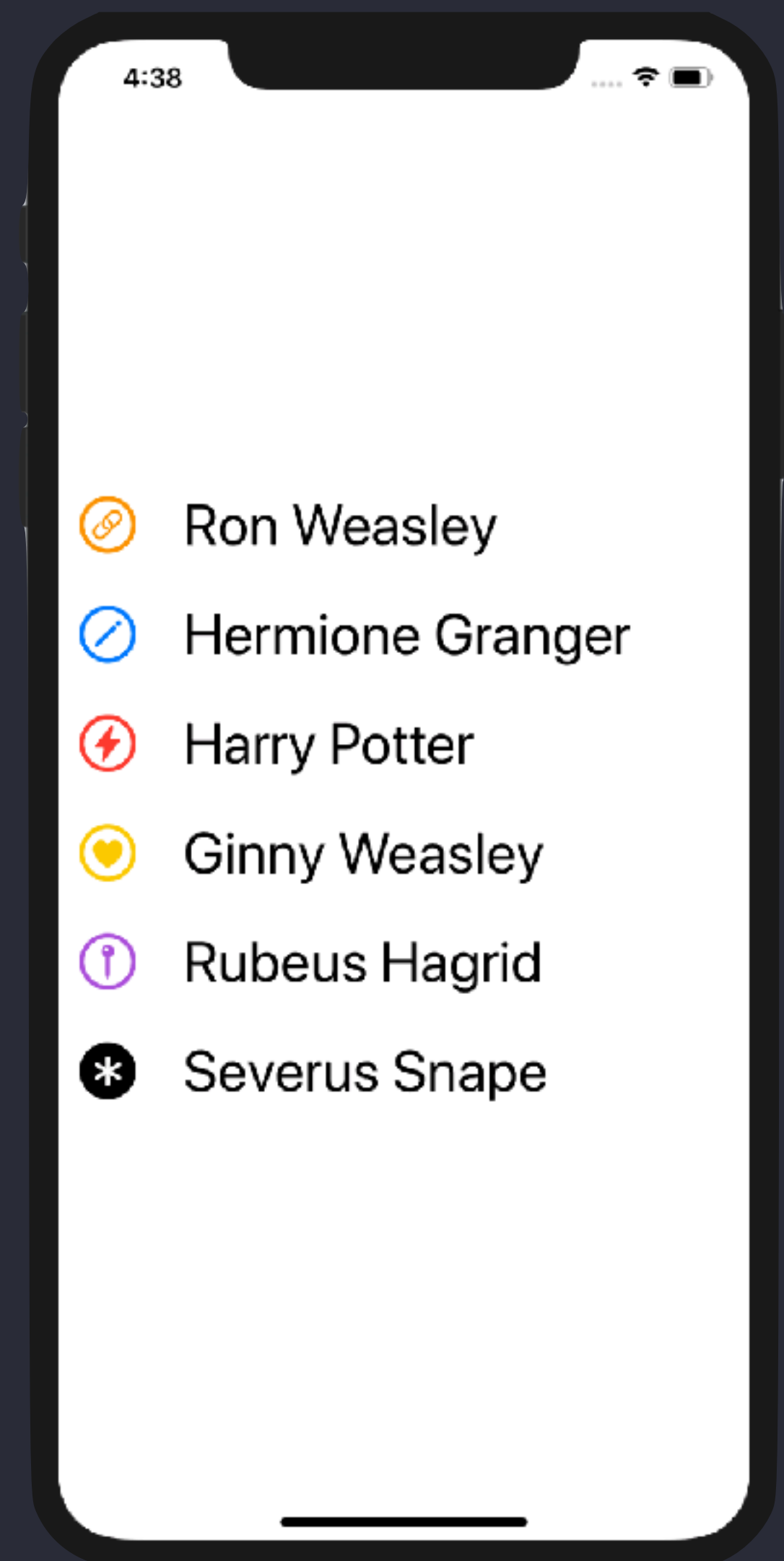


Extraction de vues

```

struct DumbledoreArmyView: View {
    var body: some View {
        VStack (alignment: .leading) {
            HStack {
                Image(systemName: "link.circle")
                    .foregroundColor( .orange )
                    .frame(width: 60)
                Text(" Ron Weasley")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "pencil.circle")
                    .foregroundColor( .blue )
                    .frame(width: 60)
                Text(" Hermione Granger")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "bolt.circle")
                    .foregroundColor( .red )
                    .frame(width: 60)
                Text(" Harry Potter")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "heart.circle")
                    .foregroundColor( .yellow )
                    .frame(width: 60)
                Text(" Ginny Weasley")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "mappin.circle")
                    .foregroundColor( .purple )
                    .frame(width: 60)
                Text(" Rubeus Hagrid")
            }.font(.largeTitle)
        }
    }
}

```

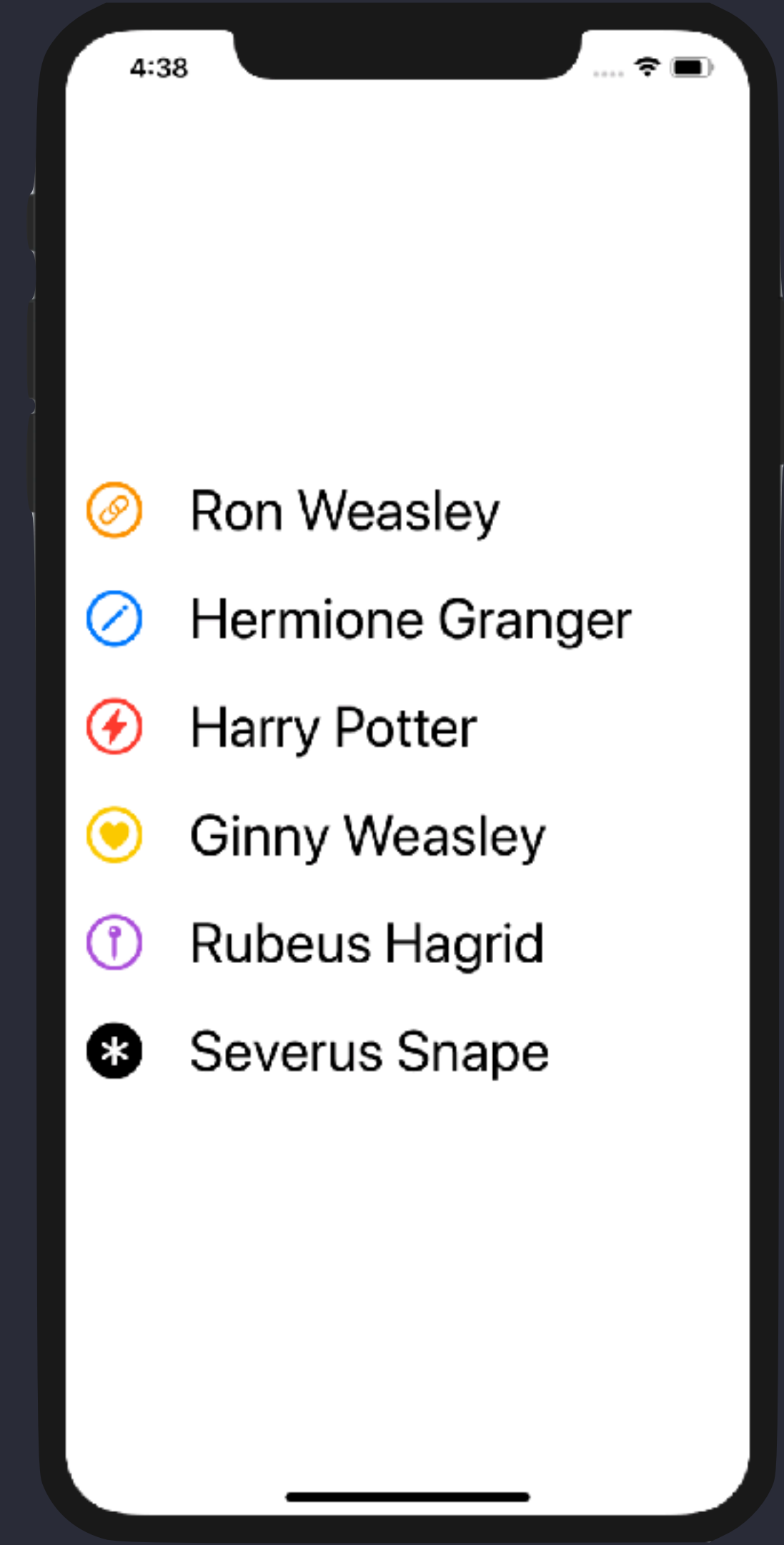


```

struct DumbledoreArmyView: View {
    var body: some View {
        VStack (alignment: .leading) {
            HeroView(icon: "link.circle", name: " Ron Weasley", color: .orange)
            HeroView(icon: "pencil.circle", name: " Hermione Granger", color: .blue)
            HeroView(icon: "bolt.circle", name: " Harry Potter", color: .red)
            HeroView(icon: "heart.circle", name: " Ginny Weasley", color: .yellow)
            HeroView(icon: "mappin.circle", name: " Rubeus Hagrid", color: .purple)
            HeroView(icon: "asterisk.circle.fill", name: " Severus Snape", color: .black)
        }
    }
}

struct HeroView: View {
    var body: some View {
        HStack {
            Image(systemName: icon)
                .foregroundColor(color)
                .frame(width: 60)
            Text(name)
        }.font(.largeTitle)
    }
}

