Animations en SwiftUi

Que couvrira cette présentation ?

- Les bases des animations avec SwiftUI.
- Des exemples pratiques.

Contexte

Les animations se déroulent entre l’état initial et l’état final de la vue

L’état initial est la vue telle qu’elle est affichée après le chargement de tout les éléments et l’état final est la vue telle qu’elle est affichée après l’animation de ces derniers

1. Animations de base

Les animations de base consistent à changer une propriété d'un élément en lui ajoutant un type d'animation, une transition. Par défaut, l'animation commence lentement, s'accélère et se termine lentement.
Les animations gèrent les différents changements de propriétés des éléments, elles peuvent devenir assez complexe si l'on change beaucoup de propriétés pour une seule anim.
Ici, l'animation fait faire une rotation à l'élément avec un angle définit.

```
struct ContentView: View {
    @State private var angle: Double = 0

    var body: some View {
        Button("Press here") {
            angle += 45
        }
        .padding()
        .rotationEffect(.degrees(angle))
        .animation(.easeIn)
    }
}
```

Outre les animations de base, SwiftUI prend en charge d'autres types d'animation prédéfinies comme l'animation string qui ajoute un effet de ressort à l'anim.

```swift
import SwiftUI

struct ContentView: View {
    @State private var angle: Double = 0

    var body: some View {
        Button("Press here") {
            angle += 45
        }
        .padding()
        .rotationEffect(.degrees(angle))
        .animation(.spring())
    }
}
```

Conclusion

Dans SwiftUI, l'animation n'est rien d'autre que le changement d'état du début à la fin avec différentes courbes et vitesses. La syntaxe est généralement facile à comprendre et rapide à mettre en œuvre. L'animation est assez facile avec l'utilisation de différentes animations implicites comme easeIn, easeOut, linéaire, spring, et interpolating spring. On peut créer des animations sympa avec quelques lignes de code.