

Conditions :

- travail individuel :
2h/semaine encadré + **minimum** 2h en autonomie.
- dépôt moodle obligatoire en fin de **chaque** semaine (samedi soir)(zip de vos codes sources commentés) :
 - o de façon à vérifier votre avancement régulier,
 - o pour permettre à votre chargé de TP de vous recadrer d'une séance sur l'autre.
 - o **en préfixant le nom de l'archive par l'OS** sur lequel le code est portable.
- dépôt moodle de la version finale en fin de semaine calendaire n°51 contenant une archive compressée des codes sources commentés de toutes les étapes travaillées (en indiquant quelles sont celles terminées).
- dépôt à votre chargé de TP d'un rapport de synthèse avec :
 - o le code source de la dernière étape réalisée et parfaitement commentée,
- utilisation du [forum](#) pour poser des questions.

Critères d'évaluation :

- autonomie, prise d'initiatives,
- investissement pendant et en dehors des séances encadrées (travail des parties "**prolongements**"),
- pertinence des algorithmes,
- qualité du découpage fonctionnel et du choix des structures de données,
- pertinence des commentaires,
- les efforts individuels seront valorisés,
- le travail collectif sera pénalisé mais l'entraide est fortement encouragée.

Exemple de grille d'évaluation :

dépôt S46	dépôt S47	dépôt S48	dépôt S49	dépôt S50	dépôt S51(final)	nommage des variables	typage des variables	découpage fonctionnel	découpage en fichiers	qualité des algorithmes	commentaire/documentation/rapport annexe	étape finale atteinte et/ou terminée + prolongements
0,5	0,5	0,5	0,5	0,5	0,5	1	1	2	1	5	2	5

NB : les notes de dépôts sur Moodle sont sur 5 mais seront divisées par 10 ultérieurement.

Sujet :

Le TP comporte 8 étapes de façon à concevoir un petit jeu en console.

Le jeu s'inspire du jeu BLOKUS de [Bernard Tavitian](#) (éditeur [Sekkoïa](#) / distributeur français [Winning Moves](#)).



Etape 1 : Affichage du plateau monochrome :

Le plateau du jeu est composé de 400 emplacements répartis sur 20 lignes et 20 colonnes.
Cette étape permet d'afficher le plateau avec que des emplacements vides représentés par des points.

											1	1	1	1	1	1	1	1	1	2
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T

Etape 2 : Affichage du plateau avec couleurs :

Le plateau sera stocké en mémoire sous la forme d'un tableau 20 x 20.

Le plateau sera initialisé grâce à une fonction de façon à placer des points dans chaque case du tableau.

L'affichage utilisera une scène composée d'un tableau 23lignes x 22colonnes de cases.

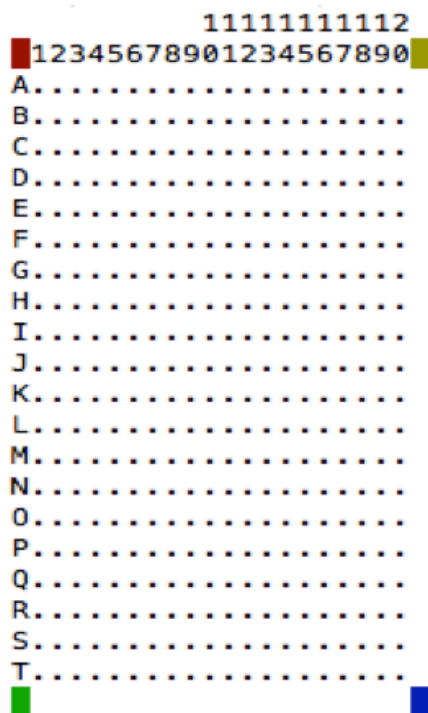
Une case est la combinaison de :

- la notion de contenu (un caractère ascii)
- la notion de couleur de fond.

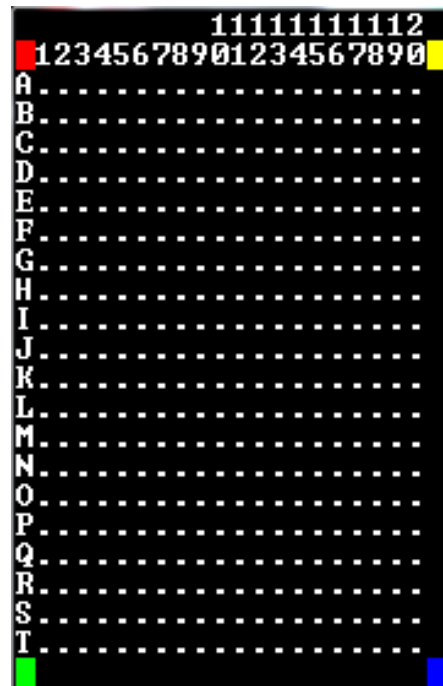
4 marques de couleurs différentes seront ajoutées pour marquer les repères de démarrage de chaque joueur.