```



```
struct DumbledoreArmyView: View {
    var body: some View {
        VStack (alignment: .leading){
            HeroView(icon: "link.circle", name: " Ron Weasley", color: .orange)
            HeroView(icon: "pencil.circle", name: " Hermione Granger", color: .blue)
            HeroView(icon: "bolt.circle", name: " Harry Potter", color: .red)
            HeroView(icon: "heart.circle", name: " Ginny Weasley", color: .yellow)
            HeroView(icon: "mappin.circle", name: " Rubeus Hagrid", color: .purple)
            HeroView(icon: "asterisk.circle.fill", name: " Severus Snape", color: .black)
        }
    }
}

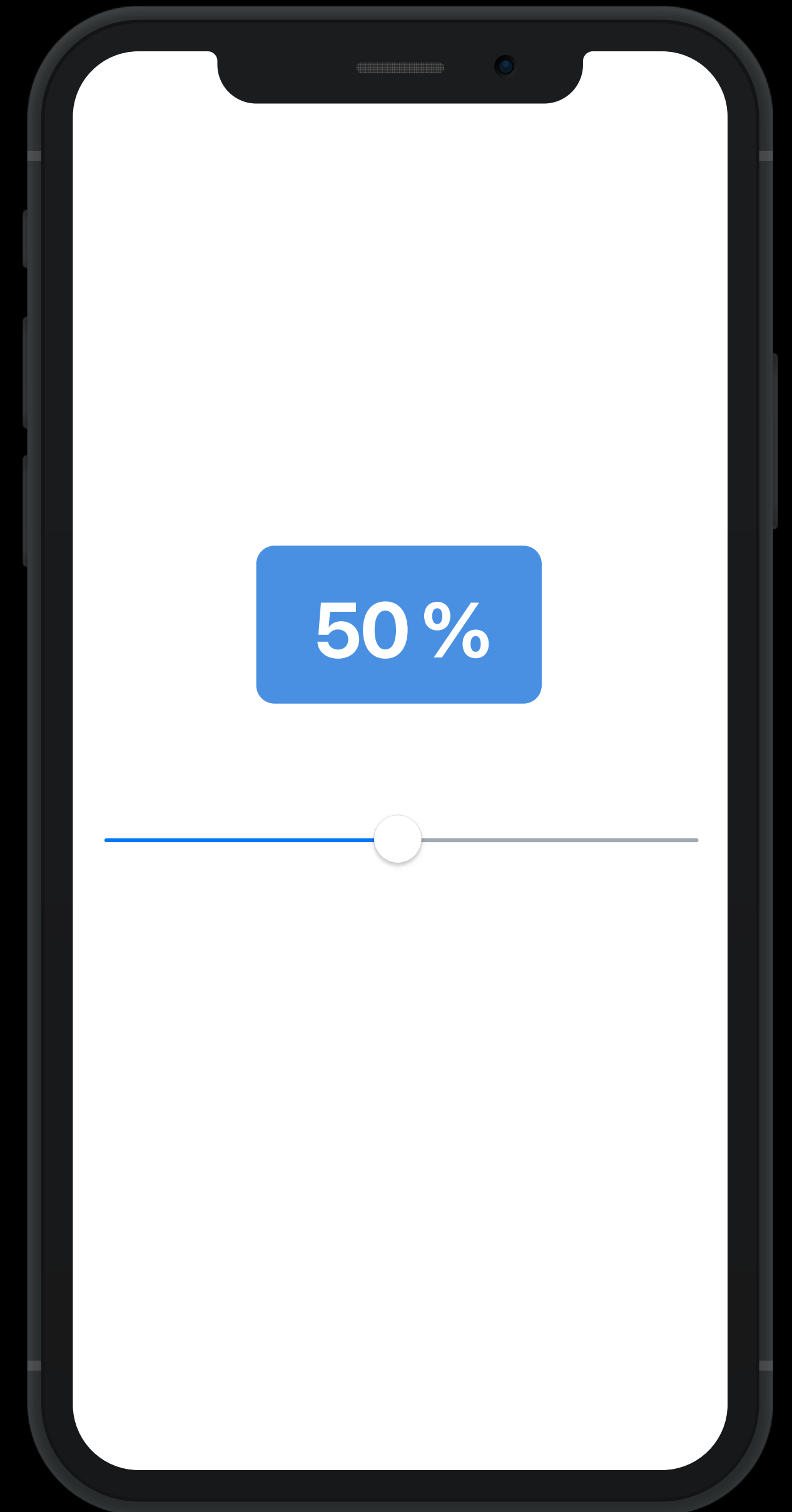
struct HeroView: View {
    let icon: String
    let name: String
    let color: Color
    var body: some View {
        HStack {
            Image(systemName: icon)
                .foregroundColor(color)
                .frame(width: 60)
            Text(name)
        }.font(.largeTitle)
    }
}
```

Pourquoi ? 🤔

- Éviter de dupliquer du code
- Plus facile à lire
- Plus facile à maintenir
 - ↳ Moins de risque de bugs
- Réutilisabilité
 - Un composant complexe personnalisable avec quelques paramètres !

Et les modifieurs ?

