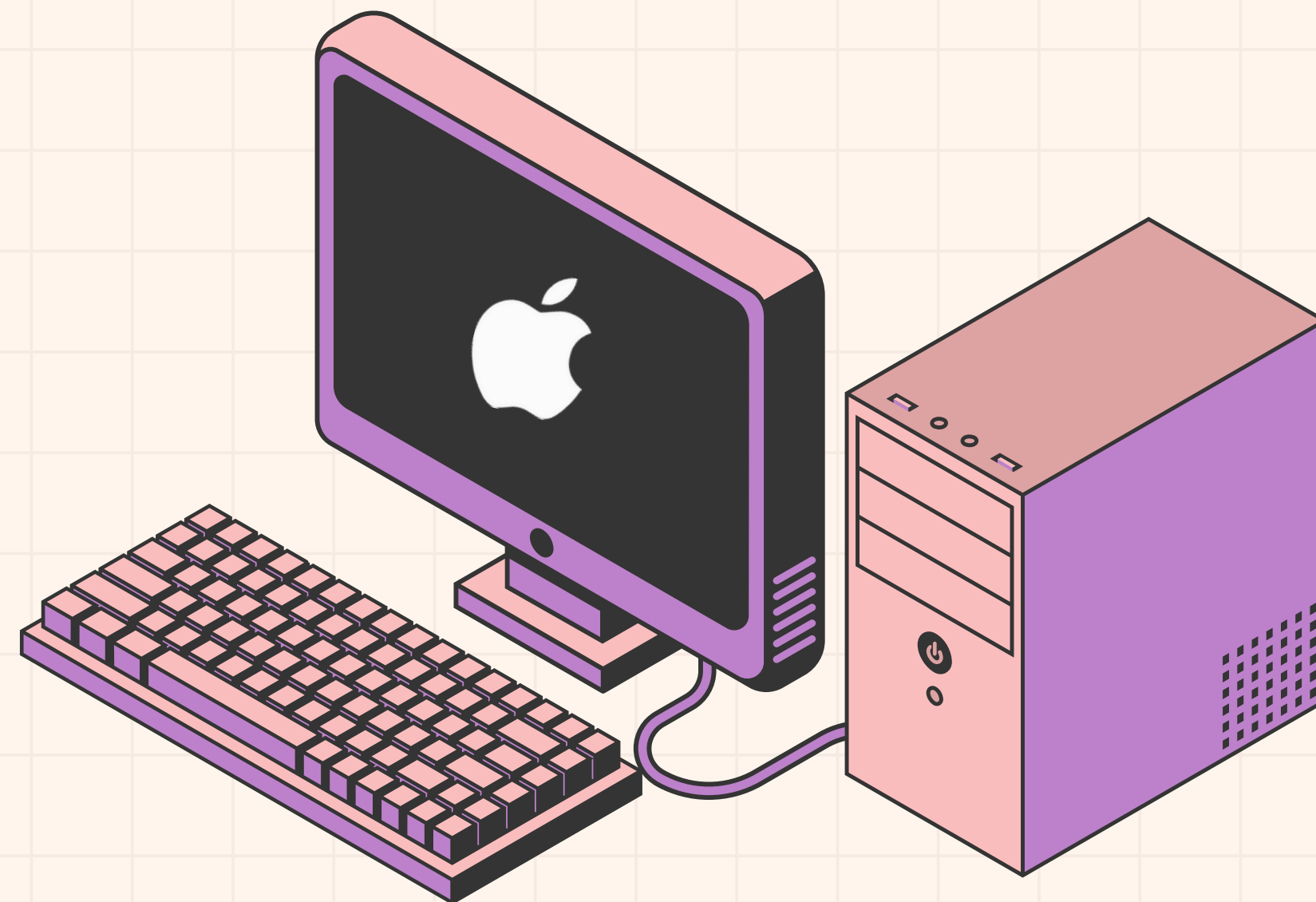


SWIFTUI ANIMATIONS

Présentation des concepts de base des animations dans SwiftUI



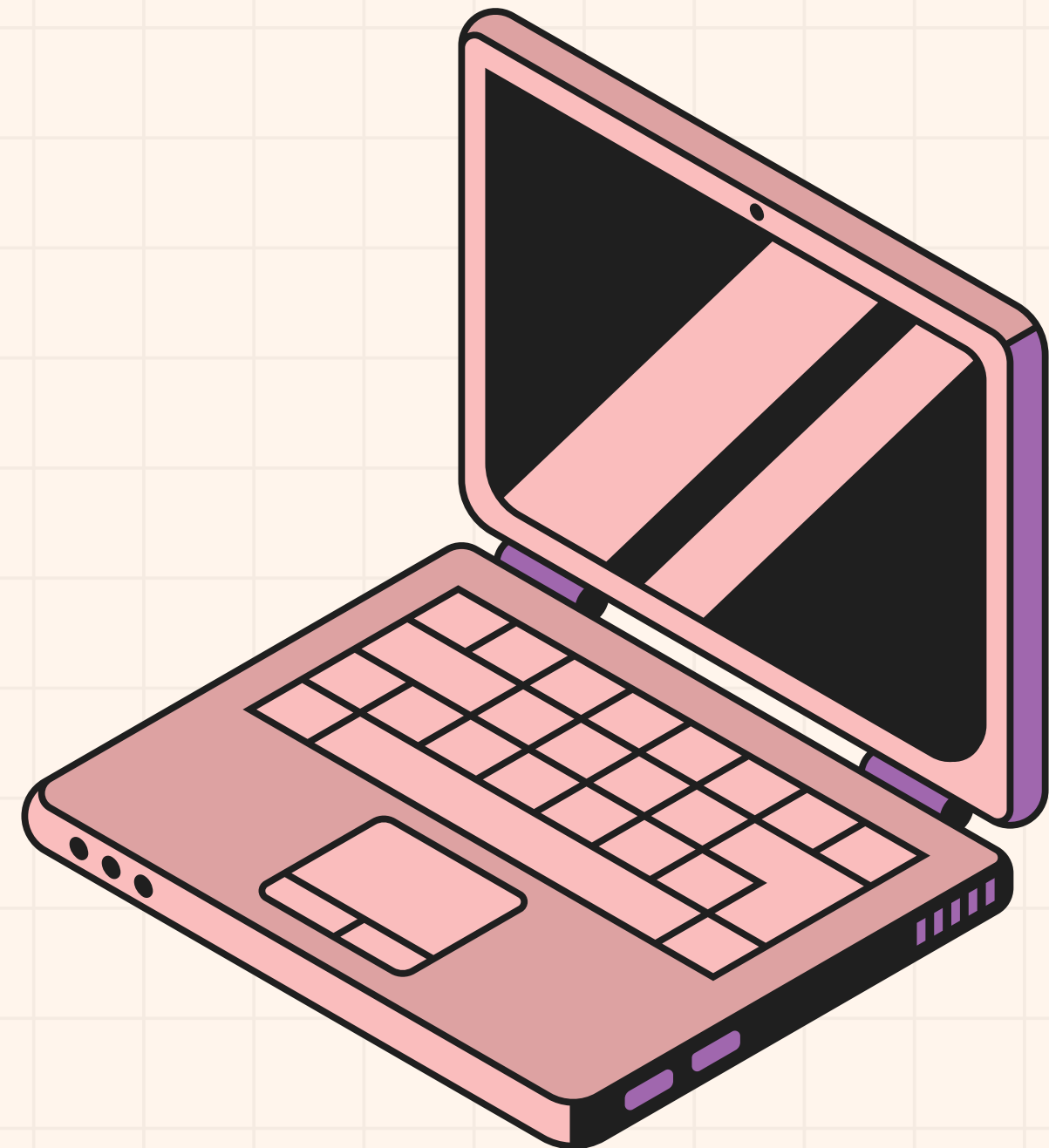
EST-CE QUE C'EST BIEN ?

SwiftUi est assez intuitif pour faire des animations, Apple voulait vraiment rendre ça facile à utiliser.

Points fort de SwiftUI :

- Syntaxe simple et intuitive.
- Animations intégrées directement dans la structure de la vue.
- Prise en charge des animations implicites et explicites.

Donc oui c'est bien pour faire des animations, maintenant on va voir comment on fait →

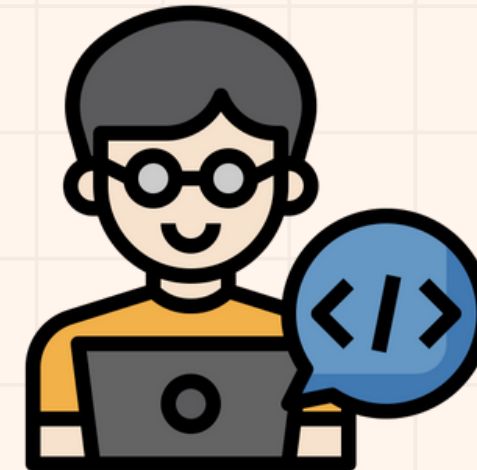


IMPLICITE VS EXPLICITE



ANIMATION IMPLICITE

Les animations implicite sont automatiquement gérées par SwiftUI. Une fois définie avec **.animation** ou **.transaction**, elles se joueront automatiquement lors de l'apparition de l'objet visé. C'est le moyen le plus simple de faire des animations.

