



SWIFT UI

MORPION



Présenté par Ethan Reina, Adamou Fatihou et Thomas Cheio (Groupe 9)

SEPTEMBRE 2023

Mr Fouques



SOMMAIRE

- Présentation du projet
- Affichage de la grille
- Fonctionnement du système de jeu
- Les problèmes
- Source
- Conclusion

1 - Contraintes et Résultat attendu:

On doit donc réaliser une application de Morpion en Swift UI 

On s'est donc fixé plusieurs objectifs avant de commencer à réfléchir à notre application:



Une appli fonctionnel



Simplicité d'utilisation



Possibilité d'évolution

2 - Travail sur le projet

Contribution des membres :

- Thomas : Grille, implémentation des ronds et croix et design de l'application
- Fatihou : Conditions de fin de parties et commentaires
- Ethan : Recherche, test fonctionnel, source et présentation



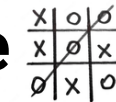
3- Affichage de la grille

Avant de commencer à développer l'application, nous avons commencer par réfléchir à la façon dont nous allons faire la grille



01

Image grille



Superposer les images de ● et de ✕ par dessus une image de grille.

Problème: Placer une image sur une autre se révèle plus compliqué que prévu car les images s'affiche les unes à la suite des autres

02

Tableau de boutons

Former un tableau de 3 x 3 de bouton qui une fois un clique effectuer dessus se transformerai en image en fonction du joueur.

```
@State private var grille = Array(repeating: "", count: 9) // variable grille
```

4 - Fonctionnement du système de jeu



```
LazyVGrid(columns: [GridItem(), GridItem(), GridItem()]) {  
  ForEach(0..<9) { index in // 9 cases  
    Button(action: {  
      if grille[index] == "" && !gameOver { // l'index sert a pointer les une case  
        grille[index] = joueur  
        checkVictoire()  
        joueur = (joueur == "X") ? "O" : "X"  
      }  
    }) {  
  
      if (grille[index] == "X"){ // le joueur X joue  
        Image("croix 1") // l'image de la croix sera sélectionnés  
          .resizable()  
          .padding()  
          .frame(width: 80.0, height: 80.0)  
      }  
  
      else if (grille[index] == "O") {  
        Image("rond")  
          .resizable()  
          .padding()  
          .frame(width: 80.0, height: 80.0)  
      }  
  
      else {  
        Text(grille[index]) // les config des case de a gille  
          .font(.system(size: 60))  
          .frame(width: 80, height: 80)  
          .background(Color.orange)  
          .foregroundColor(.white)  
          .cornerRadius(10)  
      }  
    }  
  }  
}
```

4 - Fonctionnement du système de jeu

```
private func checkVictoire() { // la fonction qui verifie si une des conditions de fin de jeu est remplis
    let victoire: [[Int]] = [[0, 1, 2], [3, 4, 5], [6, 7, 8], [0, 3, 6], [1, 4, 7], [2, 5, 8], [0, 4, 8], [2, 4, 6]]

    for pattern in victoire {
        if grille[pattern[0]] == joueur && grille[pattern[1]] == joueur && grille[pattern[2]] == joueur { //paterne de victoir
            gameOver = true
            return
        }
    }

    if !grille.contains("") {
        gameOver = true
    }
}
```


5 - Problèmes rencontrés:

- La détection de victoire, bien que le jeu s'arrête directement lorsque l'un des deux joueurs gagne ou qu'il y a une égalité, nous n'avons pas réussi à faire en sorte que l'égalité soit détectée, la victoire du joueur X et O l'est sans soucis mais l'égalité est systématiquement prise comme étant la victoire du dernier joueur qui a placé son symbole.

```
if gameOver { //affichage quand une game est terminée (Regler la condition de victoire)

    if (joueur == "X" && gameOver){
        Text("Le gagnant est le joueur O")
        .font(.title)
        .foregroundColor(.red)
    }

    else if (joueur == "O" && gameOver){
        Text("Le gagnant est le joueur X")
        .font(.title)
        .foregroundColor(.red)
    }

    else {
        Text("Egalité")
        .font(.title)
        .foregroundColor(.red)
    }

    Text("Partie terminée!")
    .font(.title)
    .foregroundColor(.red)
}
```

