

Übung 4

- Grundlage dieser Übung ist ein einfaches Datenmodell von einer Community mit Mitglieder
 - Erstellen Sie zuerst ein neues, leeres SwiftUI-Projekt.
- Erstellen Sie zwei **Klassen** **Community** und **Member** um dieses Model abzubilden
 - Jedes Member soll initial 10'000.- Vermögen als Integer speichern
 - Member hat eine Methode **lend**, die einen zufälligen verfügbaren Betrag an ein anderes Member sendet.
 - Die Community enthält ein Array (n=5) von Members

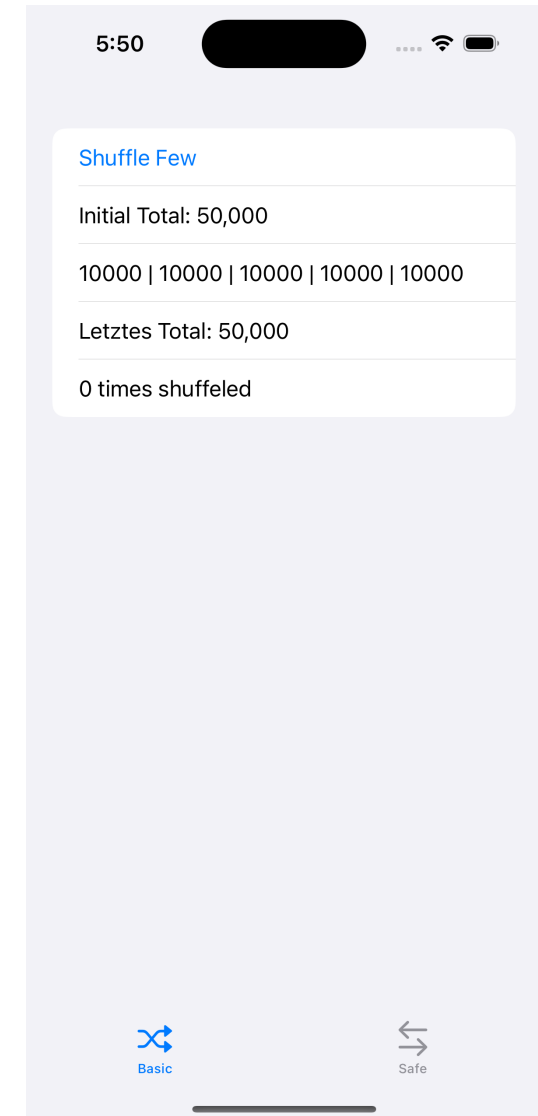


Shuffle Few

- Erstellen Sie in der ContentView einen Button «Shuffle Few», der wiederholt einen Betrag von einem Member zu einem zweiten sendet
 - Verwenden Sie dazu die Random-Methoden von Int und Array
- Finden Sie eine möglichst grosse Zahl an Iterationen, mit denen noch kein Einfrieren des Interfaces festzustellen ist

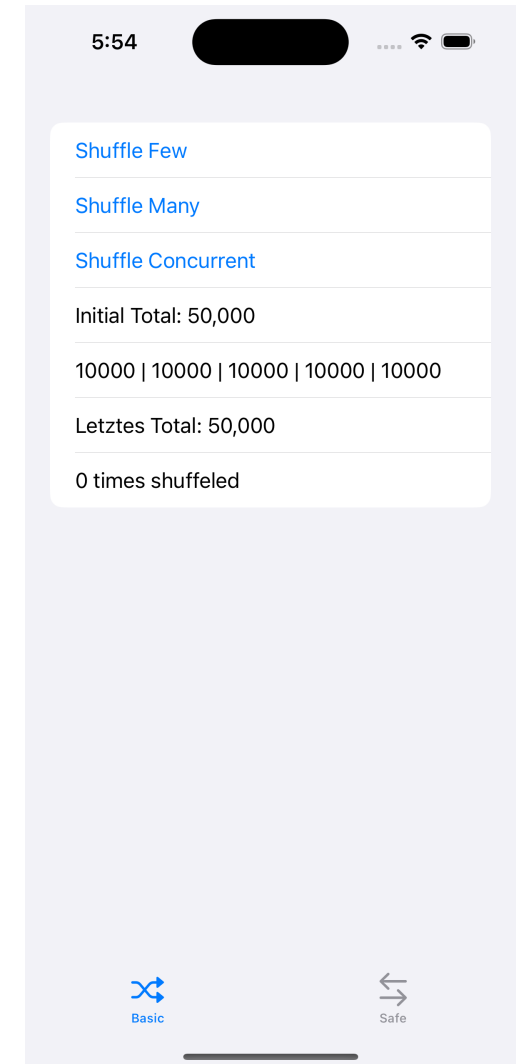


```
Int.random(in: 0...10_000)  
members.randomElement()
```



