

TP LE JEU DU MORPIONS DANS LE LANGAGE DE PROGRAMMATION GO

Présentation du TP

Il est temps de **pratiquer** un peu !

Je vous ai préparé un **TP** qui reprend la base de tout ce qu'on a pu étudier jusqu'ici.

Nous allons créer un **jeu de morpion** appelé aussi le **tic tac toe** avec le langage de programmation Go.

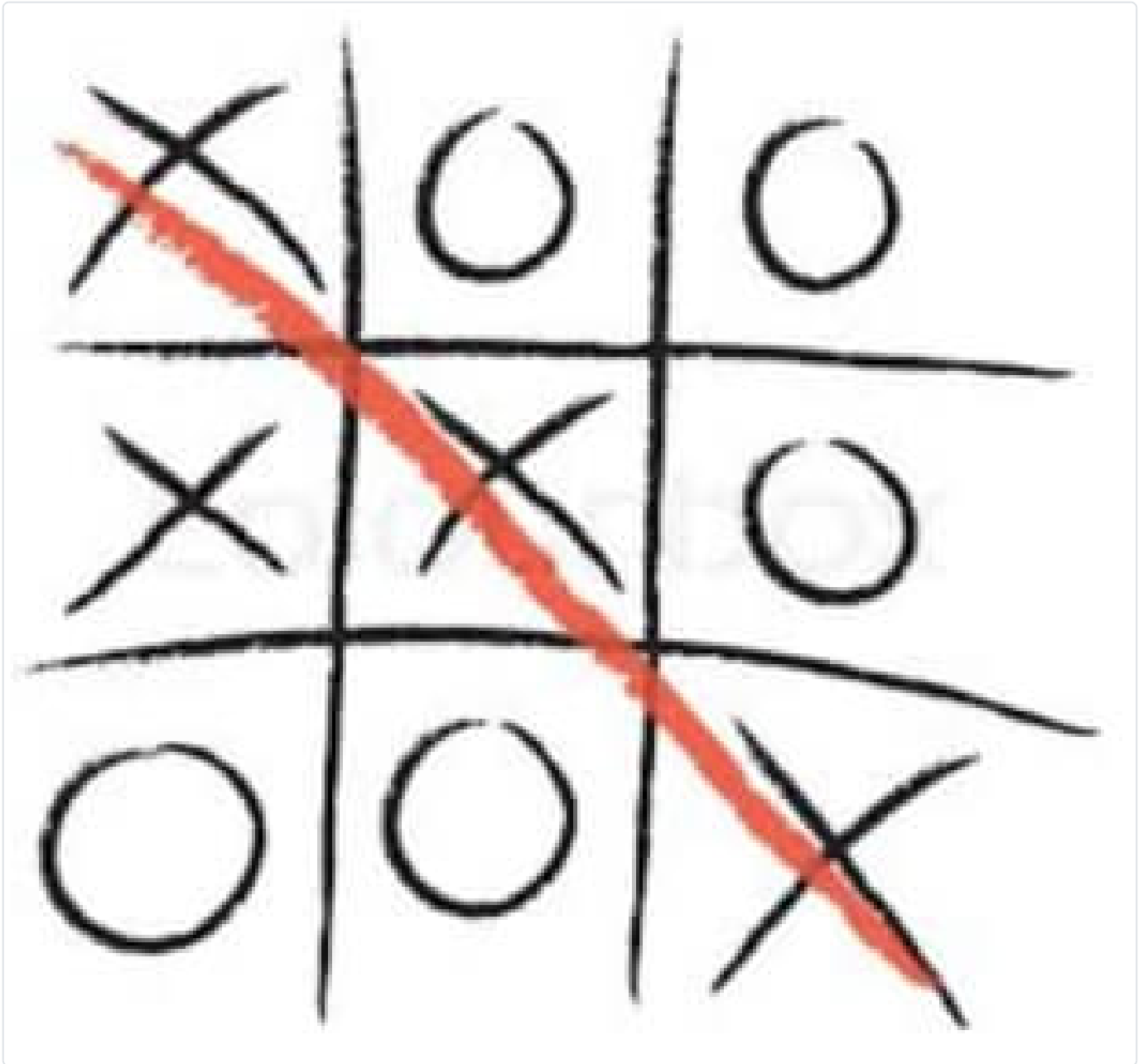


Image du jeu de morpion

Avant de commencer la création du jeu voici déjà quelques règles à respecter :

- Le jeu se déroulera sur un damier de 3 cases par 3 cases
- Le jeu se joue à deux. Un premier joueur dessine son symbole sur une case.
Puis c'est au tour de l'autre joueur de dessiner son symbole sur une case libre.