Concevoir des fonctions d'initialisation et d'affichage pour obtenir les résultats suivants :

Exemple d'affichage sous Mac/Linux



Exemple d'affichage sous Windows



Ressources pour l'affichage de couleurs sous windows :

```
#include <iostream>
#include <windows.h>

int main(){

    HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
    for (int i=0; i < 256 ; i++){
        SetConsoleTextAttribute(h, i);
        std::cout << i << " : texte\n";
    }
}
```

Ressources pour l'affichage de couleurs sous Mac/Linux :

[lien n°1](#) / [lien n°2](#)

```
#include <iostream>

int main() {
    std::cout << "\033[31m" << "Blokus" << "\033[32m" << " bon courage\n";
    return 0;
}
```

Prolongements :

Le placement des 4 couleurs est aléatoire (les 4 couleurs sont toujours les mêmes mais leur position (angle) est variable) à chaque nouvelle exécution du programme

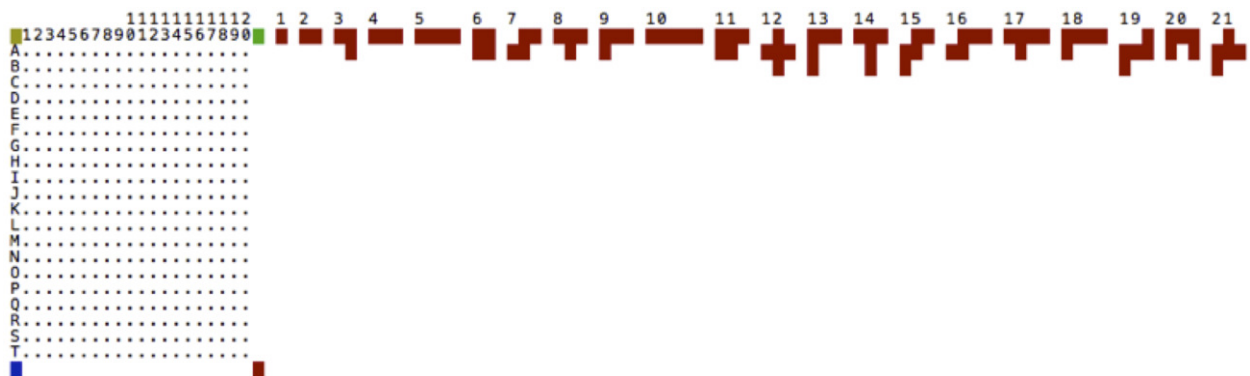
Etape 3 : chargement d'un plateau à partir d'un fichier :

A partir de cette étape,
- toutes les déclarations de fonctions,
- tous les entêtes de fichier,
- toutes les déclarations de structures,
devront être précédés d'un commentaire au format [doxygen](#).

Concevoir une fonction d'initialisation du plateau à partir d'un fichier pieces.txt (fourni sur Moodle) dont le prototype est le suivant :

```
void init(Piece desPieces[21], std::string unNomFichier);
```

L'affichage et la numérotation des pièces chargées se fera sur le côté droit du plateau comme suit :



Remarque : élargissement de la largeur du terminal :