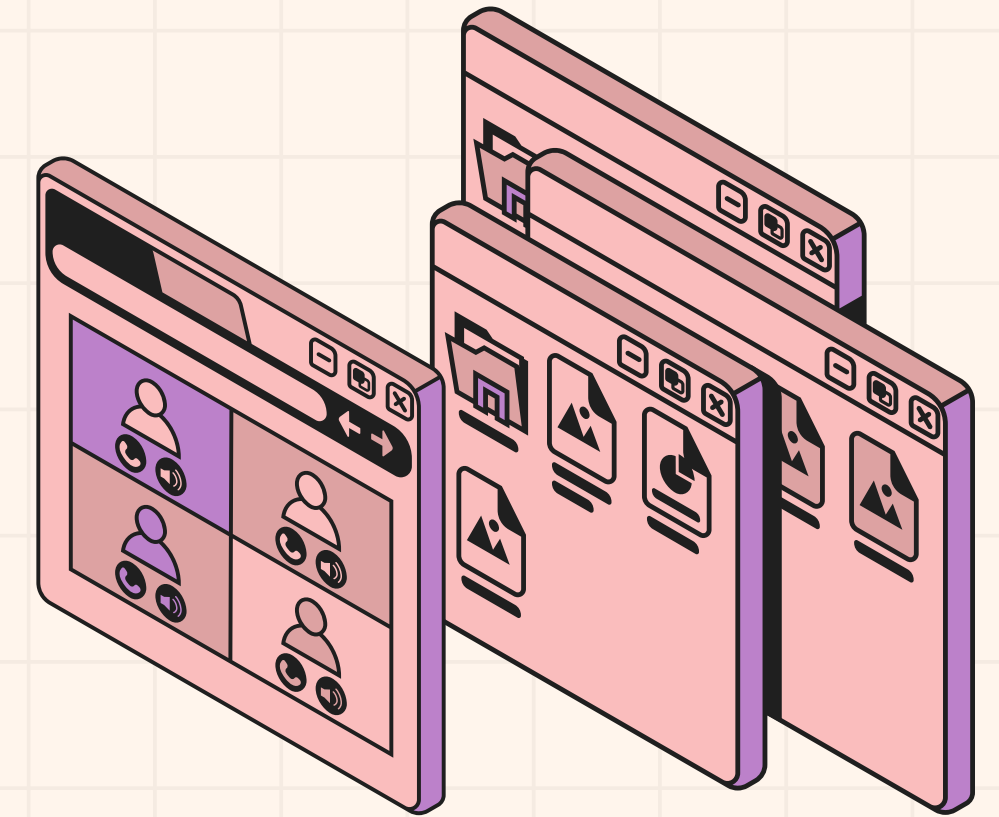


ANIMATION EXPLICITE

Les animations explicites sont plus gérées par le développeur. Une fois définie avec **withAnimation** ou **withTransaction**, elles peuvent être trigger avec une variable ou un event dans le code pour lancer l'animation. C'est le meilleur moyen pour avoir plus de contrôle sur les animations et pousser la customisation. (pour faire certaines choses on est obligé de faire des animations explicites)

PLEINS DE TYPE D'ANIMATIONS

SwiftUI possède une grande variété d'animations prédéfinies pour personnaliser le comportement des animations. Par exemple on trouve des animations linéaires, `easeIn`, `easeOut`, et même des animations de type `spring`. Ces options permettent de créer des effets visuels vraiment sympas.