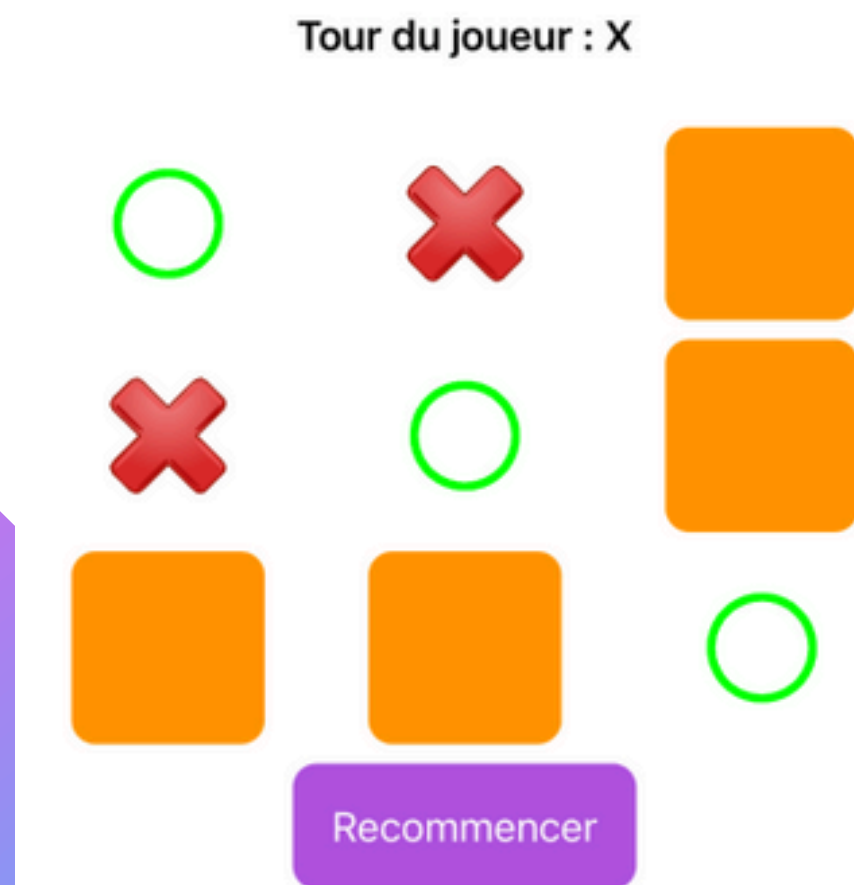

5 - Problèmes rencontrés:

- L'affichage des images de X et O a également été problématique, il a fallut prendre en compte dans la construction même de la grille l'affiche des images de X et O.

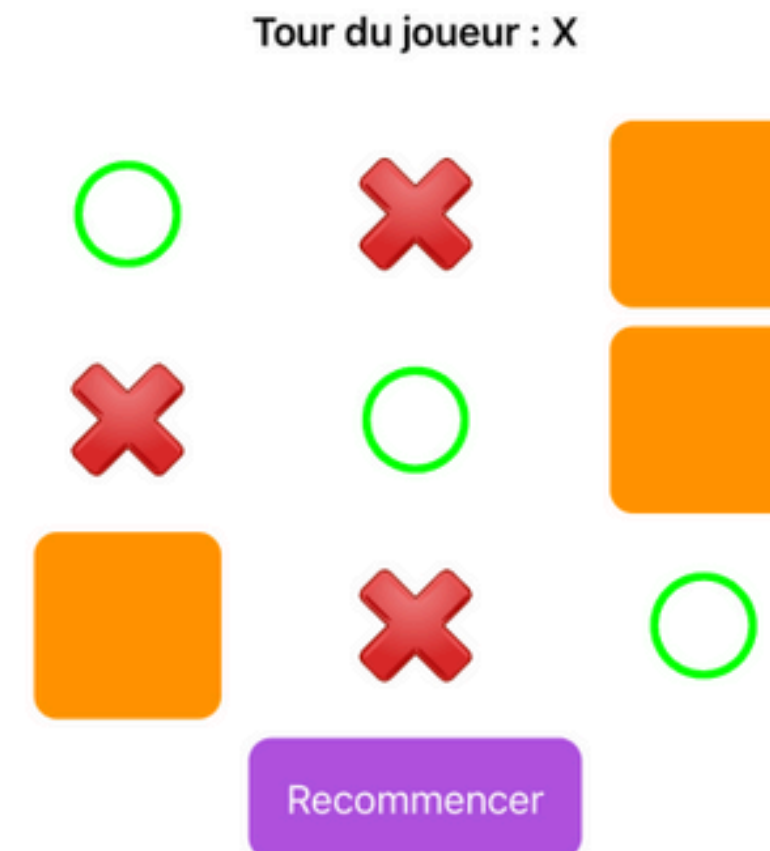
```
LazyVGrid(columns: [GridItem(), GridItem(), GridItem()]) {  
  ForEach(0..<9) { index in // 9 cases  
    Button(action: {  
      if grille[index] == "" && !gameOver { // l'index sert a pointer les une case  
        grille[index] = joueur  
        checkVictoire()  
        joueur = (joueur == "X") ? "O" : "X"  
      }  
    }) {  
  
      if (grille[index] == "X"){ // le joueur X joue  
        Image("croix 1") // L'image de la croix sera sélectionnés  
          .resizable()  
          .padding()  
          .frame(width: 80.0, height: 80.0)  
      }  
  
      else if (grille[index] == "O") {  
        Image("rond")  
          .resizable()  
          .padding()  
          .frame(width: 80.0, height: 80.0)  
      }  
  
      else {  
        Text(grille[index]) // les config des case de a gille  
          .font(.system(size: 60))  
          .frame(width: 80, height: 80)  
          .background(Color.orange)  
          .foregroundColor(.white)  
          .cornerRadius(10)  
      }  
    }  
  }  
}
```

5 - Problèmes rencontrés:

- Le dernier problème majeur que nous avons rencontrés est lorsqu'un joueur réussit à gagner la partie, cette dernière n'était pas immédiatement stoppé mais demandait encore au joueur suivant de jouer un coup avant de détecté qu'un joueur avait gagné la partie.



**Partie toujours en cours
malgré victoire**



Partie terminé


6 - Possibilité d'évolution

Nous avons déjà imaginé plusieurs évolution possible pour cette application:

- Ajout de la Victoire et Egalité
- Ajout d'un mode de jeu jouant contre l'ordinateur
- Ajout d'animation de victoire



SOURCE

 Plusieurs prompts Chat GPT permettant de créer la base du jeu (Grille basique et aide à la mise en place du système de jeu) [Lien Prompts](#)

 Documentation Apple:

<https://developer.apple.com/documentation/swiftui/griditem>

<https://developer.apple.com/documentation/swiftui/lazyvgrid>

Autres sources

<https://www.hackingwithswift.com/read/fr/1/4/concevoir-un-ecran-affichant-une-image>



**MERCI POUR
VOTRE ATTENTION**