- Pour gagner il faut aligner 3 symboles identiques horizontalement, verticalement ou en diagonale avant le tour de l'adversaire
- Quand toutes les cases sont remplies alors il y a match nul.
- Vérifier toujours si l'utilisateur a rentré les valeurs attendues "Never trust user input" !
- Le jeu va avoir lieu sur l'invite de commande.

Libre à vous de choisir le design que vous souhaitez, en ce qui me concerne voici à quoi ressemble mon jeu :

```

1  2  3
4  5  6
7  8  9
Joueur 1 entrez un nombre compris entre 1 à 9 : 1
X  2  3
4  5  6
7  8  9
Joueur2 entrez un nombre compris entre 1 à 9 : 2
X  0  3
4  5  6
7  8  9
Joueur 1 entrez un nombre compris entre 1 à 9 : 5
X  0  3
4  X  6
7  8  9
Joueur2 entrez un nombre compris entre 1 à 9 : 6
X  0  3
4  X  0
7  8  9
Joueur 1 entrez un nombre compris entre 1 à 9 : 9
Joueur 1  vous avez gagné !

```

Information

Ici les chiffres correspondent à des cases libres. Ça permet à l'utilisateur de savoir si sa case est libre. Libre à vous bien sûr de mettre autre chose à la place.

Gestion des différents cas

Voici la sortie standard des différents cas d'usage :

- Si le joueur rentre autre chose que ce que je souhaite :

```
1 2 3
4 5 6
7 8 9
Joueur 1 entrez un nombre compris entre 1 à 9 : 15
Votre nombre doit être compris entre 0 à 9 !
Joueur 1 entrez un nombre compris entre 1 à 9 : dsdsds
Entrez un nombre et non autre chose !
Joueur 1 entrez un nombre compris entre 1 à 9 : 1
X 2 3
4 5 6
7 8 9
Joueur2 entrez un nombre compris entre 1 à 9 : 1
Cette case est déjà prise !
```

- Si il y a match nul :

```
1 2 3
4 5 6
7 8 9
Joueur 1 entrez un nombre compris entre 1 à 9 : 1
X 2 3
4 5 6
7 8 9
Joueur2 entrez un nombre compris entre 1 à 9 : 4
X 2 3
O 5 6
7 8 9
Joueur 1 entrez un nombre compris entre 1 à 9 : 7
X 2 3
O 5 6
X 8 9
Joueur2 entrez un nombre compris entre 1 à 9 : 5
X 2 3
O O 6
```

```

X 8 9
Joueur 1 entrez un nombre compris entre 1 à 9 : 2
X X 3
O O 6
X 8 9
Joueur2 entrez un nombre compris entre 1 à 9 : 3
X X O
O O 6
X 8 9
Joueur 1 entrez un nombre compris entre 1 à 9 : 6
X X O
O O X
X 8 9
Joueur2 entrez un nombre compris entre 1 à 9 : 9
X X O
O O X
X 8 O
Joueur 1 entrez un nombre compris entre 1 à 9 : 8
X X O
O O X
X X O
Partie nulle !

```

Sachez que vous possédez les connaissances nécessaires vues dans les chapitres précédents, et donc vous êtes tout à fait capables de réaliser ce jeu.

Good Luck ! et à vous de jouer !

Solution

J'espère que vous avez réussi à réaliser ce tp ! Même si vous n'avez pas forcément réussi à tout faire. L'essentiel c'est qu'au moins quelques fonctionnalités du jeu fonctionnent, ça restera toujours mieux que de voir la solution directement sans même essayer.

Je vous présente ici ma solution avec beaucoup de commentaires. J'ai essayé de reprendre tout ce qu'on a pu découvrir jusqu'ici. Je ne détiens pas la meilleure solution donc n'hésitez pas à **améliorer le code** en rajoutant ou modifiant les fonctionnalités déjà présentes et de partager votre code dans les commentaires avec un lien de votre repository github ou autres !

```

package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
)

// constantes globales
const (
    tailleDamier    = 9
    symboleJoueur1  = "X"
    symboleJoueur2  = "O"
)

// Variables globales
var (
    tableauMorpion = [tailleDamier]string{ // création d'un damier sans aucune cases
        "1", "2", "3",
        "4", "5", "6",
        "7", "8", "9"}
    joueur1 = true // c'est le joueur 1 qui commence en 1er
)

func main() {
    jouer() // Lancement du jeu
}

/**
 * description de la fonction : Permet de lancer le jeu
 *
 * @return rien
 */
func jouer() {

    var numeroCase int
    for true {
        affichage()
        numeroCase = gestionEntreeUtilisateur() // Récupération de l'entrée utilisateur
        remplirCase(numeroCase)
        if gagner() { // Vérifier si le joueur a gagné
            affichage()
            fmt.Println(nomJoueur(), "vous avez gagné !")
            os.Exit(0) // on quitte la partie
        } else if partieNulle() { // Vérifier si match nul
            affichage()
            fmt.Println("Partie nulle !")
            os.Exit(0) // on quitte la partie
        }
        joueur1 = !joueur1 // on change de joueur
    }
}