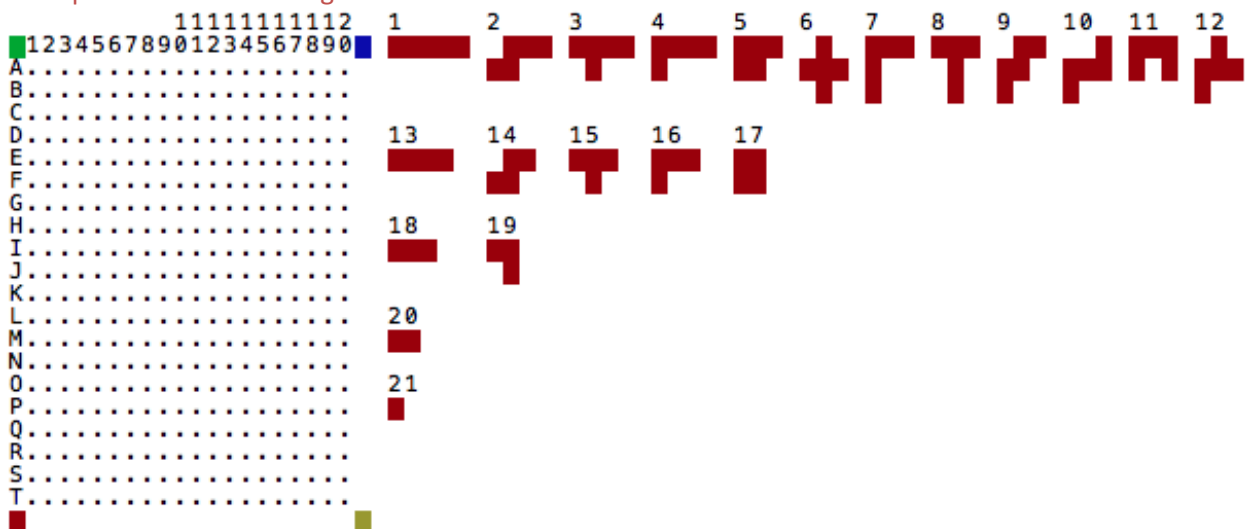
- sous Windows -> clic droit dans la barre de titre -> propriétés -> configuration -> taille de la fenêtre.
- sous MacOS -> menu préférences -> sélectionner un profil -> onglet fenêtre -> taille de la fenêtre.

Prolongements :

Les pièces doivent être affichées alignées les unes par rapport aux autres sous forme de lignes et de colonnes.

La pièce en haut à gauche est celle qui contient le plus de cases.

Deux pièces sur la même ligne contiennent le même nombre de cases.



Etape 4 : Mouvements des pièces en attente :

L'objectif est de pouvoir manipuler sur place une ou plusieurs pièces quand elles ne sont pas encore placées.

Etape 4.1 :

L'utilisateur doit répondre en boucle à 2 questions posées par le programme.

Premièrement : le n° de la pièce à manipuler,

Deuxièmement : la liste des rotations dans le sens horaire (r), miroirs selon un axe horizontal (h), miroirs selon un axe vertical (v).

Exemples :

```

      111111111112 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
12345678901234567890 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
A.....
B.....
C.....
D.....
E.....
F.....
G.....
H.....
I.....
J.....
K.....
L.....
M.....
N.....
O.....
P.....
Q.....
R.....
S.....
T.....
■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
no de la piece a manipuler ? 3
sequence d'action sur la piece (rrh) ? v
      111111111112 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
12345678901234567890 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
A.....
B.....
C.....
D.....
E.....
F.....
G.....
H.....
I.....
J.....
K.....
L.....
M.....
N.....
O.....
P.....
Q.....
R.....
S.....
T.....
■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
no de la piece a manipuler ? 13
sequence d'action sur la piece (rrh) ? hv
      111111111112 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
12345678901234567890 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
A.....
B.....
C.....
D.....
E.....
F.....
G.....
H.....
I.....
J.....
K.....
L.....
M.....
N.....
O.....
P.....
Q.....
R.....
S.....
T.....
■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
no de la piece a manipuler ? |
```

Etape 4.2 :

L'utilisateur peut cette fois saisir en une seule séquence plusieurs mouvements et le programme contrôle les erreurs de formatage :

Exemples :

The image displays three examples of a Go board interface. Each example shows a 19x19 board with a sequence of moves entered. The moves are represented by red and black stones. The sequence is entered in a text field, and the board shows the resulting position. The examples illustrate how the program handles errors in the sequence format.

Example 1: The sequence entered is "12rrh : 2r3h". The board shows a sequence of moves starting with a black stone at 12, followed by a red stone at 13, and so on. The sequence is correct.

Example 2: The sequence entered is "12rrh : 23r2r". The board shows a sequence of moves starting with a black stone at 12, followed by a red stone at 13, and so on. The sequence is correct.

Example 3: The sequence entered is "12rrh : 11g2r". The board shows a sequence of moves starting with a black stone at 12, followed by a red stone at 13, and so on. The sequence is correct.

ici la séquence voudrait sélectionner une pièce qui n'existe pas, toute la fin de la séquence est ignorée même si la fin de séquence est correcte

ici la séquence voudrait faire un mouvement qui n'existe pas, toute la fin de la séquence est ignorée même si la fin de séquence est correcte

Etape 5 : Placement des pièces sur le plateau :

En complément des séquences de l'étape 4, la dernière pièce sélectionnée est placée sur le plateau 20x20 en terminant la séquence par les repères de ligne et de colonne

Voici des exemples de dialogues avec les explications :