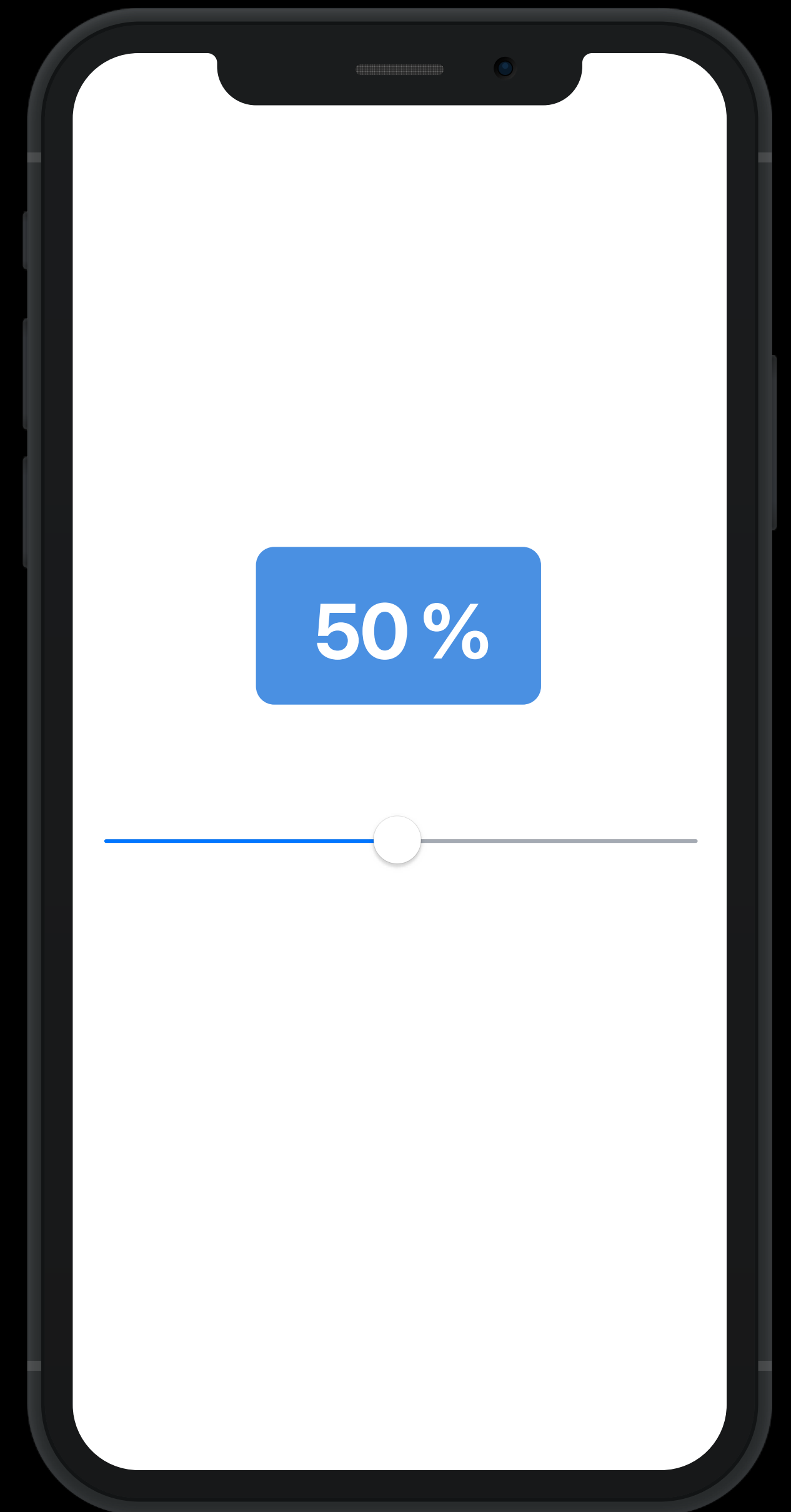
```
struct ContentView: View {  
    @State private var pourcentage: Double = 0.5  
    var body: some View {  
        VStack {  
            Text("\(Int(pourcentage*100)) % »")  
                .font(.largeTitle)  
                .foregroundColor(.white)  
                .padding()  
                .background(color)  
                .clipShape(RoundedRectangle(cornerRadius: 10))  
            Slider(value: $pourcentage, in: 0...1)  
        }  
    }  
}
```



Et les modifieurs ?

```
struct ContentView: View {
    @State private var pourcentage: Double = 0.5
    var body: some View {
        VStack {
            Text("\(Int(pourcentage*100)) % »")
                .framedStyle()
            Slider(value: $pourcentage, in: 0...1)
        }
    }
}

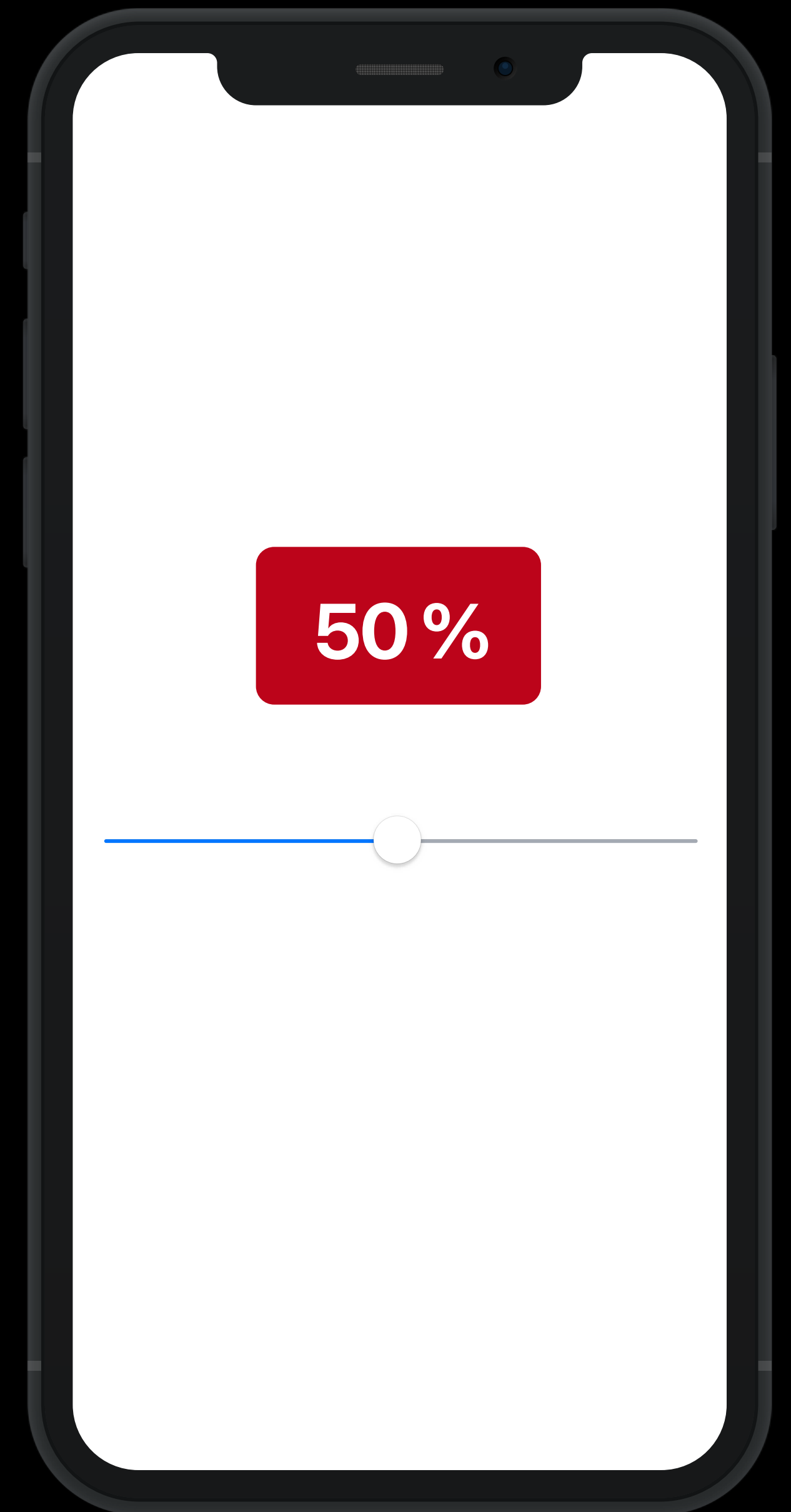
extension View {
    func framedStyle(color: Color = .blue) -> some View {
        self
            .font(.largeTitle)
            .foregroundColor(.white)
            .padding()
            .background(color)
            .clipShape(RoundedRectangle(cornerRadius: 10))
    }
}
```