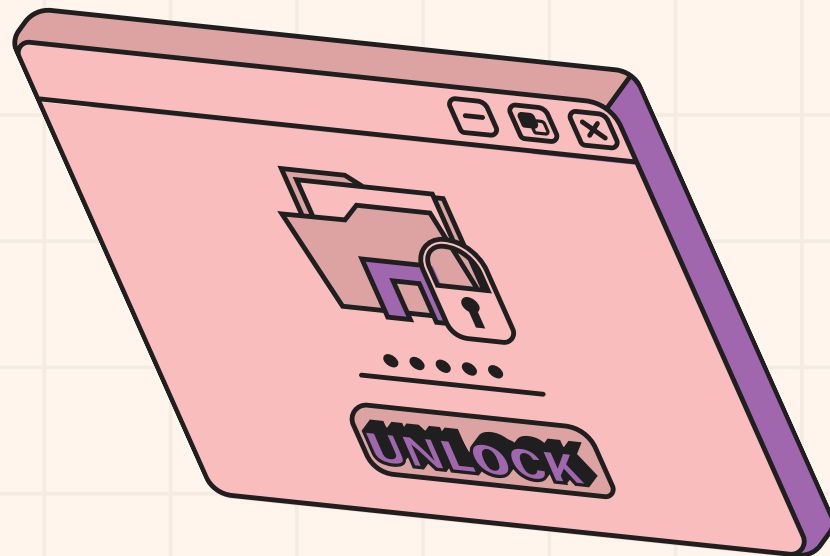


```
withAnimation(.spring(response: 0.6).delay(0.4)) {  
    isAppeared = true  
}
```

```
.animation(.easeOut(duration: 1).delay(0.7),  
    value: isAppeared)
```

CHANGEMENT DE PAGE

SwiftUI permet de créer des transitions fluides entre les pages ou avec des modals. On peut utiliser des opérateurs comme `.opacity`, `.scale`, et `.slide` pour des effets prédéfinis, mais SwiftUI offre également la possibilité de créer des transitions personnalisées pour des animations uniques.



```
struct DetailView: View {
    var body: some View {
        VStack {
            Text("Page Détail")
                .font(.largeTitle)
                .bold()
        }
        .transition(.opacity) // Transition en fondu
        .animation(.easeInOut, value: UUID()) // Animation fluide
    }
}
```

```
var body: some View {
    VStack {
        Button("Ouvrir une page en modale") {
            isModalPresented = true
        }
    }
    .sheet(isPresented: $isModalPresented) {
        ModalView()
            .transition(.move(edge: .bottom)) // Transition personnalisée
            .animation(.spring()) // Animation avec effet ressort
    }
}
```

TRANSITIONS ET ANIMATIONS PERSONNALISÉES

SwiftUI permet de créer des transitions personnalisées pour des scénarios spécifiques, au-delà des types prédéfinis (scale, slide, etc.).

Une transition personnalisée combine :

- Une animation d'apparition (insertion).
- Une animation de retrait (removal).



Exemple de création de transition

```
extension AnyTransition {
    static var rotateAndFade: AnyTransition {
        AnyTransition.modifier(
            active: RotateAndFade(scale: 0.1, opacity: 0),
            identity: RotateAndFade(scale: 1, opacity: 1)
        )
    }
}

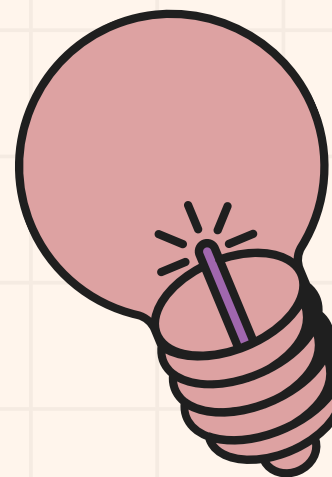
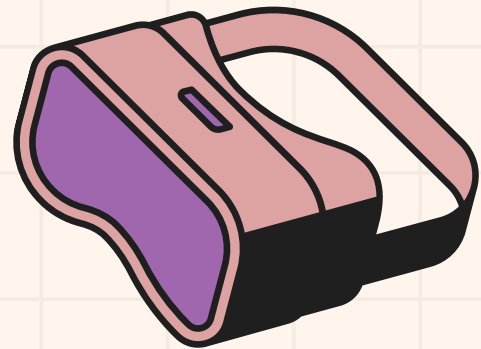
struct RotateAndFade: ViewModifier {
    var scale: CGFloat
    var opacity: Double

    func body(content: Content) -> some View {
        content
            .scaleEffect(scale)
            .opacity(opacity)
            .rotationEffect(.degrees(scale == 1 ? 0 : 90))
    }
}
```

Et ensuite on l'appelle comme un paramètre →

```
var body: some View {
    VStack {
        if showDetail {
            Text("Détails")
                .padding()
                .background(Color.blue)
                .cornerRadius(10)
                .transition(.rotateAndFade)
        }

        Button("Afficher les détails") {
            withAnimation {
                showDetail.toggle()
            }
        }
    }
}
```



MERCI

