Bachelor in de Toegepaste Informatica

	**Labo Apps for iOS** Dirk Hostens Milan Dima
Opdracht 4	http://developer.apple.com

In deze opdracht maken we een app waarbij we opnieuw data uitlezen uit een JSON.

De JSON bevat informatie over de gespeelde wedstrijden op het WK in Qatar ([WKResult]). De JSON wordt toegevoegd aan het project en kan dan ingeladen worden via de load function (zie startbestanden).

Een WKResult bevat volgende velden:

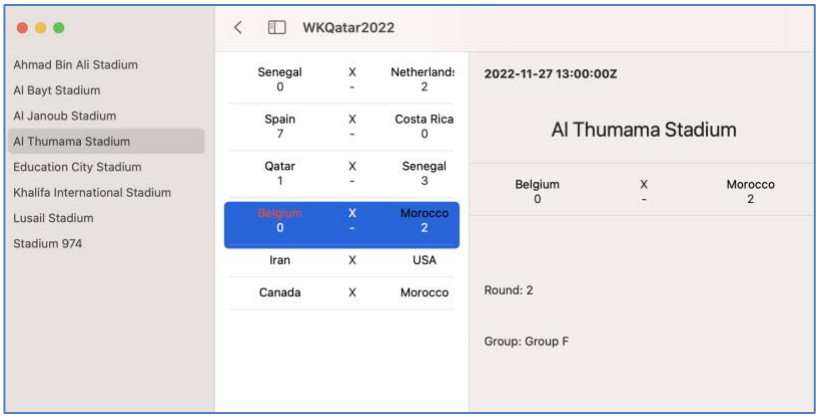
let matchNumber, roundNumber: Int
let dateUTC, location, homeTeam, awayTeam: String
let group: String?
let homeTeamScore, awayTeamScore: Int?

Let op de Optionals, je zal hiermee rekening moeten houden bij de uitwerking van de app.

Bij de start van de app moet je jouw favoriete team kiezen. De lijst van team bekom je door de array van WKResults te bewerken zodat er een lijst van teams overblijft. Denk na waar je die code zal uitwerken. Na keuze van je favoriete team kom er onderaan een knop NEXT beschikbaar en kleurt het geselecteerde team rood in de lijst.
Next brengt je naar het volgende scherm. De navigatie gebeurt in een NavigationStack.

Dit is een NavigationSplitView met 3 onderverdelingen.
Links vind je de lijst van stadions, na selectie van het stadion komen de matchen gesorteerd op datum in de middenste kolom. Selectie van de match toont dan de details in de rechtste kolom.

Voorbeeld van start scherm en welkom scherm.



Merk op dat Belgium in het rood staat bij de details in de middenste kolom als favoriete team geselecteerd in het vorige scherm. Om een score te tonen maken we gebruik van een Grid. Dan kan je de doelpunten gescoord mooi onder de naam van de ploeg krijgen.

```
Grid {
        GridRow {

        }
        GridRow {

        }
}
```

Enkele tips:

- De gegevens die je uit de JSON haalt, heb je op bijna alle schermen nodig. Daarom is het zinvol om een WKDataStore klasse aan te maken waarin je de JSON omzet naar results van het type [WKResult]. WKDataStore maak ik dan beschikbaar als Environment variabele. In deze klasse voeg ik ook alle nodige functies toe zoals bv. getAllTeams() om alle landen uit de array te halen (zie eerste scherm).
- De navigatie op het eerste scherm via de Next knop gebeurt aan de hand van een NavigationLink in een NavigationStack.
- We maken telkens gebruik van de List constructor waarbij je selection als parameter hebt. Op die manier kan je in een State var het geselecteerde item bijhouden. Als id kan je \.self gebruiken.
  List(**data:** RandomAccessCollection, **id**: RandomAccessCollection.Element. Hashable, **selection**: Binding<Hashable>, **rowContent**: (RandomAccessCollection.Element) -> View)
- In de NavigationSplitview worden de stadions getoond. Na selectie van een stadion worden de resultaten van de gespeelde matchen in dat stadion getoond. Selectie van een match toont dan de details in de rechterkolom. Als ik daarna een nieuw stadion selecteer, moeten de gespeeld matchen worden aangepast en moet de geselecteerde match terug leeg worden. Daarom neem ik alle variabelen die een selectie in de navigationsplitview/list bijhouden op in een State var. Dan kan ik ingrijpen en bij het aanpassen van het geselecteerde stadion door het geselecteerde match te resetten (nil value geven). Hierbij voorbeeld waarbij selectedLocation dus het stadion is en selectedWkResult de geselecteerde match .

```
var selectedWkResult: WKResult?
private var _selectedLocation : String?
var selectedLocation: String? {
    get {
      return _selectedLocation
    }
    set {
      selectedWkResult = nil
      _selectedLocation = newValue
    }
  }
```

-

Bachelor in de Toegepaste Informatica

Veel succes!

Documentatie: TCDC, ETBN, ETCNH