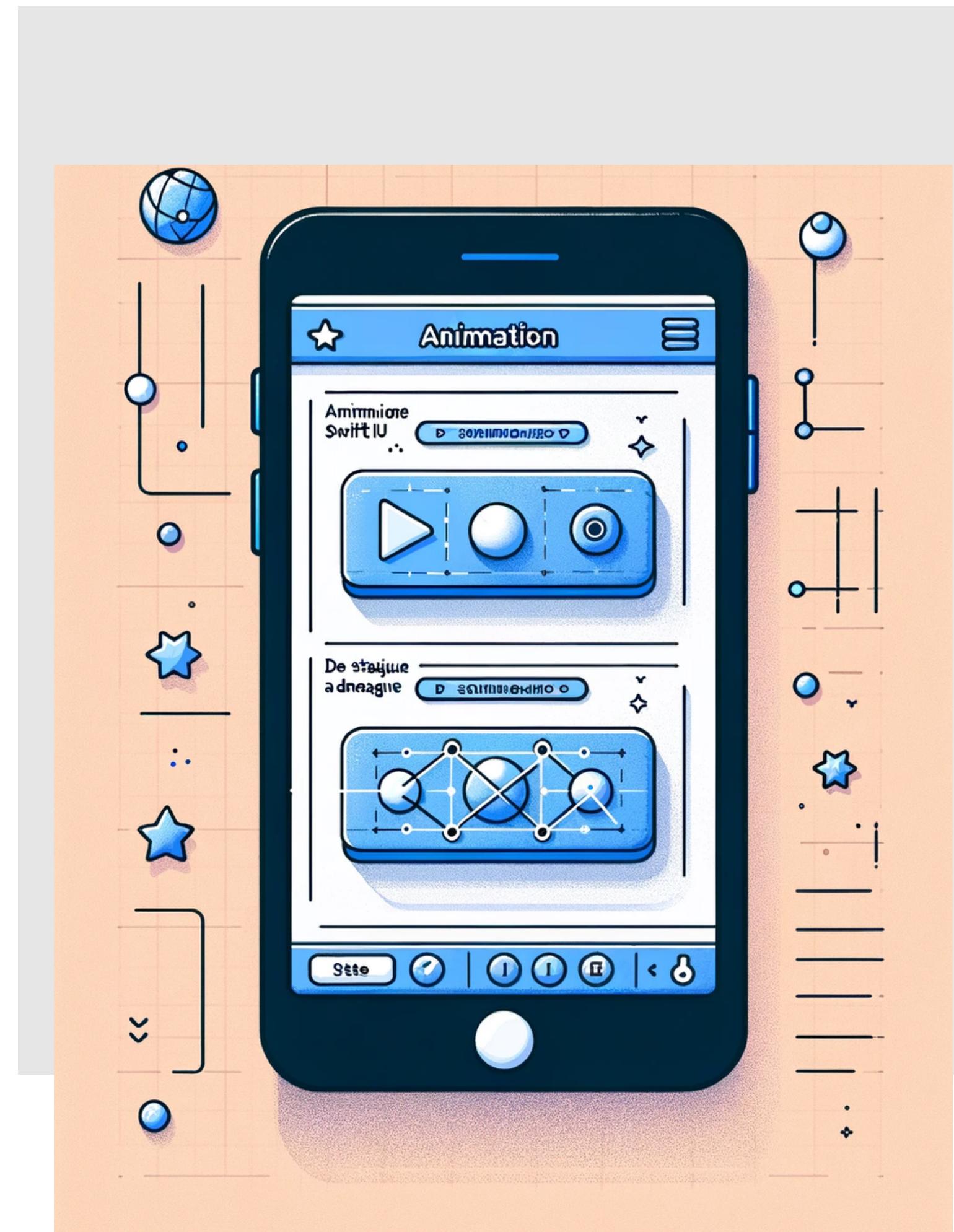


# Animations

# Swift UI





# Introduction

---

Plongeons dans l'univers captivant des animations avec SwiftUI, où chaque interaction devient une expérience fluide et engageante, transformant les interfaces statiques en aventures visuelles dynamiques.



# Animations

---

Les animations sont directement liées aux changements d'état de votre application. En utilisant le système de binding et les propriétés `@State`, chaque fois qu'une valeur d'état change, vous pouvez choisir de la faire de manière animée. SwiftUI propose des animations implicites et explicites. Avec une animation implicite, vous définissez simplement une animation par défaut pour une vue qui s'appliquera à tous les changements d'état. Pour plus de contrôle, vous pouvez utiliser des animations explicites pour déclencher des animations spécifiques en réponse à des actions utilisateur, comme des gestes.

# Exemple

---

```
struct FlipCardView: View {
    @State private var isFlipped = false

    var body: some View {
        VStack {
            Spacer()
            Rectangle()
                .fill(isFlipped ? Color.blue : Color.red)
                .frame(width: 200, height: 100)
                .overlay(Text(isFlipped ? "Bonjour !" : "Tappez pour flipper")
                    .font(.headline)
                    .foregroundColor(.white))
                .rotation3DEffect(
                    .degrees(isFlipped ? 180 : 0),
                    axis: (x: 0, y: 1, z: 0))
            .onTapGesture {
                withAnimation(.easeInOut(duration: 1)) {
                    isFlipped.toggle()
                }
            }
            Spacer()
        }
    }
}
```

Lorsque l'utilisateur appuie sur la carte, la propriété `isFlipped` change, ce qui déclenche l'animation. La carte effectue une rotation de 180 degrés autour de son axe vertical, créant ainsi un effet de retournement. L'animation est douce et progressive grâce à `.easeInOut`, rendant l'interaction à la fois agréable et intuitive.