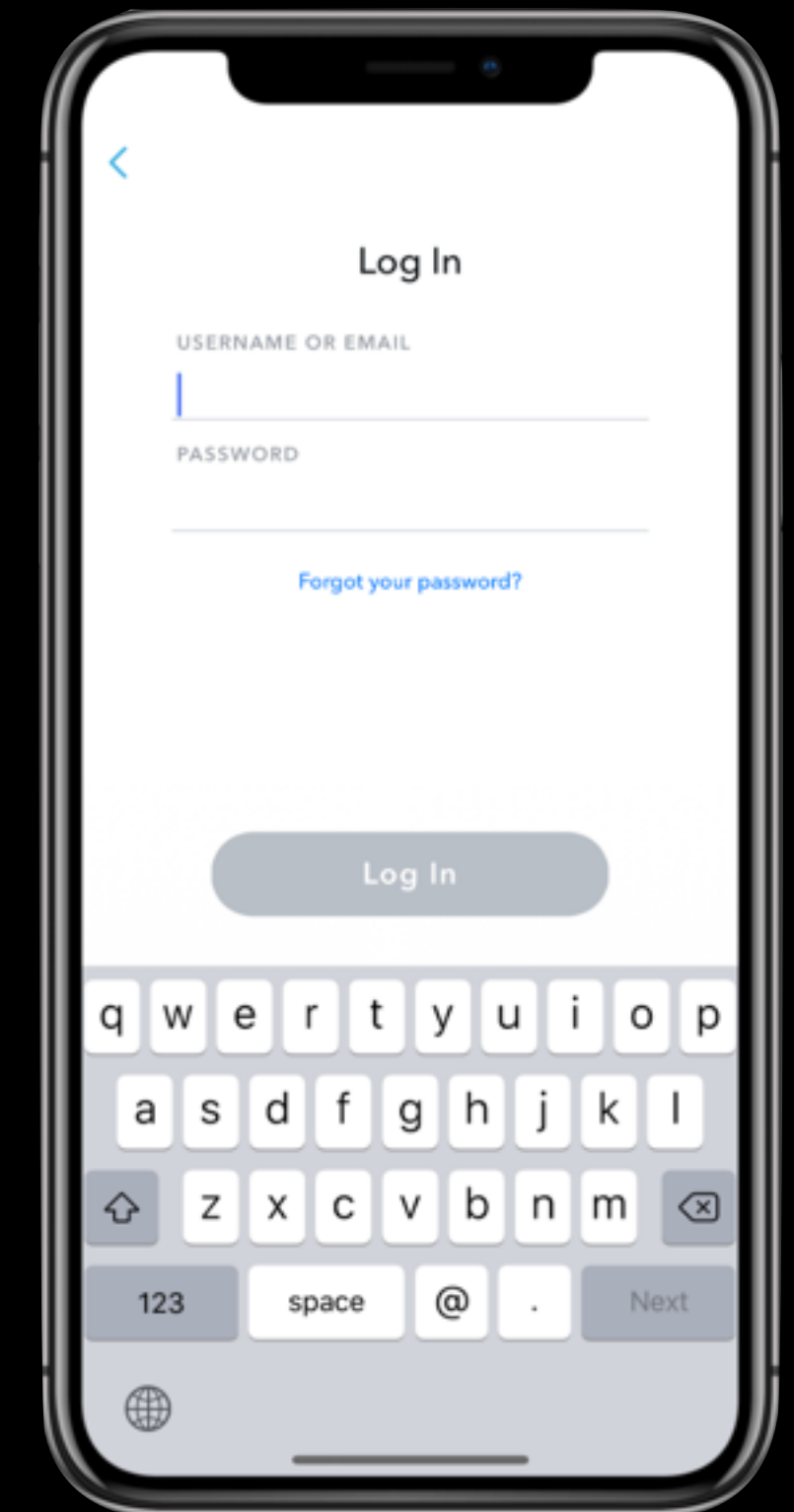
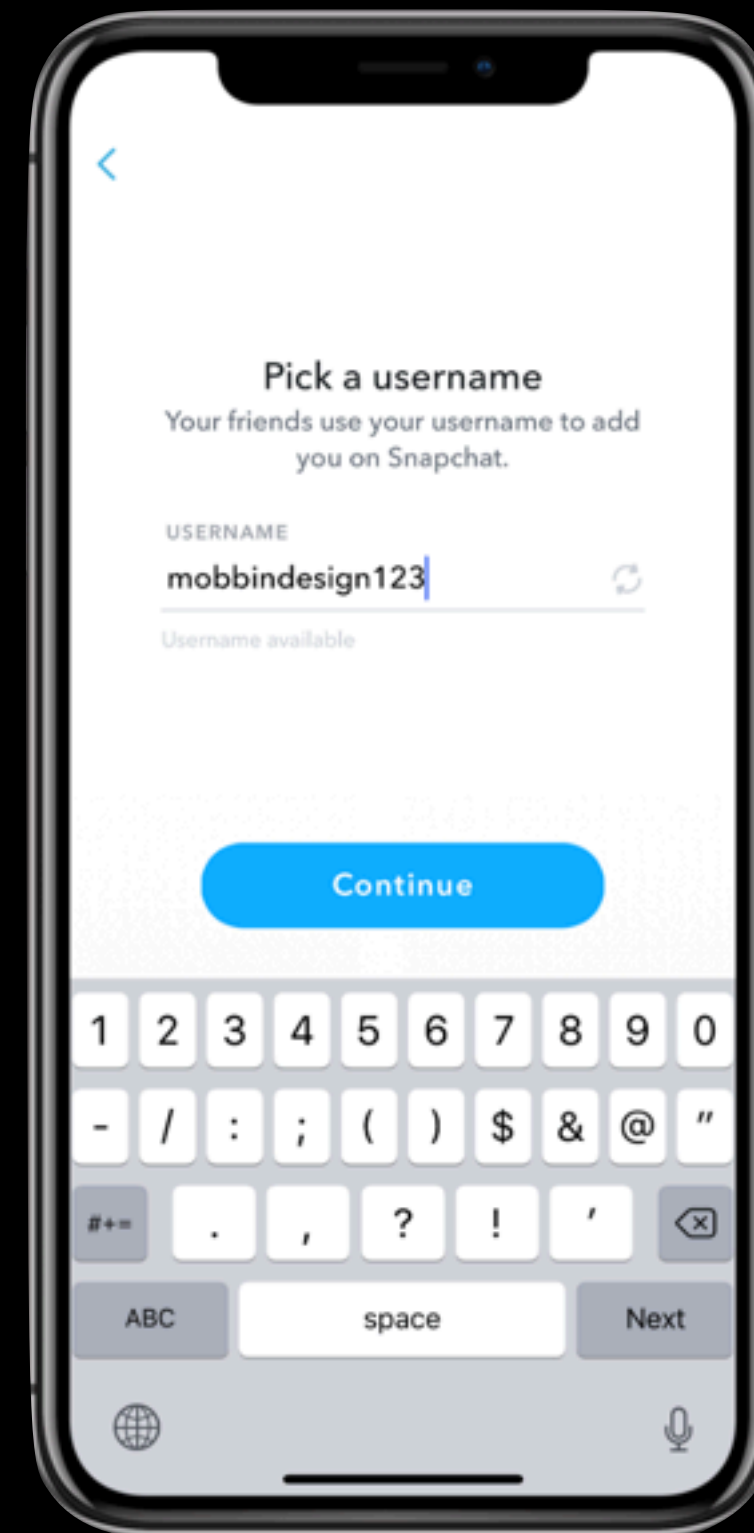
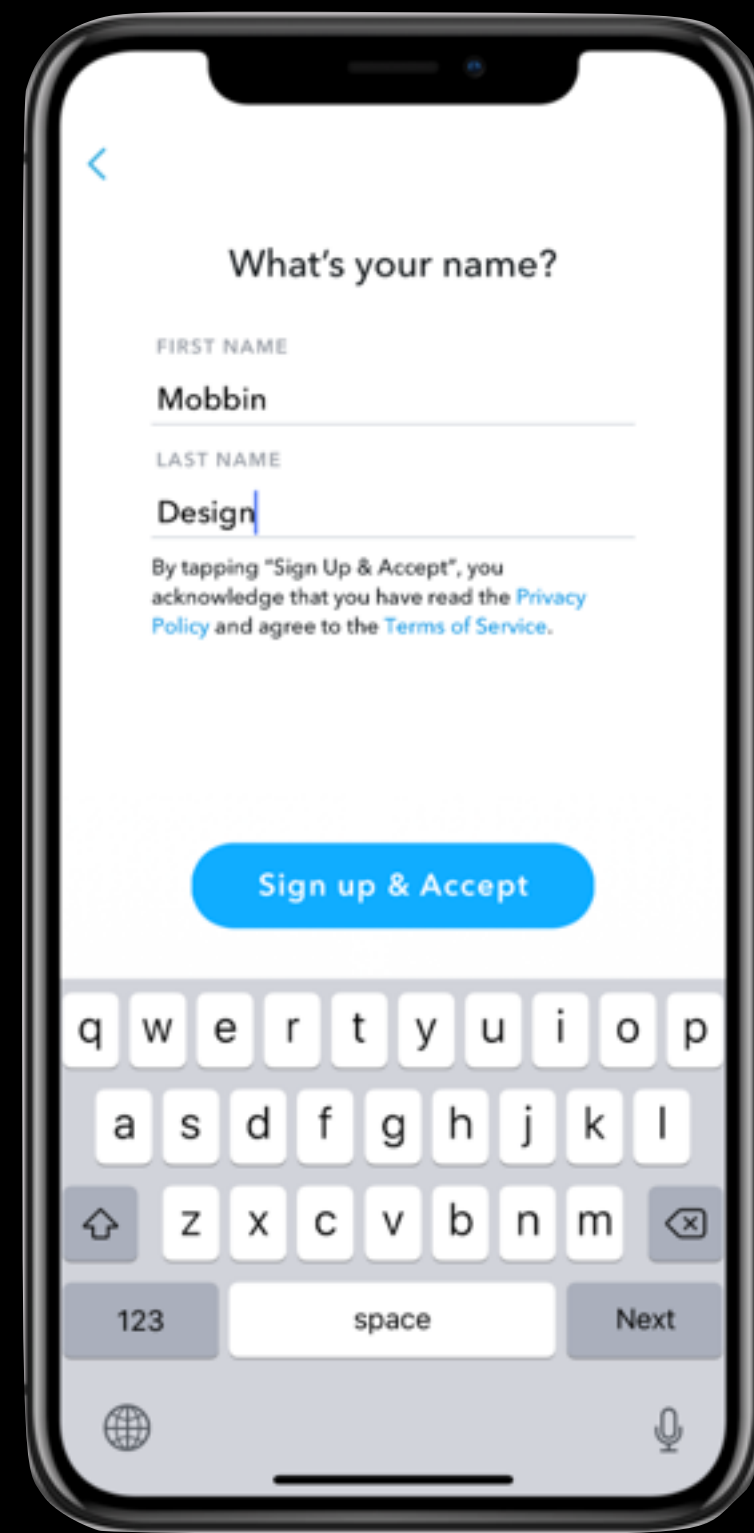