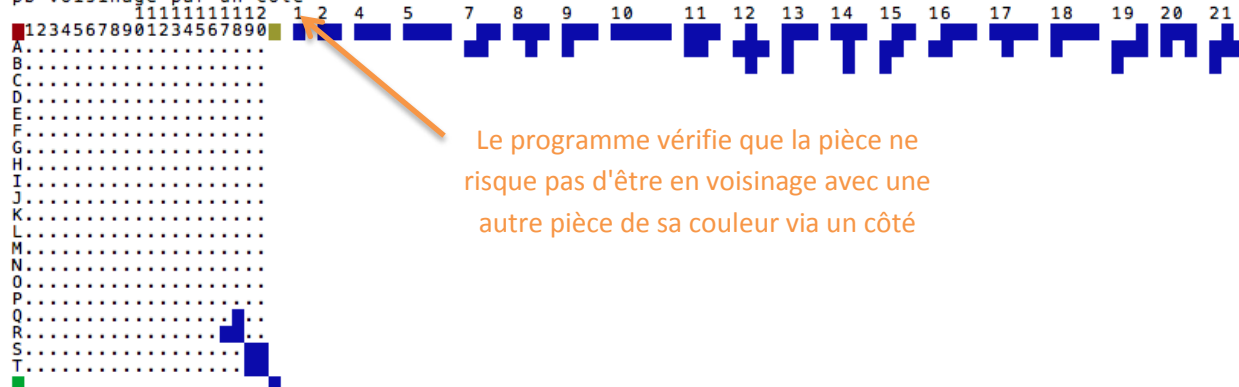
The diagram illustrates the process of placing a piece on a 20x20 board. It shows a list of pieces (A-T) and a sequence input field. The sequence is entered as a string of numbers and letters, representing the piece number, the line, and the column. The board is a 20x20 grid with letters A-T on the left and numbers 1-20 on the top. The sequence input field is labeled "Veuillez entrer une sequence de la forme 12rrh3vR8 ? 6S19".

Annotations and explanations:

- repère de ligne**: Points to the line number in the sequence.
- repère de colonne**: Points to the column number in the sequence.
- n° de la pièce**: Points to the piece number in the sequence.
- Quand une pièce est placée, elle est enlevée de la liste des pièces restantes**: Explains that the piece is removed from the list after placement.
- S19 sera la position de la case supérieure gauche de la pièce**: Points to the starting position of the piece on the board.
- Selon la règle du jeu, la première pièce doit être placée dans le coin attribué au joueur correspondant**: Explains the rule that the first piece must be placed in the player's corner.
- conformément à l'étape4, on peut manipuler la pièce avant de la placer**: Explains that the piece can be moved before placement.
- Q17 sera la position de la case supérieure gauche de la pièce même si cette case est vide**: Points to the starting position of the piece on the board.
- Selon la règle du jeu, la pièce ajoutée possède un voisinage avec une autre pièce via un angle**: Explains the rule that the piece must have an adjacent piece via an angle.
- Le programme vérifie que la pièce est placable et qu'elle ne risque pas de "déborder" du plateau**: Explains that the program checks if the piece can be placed without going off the board.

Veuillez entrer une sequence de la forme 12rrh3vR8 ? 1P18

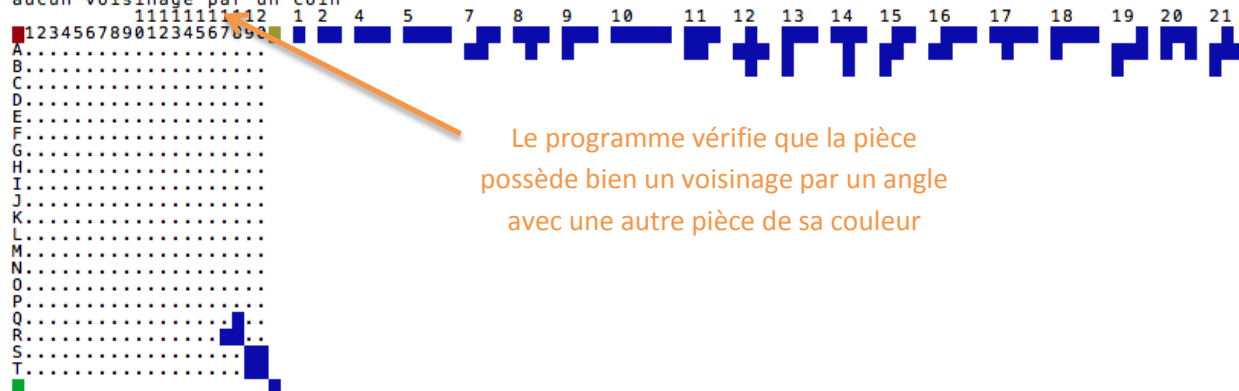
pb voisinage par un côté



Le programme vérifie que la pièce ne risque pas d'être en voisinage avec une autre pièce de sa couleur via un côté

