

Swift UI Animations

Comment fonctionne une animation avec SwiftUI et comment s'en servir ?

Jules Silvestri

Etudiant en Informatique

23 Janvier 2024

IUT Annecy

Introduction

Swift UI ?

SwiftUI offre des fonctionnalités puissantes pour créer des animations fluides dans les applications iOS.

Pourquoi des animations ?

Les animations améliorent l'expérience utilisateur en rendant l'interface plus dynamique et attrayante.

Comment les utiliser ?

Comment utiliser SwiftUI pour intégrer des animations dans un projet.



Principes fondamentaux des animations SwiftUI

Sur quoi ?

Les animations sont basées sur des transitions entre les états de la vue.

Avec quoi ?

Les propriétés telles que `opacity` et `position`, peuvent être animées en modifiant leur valeur au fil du temps.

Comment ?

Utilisez le modificateur `.animation` pour ajouter une animation à une vue.



Gestion des états pour des animations plus complexes

Les variables @State

Utilisation des variables d'état (@State) pour déclencher des animations en fonction des changements d'état.

Déclenchement de l'animation

Changement la valeur de @State pour déclencher l'animation.



Exemple pratique - Bouton animé

```
12 struct AnimatedButton: View {
13     @State private var isAnimated = false
14
15     var body: some View {
16         Button(action: {
17             withAnimation {
18                 self.isAnimated.toggle()
19             }
20         }) {
21             Text("Appuyez-moi")
22                 .padding()
23                 .foregroundColor(.white)
24                 .background(isAnimated ? Color.green : Color.blue)
25                 .cornerRadius(10)
26         }
27     }
28 }
```

Cas concret

Ce code SwiftUI définit une vue de bouton animé appelée `AnimatedButton` avec un état interne (`isAnimated`) qui, lorsqu'il est basculé en cliquant dessus, déclenche une animation fluide faisant passer la couleur de fond du bouton de bleu à vert, le tout encapsulé dans une interface utilisateur déclarative.