Shuffle Many

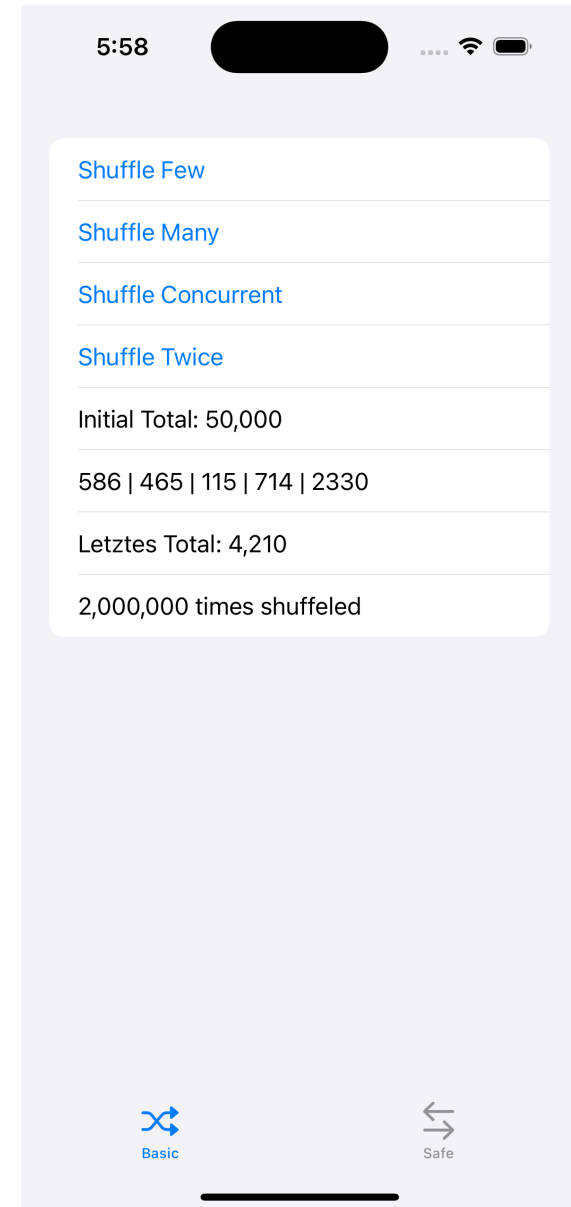
- Erstellen Sie einen weiteren Button «Shuffle Many» und führen damit so viele Iterationen durch, dass das Interface spürbar (für 1-2s) einfriert.
- Erstellen Sie eine weitere Methode, welche die gleiche Anzahl Iterationen in einem `Task { }` asynchron durchführt.



Shuffle Twice

- Erstellen Sie einen weiteren Button «Shuffle Twice», der zwei Tasks mit unterschiedlicher Priorität startet, die unabhängig ein shuffle durchführen.
- Erwartung: Durch eine Race Condition ist das Total danach nicht mehr erhalten.

```
Button("Shuffle Twice") {  
    Task(priority: .high) {  
        await community.shuffleConcurrent()  
    }  
    Task(priority: .low) {  
        await community.shuffleConcurrent()  
    }  
}
```



Actors

- Erstellen Sie eine Kopie von der View und den beiden Models
- Sie können die View manuell ersetzen oder mit zweit Tabs abbilden
- Verändern Sie den Typ von **Community** und **Member** zu **actor**
 - Sie werden einige **async** einfügen müssen
- Reduzieren Sie die Klassen und Views auf die letzten beiden Buttons «Shuffle Concurrent» und «Shuffle Twice»
- Erwartung: Das Total bleibt nun auch bei «Shuffle Twice» erhalten