```

```

/**
 * description de la fonction : Permet d'afficher le damier en prenant en compte les c
 *
 * @return rien
 */
func affichage() {
    for i := 0; i < len(tableauMorpion); i++ {
        fmt.Print(" ", tableauMorpion[i], " ")
        if (i+1)%3 == 0 { // retour à la ligne après avoir affiché 3 éléments
            fmt.Println()
        }
    }
}

/**
 * description de la fonction : Retourne le nom des joueurs
 *
 * @return string
 */
func nomJoueur() string {
    if joueur1 {
        return "Joueur 1 "
    } else {
        return "Joueur2 "
    }
}

/**
 * description de la fonction : Permet de vérifier l'entrée utilisateur
 *
 * @return int : retourne l'entrée utilisateur
 */
func gestionEntreeUtilisateur() int {

    var (
        bonneEntree = false // variable qui permet de vérifier si l'utilisateur a ren
        numeroCase  = 0
        err          = error
        scanner      = bufio.NewScanner(os.Stdin)
    )

    for bonneEntree == false {
        fmt.Print(nomJoueur(), "entrez un nombre compris entre 1 à ", tailleDamier, "
        scanner.Scan()
        numeroCase, err = strconv.Atoi(scanner.Text())
        if err != nil { //vérifier si l'utilisateur a rentré un nombre
            fmt.Println("Entrez un nombre et non autre chose !")
        } else if numeroCase < 1 || numeroCase > tailleDamier {
            fmt.Println("Votre nombre doit être compris entre 0 à", tailleDamier, "!")
        } else if tableauMorpion[numeroCase-1] == symboleJoueur1 || tableauMorpion[nu
            fmt.Println("Cette case est déjà prise !")
        } else {
            bonneEntree = true

```

```

    }
}
return numeroCase - 1 // n'oubliez pas que la taille d'un tableau commence toujours à 0
}

/**
 * description de la fonction : Permet de remplir la case choisie par le joueur
 *
 * @param numeroCase
 * @return rien
 */
func remplirCase(numeroCase int) {
    if joueur1 {
        tableauMorpion[numeroCase] = symboleJoueur1
    } else {
        tableauMorpion[numeroCase] = symboleJoueur2
    }
}

/**
 * description de la fonction : Permet de savoir si le joueur a gagné
 *
 * @return bool
 */
func gagner() bool {

    /*
     * tableauxdeGain est un tableau à double dimensions où j'ai rajouté
     * les différents cas d'utilisation où il est possible de gagner.
     */
    tableauxdeGain := [][][tailleDamier]bool{
        {
            true, true, true,
            false, false, false,
            false, false, false},
        {
            false, false, true,
            false, false, true,
            false, false, true},
        {
            false, false, false,
            false, false, false,
            true, true, true},
        {
            true, false, false,
            true, false, false,
            true, false, false},
        {
            true, false, false,
            false, true, false,
            false, false, true},
        {
            false, false, true,

```



```

        false, true, false,
        true, false, false},
    {
        false, true, false,
        false, true, false,
        false, true, false}}

// création d'un damier temporaire
var tableauMorpionBool [tailleDamier]bool

for index, valeur := range tableauMorpion {
    if joueur1 && valeur == symboleJoueur1 { // si c'est le tour du joueur 1 et q
        tableauMorpionBool[index] = true
    } else if !joueur1 && valeur == symboleJoueur2 {
        tableauMorpionBool[index] = true
    }
}

ressemblance := 0 // Nombre de true qui sont sur les mêmes cases dans le tableau
for _, tableauGain := range tableauxdeGain {
    for i := 0; i < len(tableauMorpionBool); i++ {
        if tableauMorpionBool[i] == true && tableauMorpionBool[i] == tableauGain[
            ressemblance++
            if ressemblance == 3 { // si les cases du tableau tableauMorpionBool
                return true
            }
        }
    }
    ressemblance = 0 // On remet le compteur à 0 pour vérifier un autre tableau d
}
return false
}

/*
* description de la fonction : Permet de vérifier si la partie est nulle
*
* @return bool
*/
func partieNulle() bool {
    occurence := 0

    for _, valeur := range tableauMorpion {
        if valeur == symboleJoueur1 || valeur == symboleJoueur2 {
            occurence++ // incrémenter de 1 si une case est remplite par un symbole
        }
    }

    /*
    si toutes les cases sont remplies de symboles
    et que le joueur n'a pas encore gagné alors la partie est nulle
    */
    return (occurence == len(tableauMorpion))
}

```