Et les modifieurs ?

```
struct ContentView: View {
    @State private var pourcentage: Double = 0.5
    var body: some View {
        VStack {
            Text("\((Int(pourcentage*100)) % >)\n"
                .framedStyle(color: (pourcentage <
                    0.5) ? .blue : .red)
            Slider(value: $pourcentage, in: 0...1)
        }
    }
}

extension View {
    func framedStyle(color: Color = .blue) -> some View {
        self
            .font(.largeTitle)
            .foregroundColor(.white)
            .padding()
            .background(color)
            .clipShape(RoundedRectangle(cornerRadius: 10))
    }
}
```



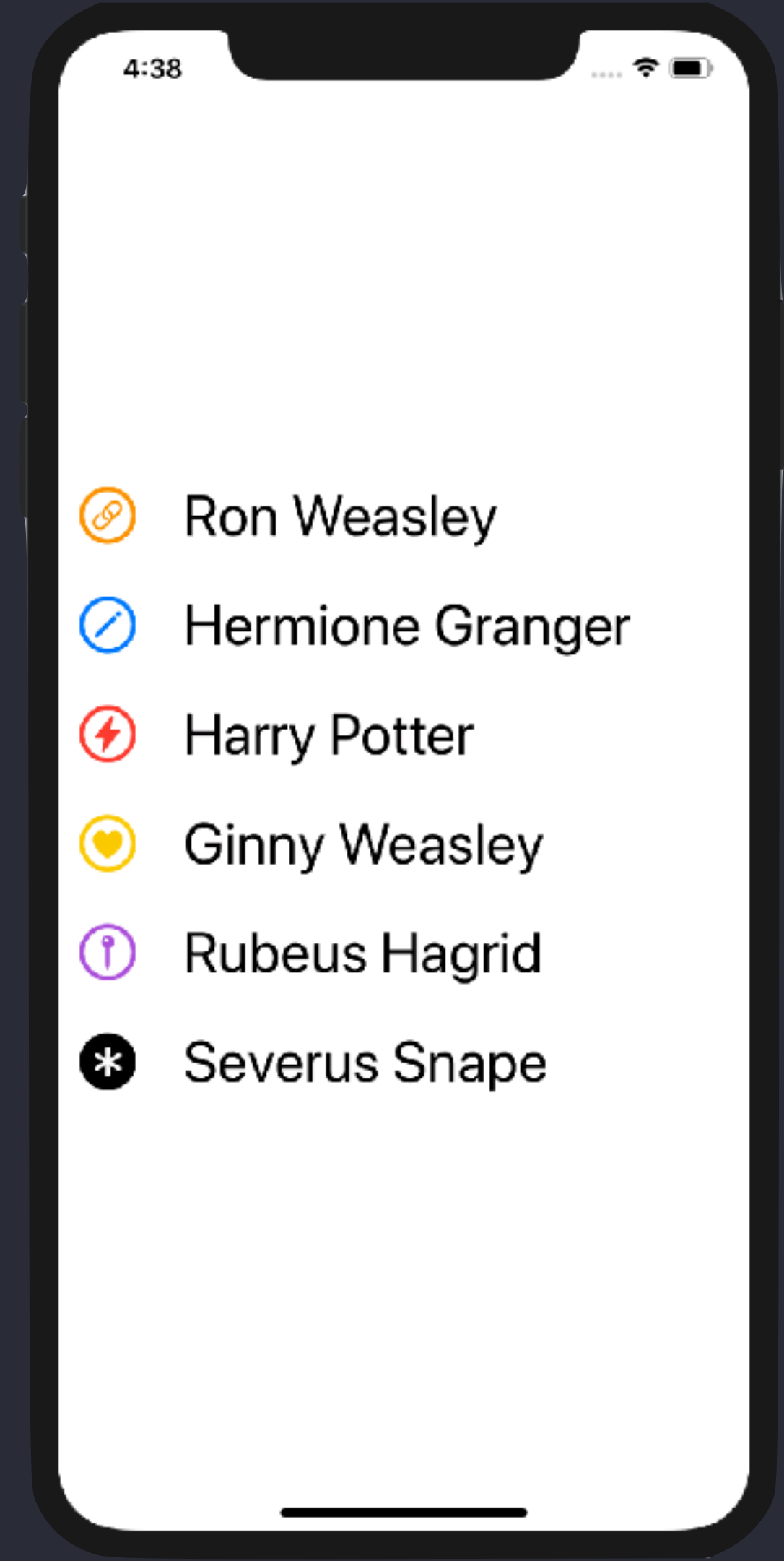


Même exercice...
mais cette fois avec un @State

```

struct DumbledoreArmyView: View {
    var body: some View {
        VStack (alignment: .leading) {
            HStack {
                Image(systemName: "link.circle")
                    .foregroundColor( .orange )
                    .frame(width: 60)
                Text(" Ron Weasley")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "pencil.circle")
                    .foregroundColor( .blue )
                    .frame(width: 60)
                Text(" Hermione Granger")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "bolt.circle")
                    .foregroundColor( .red )
                    .frame(width: 60)
                Text(" Harry Potter")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "heart.circle")
                    .foregroundColor( .yellow )
                    .frame(width: 60)
                Text(" Ginny Weasley")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "mappin.circle")
                    .foregroundColor( .purple )
                    .frame(width: 60)
                Text(" Rubeus Hagrid")
            }.font(.largeTitle)
        }
    }
}