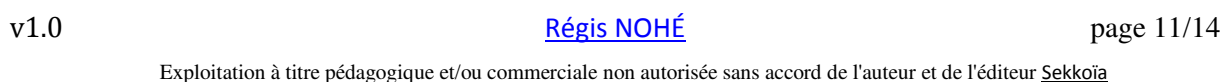
Veuillez entrer une sequence de la forme 12rrh3vR8 ? 1J8

aucun voisinage par un coin



Le programme vérifie que la pièce possède bien un voisinage par un angle avec une autre pièce de sa couleur

Au démarrage du jeu, le programme demande le nombre **N** de joueurs (entre 2 et 4).
Il récupère le nom et/ou prénom de chaque joueur.
Il affecte de façon aléatoire les couleurs BLEU, puis JAUNE, puis au besoin le ROUGE et enfin le VERT.
C'est le joueur ayant la couleur BLEU qui commence (cf. [règle officielle](#) pour l'ordre).



Julien
Veuillez entrer une sequence de la forme 12rrh3vR8 ? 2A1

Thomas
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Remi
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Julien
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Si le joueur qui a la main saisi une
séquence incohérente, il garde la main

Thomas
Veuillez entrer une sequence de la forme 12rrh3vR8 ? 23h

Thomas
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Remi
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Julien
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Si le joueur qui a la main essaie de saisir 2
séquences à suivre (collées), cela se
traduit comme une seule séquence
incohérente et il garde la main

Thomas
Veuillez entrer une sequence de la forme 12rrh3vR8 ? 1R182T17

Thomas
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Remi
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Julien
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Si le joueur qui a la main essaie de saisir 2
séquences à suivre (non collées), seule la
première séquence est validée.

Thomas
Veuillez entrer une sequence de la forme 12rrh3vR8 ? 1R18 2T17

Thomas
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Remi
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Julien
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Thomas
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Remi
Veuillez entrer une sequence de la forme 12rrh3vR8 ? |

Si un joueur se trouve bloqué, le programme détecte qu'il ne peut plus jouer et il doit systématiquement passer son tour :

Remi
1 2 3 4 5 7 9 12 14 17 19 20 21

Julien
9 10 11 12 13 14 15 16 17 18 19 20 21

Thomas
2 4 6 7 8 11 12 16 17 19 20 21

Remi
Veuillez entrer une sequence de la forme 12rrh3vR8 ? 14K2

Thomas
2 4 6 7 8 11 12 16 17 19 20 21

Remi
1 2 3 4 5 7 9 12 17 19 20 21

Julien
9 10 11 12 13 1

Normalement, cela aurait dû être le tour du joueur avec les pièces rouges, mais comme il est bloqué, la partie est finie pour lui. C'est le joueur avec les pièces bleues qui reprend la main.

Thomas
Veuillez entrer une sequence de la forme 12rrh3vR8 ? |

La partie se termine quand tous les joueurs sont bloqués ou qu'il n'y plus de pièce à placer pour aucun joueur.

Les scores sont alors affichés par ordre de classement selon les critères de la [règle officielle](#) :

Remi
1 2 3 4 5 7 9 12 14 17 19 20 21

Julien
9 10 11 12 13 14 15 16 17 18 19 20 21

Thomas
2 4 6 7 8 11 12 16 17 19 20 21

Classement :

1 : Remi , 20pts

2 : Thomas , 15pts

3 : Julien , -64pts

Etape 7 : Sauvegarde/Restauration :

Les noms sont saisis avec ou sans majuscule, avec éventuellement les caractères espaces, tiret et apostrophe, sans accent.

Le joueur qui a la main peut demander à sauver la partie en saisissant le caractère **S**.

Le fichier de sauvegarde **blokus.data** contiendra toutes les informations utiles à la reprise future de la partie.

Le fichier de sauvegarde devra être le plus petit possible.

exemple d'ordre de grandeur :

Pour 2 joueurs dont les noms sont sur 10 caractères chacun, la taille maximale du fichier blokus.data doit être de 86 octets.

On peut recharger une partie en saisissant le caractère **R** et le programme relance la partie là où elle s'était arrêtée.

Etape 8 : Jeu contre l'ordinateur :

Le programme propose au lancement de choisir de jouer contre l'ordinateur.

Dans l'affirmative, le joueur choisi son nombre d'adversaire(s) (entre 1 et 3).

Tous les adversaires sont gérés par l'ordinateur.