points des joueurs, celui qui en a le plus est déclaré vainqueur.

MODE D'EMPLOI IHM

Dans notre version du Blokus, la machine demande dans un premier temps le nombre de joueurs, 2 ou 4.

Ensuite, durant les 4 premiers tours, les joueurs choisissent leur coin de départ.

Après cela, le joueur va devoir choisir une ligne et une colonne pour placer sa pièce, il sélectionne une pièce parmi celles qui lui restent (max 21).

L'IHM lui renvoie toutes les options de cette pièce, soit la rotation et le miroir.

Ensuite, si le positionnement est bon (dans un coin d'une pièce du joueur et pas sur une pièce) la pièce est placée, sinon le joueur doit recommencer le processus de placement et de sélection d'une pièce.

EXPLICATION DE L'INTERFACE DE PLACEMENT DES PIÈCES EN SUIVANT LES RÈGLES

On affiche une grille allant de 0 à 20. Les pièces correspondantes aux joueurs s'affichent avec des lettres de A à D.

Chacun leur tour et à l'aide du plateau, les joueurs choisissent la ligne et la colonne, puis la pièce à placer parmi celles qui restent.

Ensuite avec la console, ils verront les options et pourront choisir la rotation [0-3] et le miroir [oui/non].

Une fois la pièce placée, le programme supprime l'affichage dans la console pour plus de lisibilité, et demande au joueur suivant s'il peut jouer (ou s'il choisit d'abandonner) et recommence l'exécution d'affichage et de placement des pièces.

RÉPARTITION DES TÂCHES AU SEIN DU BINÔME

◆ Commun :

- ◊ Affichage du plateau
- ◊ Rédaction du PDF

◆ Loup :

- ◊ Placement des pièces
- ◊ Options des pièces

◆ Amélie :

- ◊ Affichage du jeu
- ◊ Gestion des pièces
- ◊ Commentaires du code

BUGS SUBSISTANTS

Il existe un problème avec le placement de certaines pièces avec des formes spéciales, il est possible que ces dernières se retrouvent en contact avec le long d'une autre pièce.

AMÉLIORATIONS POSSIBLES

Nous n'avons pas pu travailler sur la partie réseau, car nous ne l'avons pas vu en cours et n'avons pas compris l'exemple de code fourni.

Cependant, nous essayerons de réaliser notre jeu en réseau, car nous aimons le challenge.

COPIES écran de réalisation DES FONCTIONS ESSENTIELLES

Fonction qui affiche le plateau

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20
```

Fonction qui demande d'entre la ligne et la colonne où placer la pièce

```
> Entrer la ligne : 5  
> Entrer la colonne : 5
```

Fonction qui affiche les pièces restantes du joueur

```
Vos pièces restantes sont : -----  
-----  
|< n° 0 >|  
| X |  
|-----|  
|< n° 1 >|  
| X |  
| X |  
| X X |  
| X |  
|-----|  
|< n° 2 >|  
| X |  
| X X |  
| X X |  
| X |  
|-----|  
|< n° 3 >|  
| X |  
| X X |  
| X |  
|-----|  
|< n° 4 >|  
| X |  
| X X |  
| X |  
| X |  
|-----|  
|< n° 5 >|  
| X |  
| X |  
| X X |  
|-----|  
-----  
|< n° 11 >|  
| X |  
| X |  
| X X |  
| X |  
|-----|  
|< n° 12 >|  
| X |  
| X X |  
| X X |  
| X |  
|-----|  
|< n° 13 >|  
| X X |  
| X |  
| X X |  
| X |  
|-----|  
|< n° 14 >|  
| X |  
| X X |  
| X |  
|-----|  
|< n° 15 >|  
| X |  
| X |  
| X X |  
|-----|  
-----  
|< n° 16 >|  
| X |  
| X |  
| X X |  
| X X |  
|-----|  
|< n° 17 >|  
| X X |  
| X X |  
| X |  
|-----|  
|< n° 18 >|  
| X |  
| X X X |  
| X |  
|-----|  
|< n° 19 >|  
| X X X |  
| X |  
|-----|  
|< n° 20 >|  
| X |  
| X X X |  
| X |  
|-----|
```

Fonction qui demande la pièce puis affiche ses options

```
> Choisir la pièce : 19
Options de la pièce :
~ Rotation 0 | Miroir Off:
X
X X X
X

~ Rotation 1 | Miroir Off:
X X
X X
X

~ Rotation 2 | Miroir Off:
X
X X X
X

~ Rotation 3 | Miroir Off:
X
X X
X X

~ Rotation 0 | Miroir On:
X
X X X
X

~ Rotation 1 | Miroir On:
X X
X X
X

~ Rotation 2 | Miroir On:
X
X X X
X

~ Rotation 3 | Miroir On:
X
X X
X X

> Rotation [0-3] : 0
> Miroir [oui/non] : ■
```

Fonction qui place la pièce et l'affiche sur le plateau

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
1 A A
2     A A
3         A
4             A
5                 A A A
6                     A
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Fonction qui demande si le joueur peut encore placer des pièces

```
◆ AU TOUR DU JOUEUR 1 ◆  
Pouvez-vous encore placer une pièce ? [oui/non] : oui
```

Fonction qui affiche les points ainsi que le nombre de tours effectués

```
≪ Le jeu a duré 3.25 tour.s ≫  
↳ POINTS ↳  
Joueur 1 : -79  
Joueur 2 : -89  
Joueur 3 : -89  
Joueur 4 : -89  
→ Joueur 1 a gagné avec -79 points !  ^(*o▽o*)^
```