```



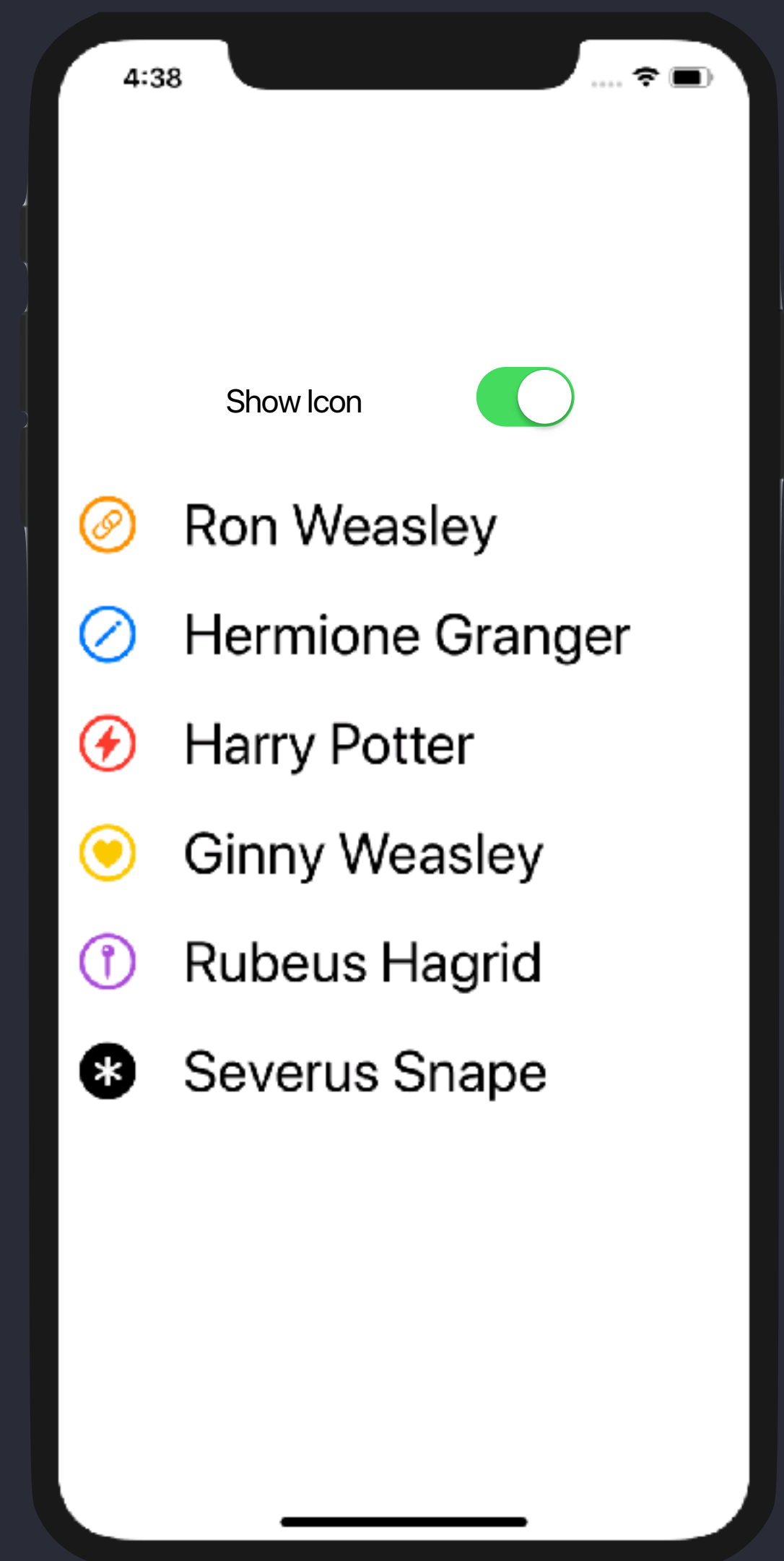
```

struct DumbledoreArmyView: View {

    @State private var showIcon: Bool = true

    var body: some View {
        VStack (alignment: .leading) {
            Toggle("Show Icon", isOn: $showIcon).padding(50)
            HStack {
                Image(systemName: "link.circle")
                    .foregroundColor( .orange )
                    .frame(width: 60)
                Text(" Ron Weasley")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "pencil.circle")
                    .foregroundColor( .blue )
                    .frame(width: 60)
                Text(" Hermione Granger")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "bolt.circle")
                    .foregroundColor( .red )
                    .frame(width: 60)
                Text(" Harry Potter")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "heart.circle")
                    .foregroundColor( .yellow )
                    .frame(width: 60)
                Text(" Ginny Weasley")
            }.font(.largeTitle)
            HStack {
                Image(systemName: "mappin.circle")

```



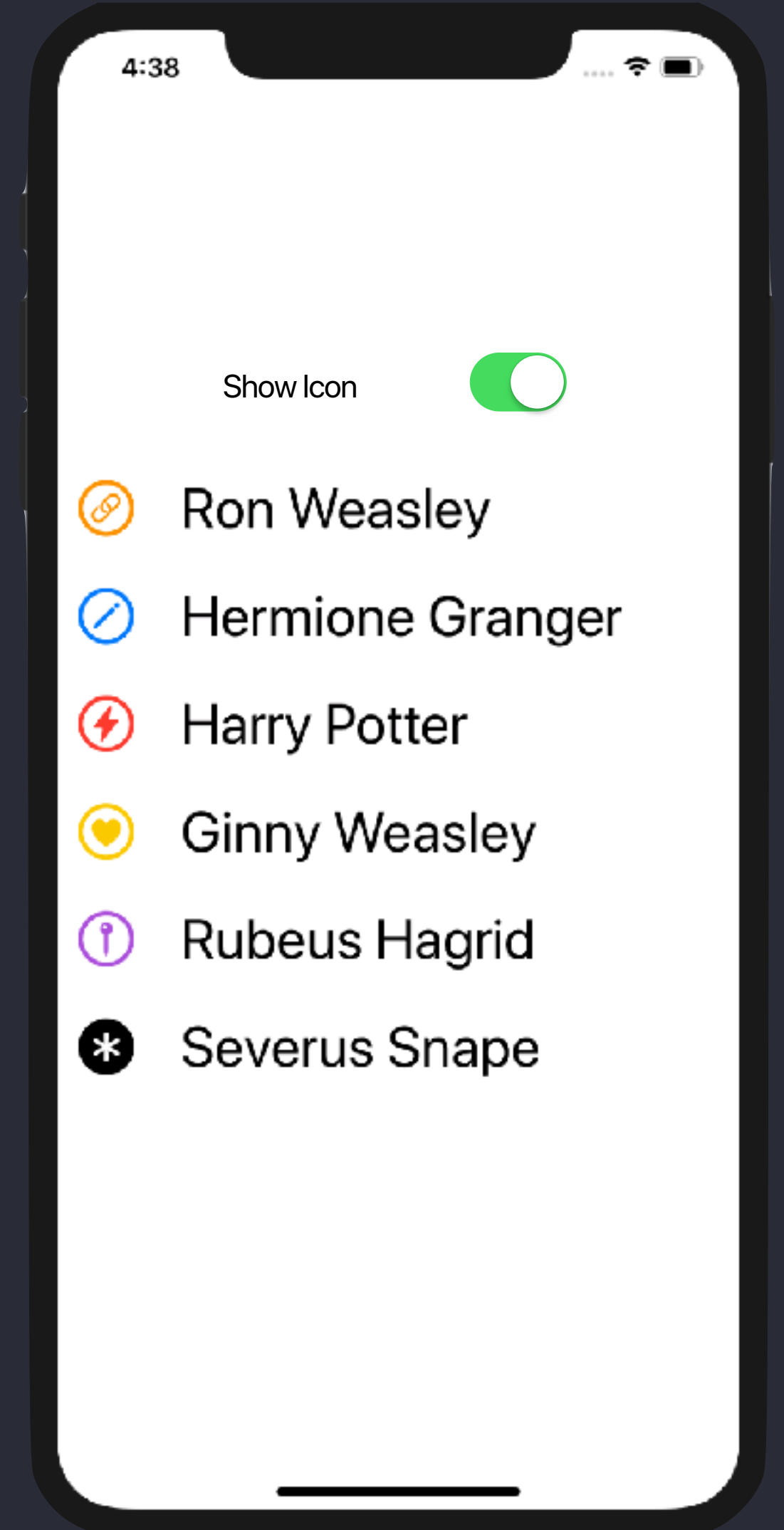
```

struct DumbledoreArmyView: View {

    @State private var showIcon: Bool = true

    var body: some View {
        VStack (alignment: .leading) {
            Toggle("Show Icon", isOn: $showIcon).padding(50)
            HStack {
                if showIcon {
                    Image(systemName: "link.circle")
                        .foregroundColor( .orange )
                        .frame(width: 60)
                }
                Text(" Ron Weasley")
            }.font(.largeTitle)
            HStack {
                if showIcon {
                    Image(systemName: "pencil.circle")
                        .foregroundColor( .blue )
                        .frame(width: 60)
                }
                Text(" Hermione Granger")
            }.font(.largeTitle)
            HStack {
                if showIcon {
                    Image(systemName: "bolt.circle")
                        .foregroundColor( .red )
                        .frame(width: 60)
                }
                Text(" Harry Potter")
            }.font(.largeTitle)
            HStack {

```



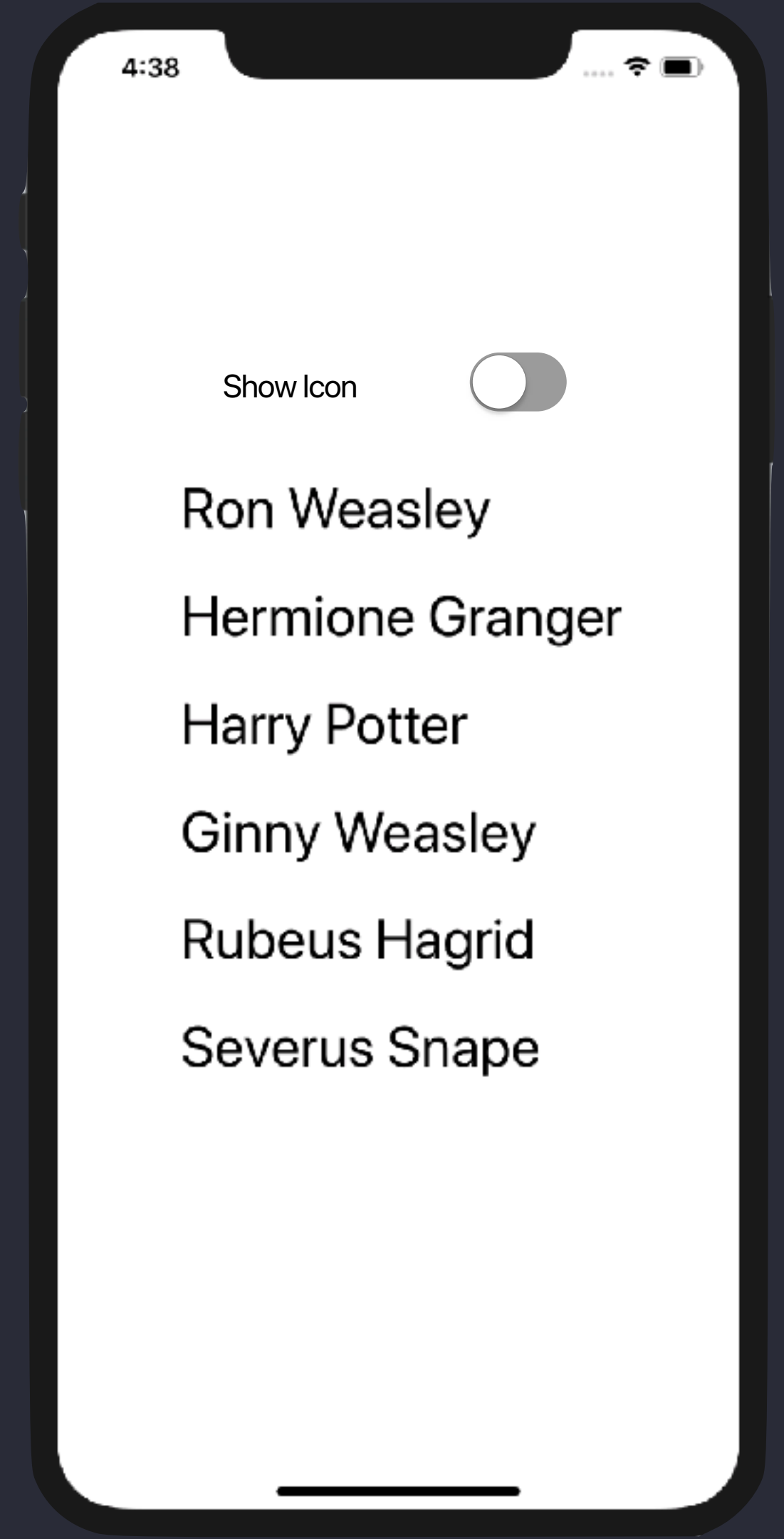
```

struct DumbledoreArmyView: View {

    @State private var showIcon: Bool = false

    var body: some View {
        VStack (alignment: .leading) {
            Toggle("Show Icon", isOn: $showIcon).padding(50)
            HStack {
                if showIcon {
                    Image(systemName: "link.circle")
                        .foregroundColor( .orange )
                        .frame(width: 60)
                }
                Text(" Ron Weasley")
            }.font(.largeTitle)
            HStack {
                if showIcon {
                    Image(systemName: "pencil.circle")
                        .foregroundColor( .blue )
                        .frame(width: 60)
                }
                Text(" Hermione Granger")
            }.font(.largeTitle)
            HStack {
                if showIcon {
                    Image(systemName: "bolt.circle")
                        .foregroundColor( .red )
                        .frame(width: 60)
                }
                Text(" Harry Potter")
            }.font(.largeTitle)
            HStack {

```



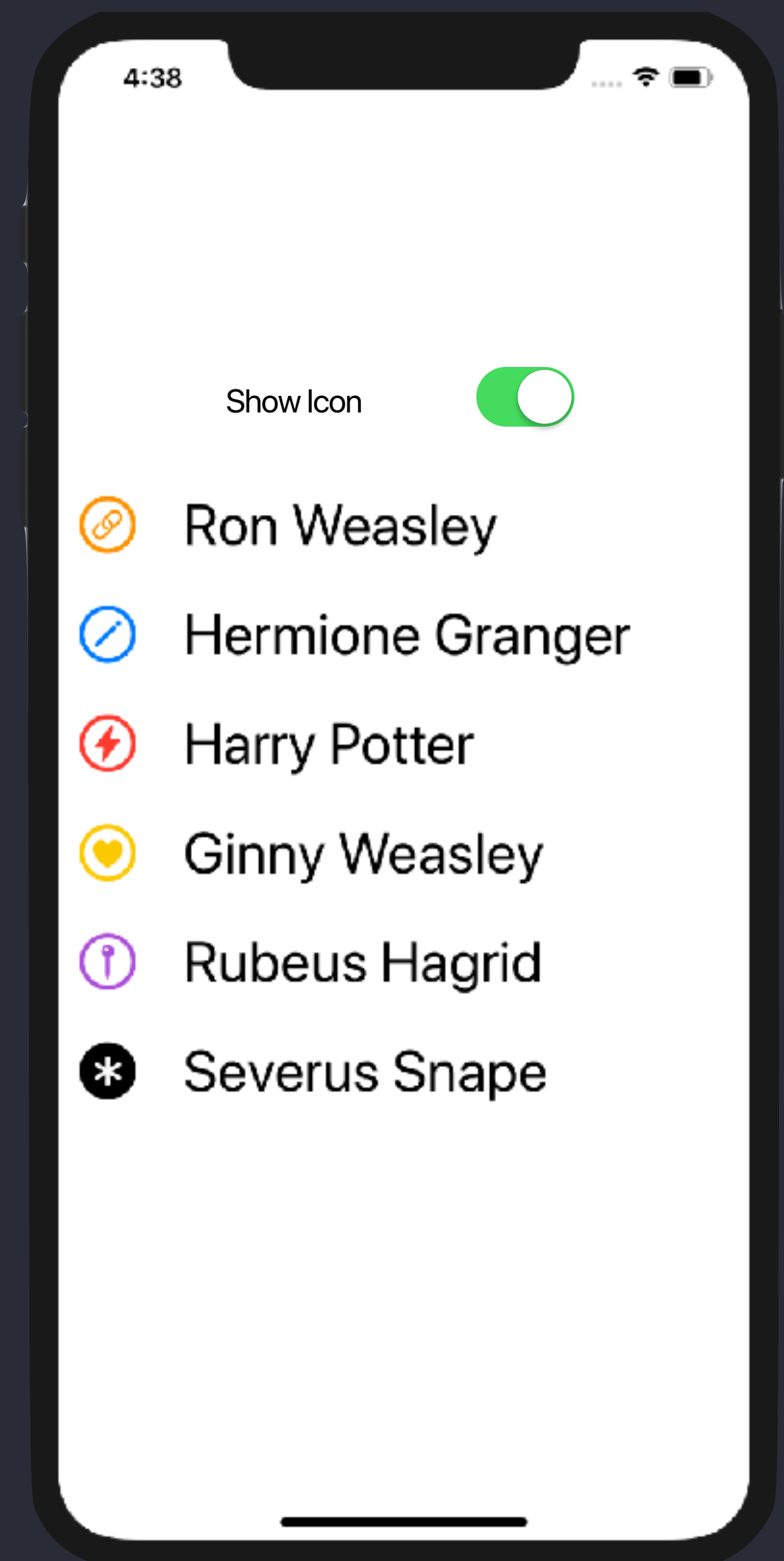

```

struct DumbledoreArmyView: View {

    @State private var showIcon: Bool = true

    var body: some View {
        VStack (alignment: .leading) {
            Toggle("Show Icon", isOn: $showIcon).padding(50)
            HStack {
                if showIcon {
                    Image(systemName: "link.circle")
                        .foregroundColor( .orange )
                        .frame(width: 60)
                }
                Text(" Ron Weasley")
            }.font(.largeTitle)
            HStack {
                if showIcon {
                    Image(systemName: "pencil.circle")
                        .foregroundColor( .blue )
                        .frame(width: 60)
                }
                Text(" Hermione Granger")
            }.font(.largeTitle)
            HStack {
                if showIcon {
                    Image(systemName: "bolt.circle")
                        .foregroundColor( .red )
                        .frame(width: 60)
                }
                Text(" Harry Potter")
            }.font(.largeTitle)
            HStack {

```



```

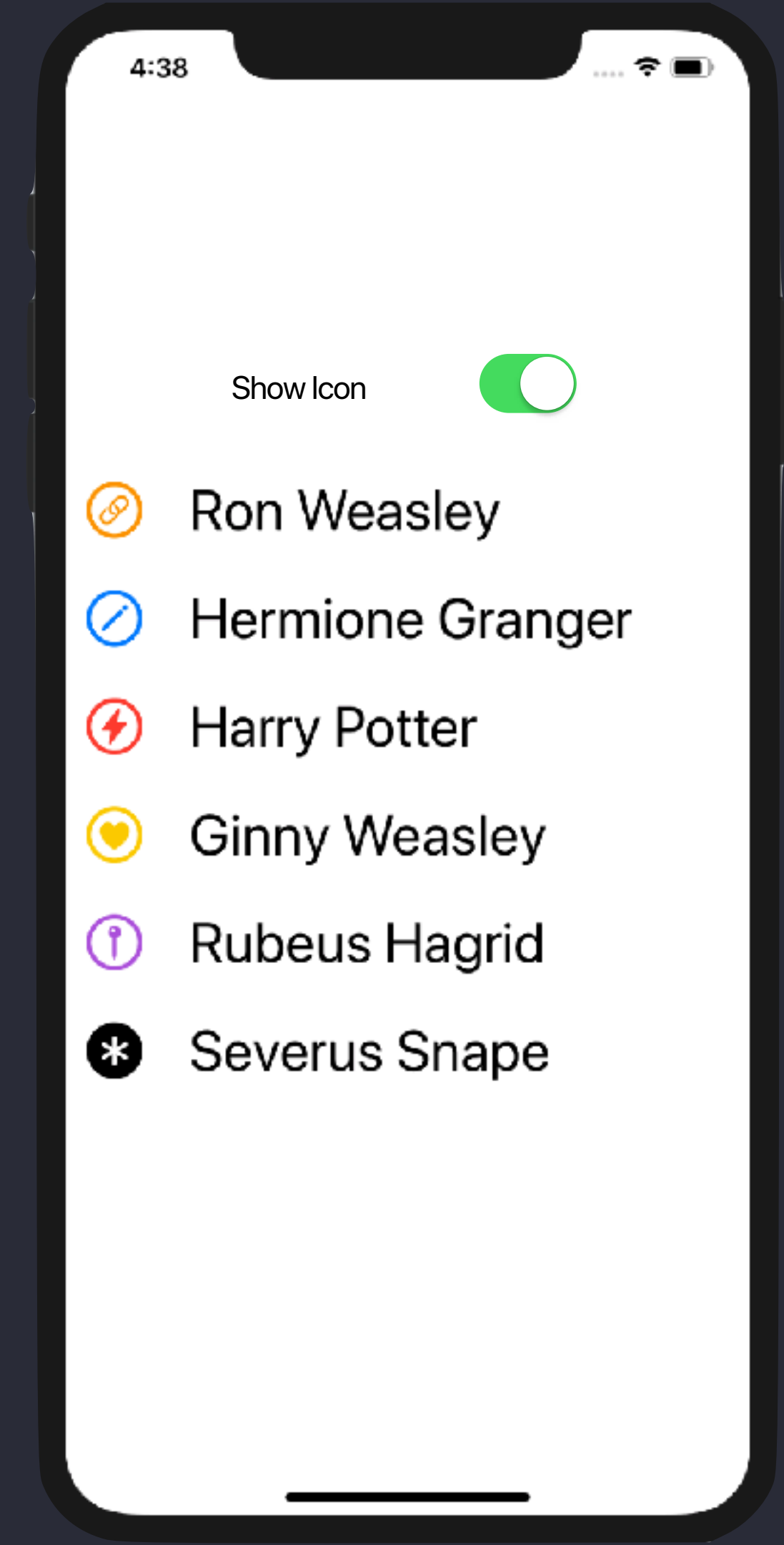
struct DumbledoreArmyView: View {

    @State private var showIcon: Bool = true

    var body: some View {
        VStack (alignment: .leading){
            Toggle("Show Icon", isOn: $showIcon).padding(50)
            HeroView(icon: "link.circle", name: " Ron Weasley", color: .orange)
            HeroView(icon: "pencil.circle", name: " Hermione Granger", color: .blue)
            HeroView(icon: "bolt.circle", name: " Harry Potter", color: .red)
            HeroView(icon: "heart.circle", name: " Ginny Weasley", color: .yellow)
            HeroView(icon: "mappin.circle", name: " Rubeus Hagrid", color: .purple)
            HeroView(icon: "asterisk.circle.fill", name: " Severus Snape", color: .black)
        }
    }
}

struct HeroView: View {
    let icon: String
    let name: String
    let color: Color
    var body: some View {
        HStack {
            if showIcon {
                Image(systemName: "link.circle")
                    .foregroundColor( .orange )
                    .frame(width: 60)
            }
            Text(" Ron Weasley")
        }.font(.largeTitle)
    }
}

```




```

struct DumbledoreArmyView: View {

    @State private var showIcon: Bool = true

    var body: some View {
        VStack (alignment: .leading){
            Toggle("Show Icon", isOn: $showIcon).padding(50)
            HeroView(icon: "link.circle", name: " Ron Weasley", color: .orange, showIcon: $showIcon)
            HeroView(icon: "pencil.circle", name: " Hermione Granger", color: .blue, showIcon: $showIcon)
            HeroView(icon: "bolt.circle", name: " Harry Potter", color: .red, showIcon: $showIcon)
            HeroView(icon: "heart.circle", name: " Ginny Weasley", color: .yellow, showIcon: $showIcon)
            HeroView(icon: "mappin.circle", name: " Rubeus Hagrid", color: .purple, showIcon: $showIcon)
            HeroView(icon: "asterisk.circle.fill", name: " Severus Snape", color: .black, showIcon: $showIcon)
        }
    }
}

```

```

struct HeroView: View {
    let icon: String
    let name: String
    let color: Color
    @Binding var showIcon: Bool
    var body: some View {
        HStack {
            if showIcon {
                Image(systemName: "link.circle")
                    .foregroundColor( .orange )
                    .frame(width: 60)
            }
            Text(" Ron Weasley")
        }.font(.largeTitle)
    }
}

```

Résumé

- Ne pas avoir un gros paquet de code répétitif, mais extraire les composants
 - S'il y a un état qui se retrouve d'un côté et de l'autre, utiliser `@Binding`
- Extraire aussi les modifieurs selon le même principe
- Permet d'avoir un code
 - plus propre
 - plus lisible
 - plus facile à modifier
 - moins de bugs