

ios 應用程式開發

Week 4

本週內容

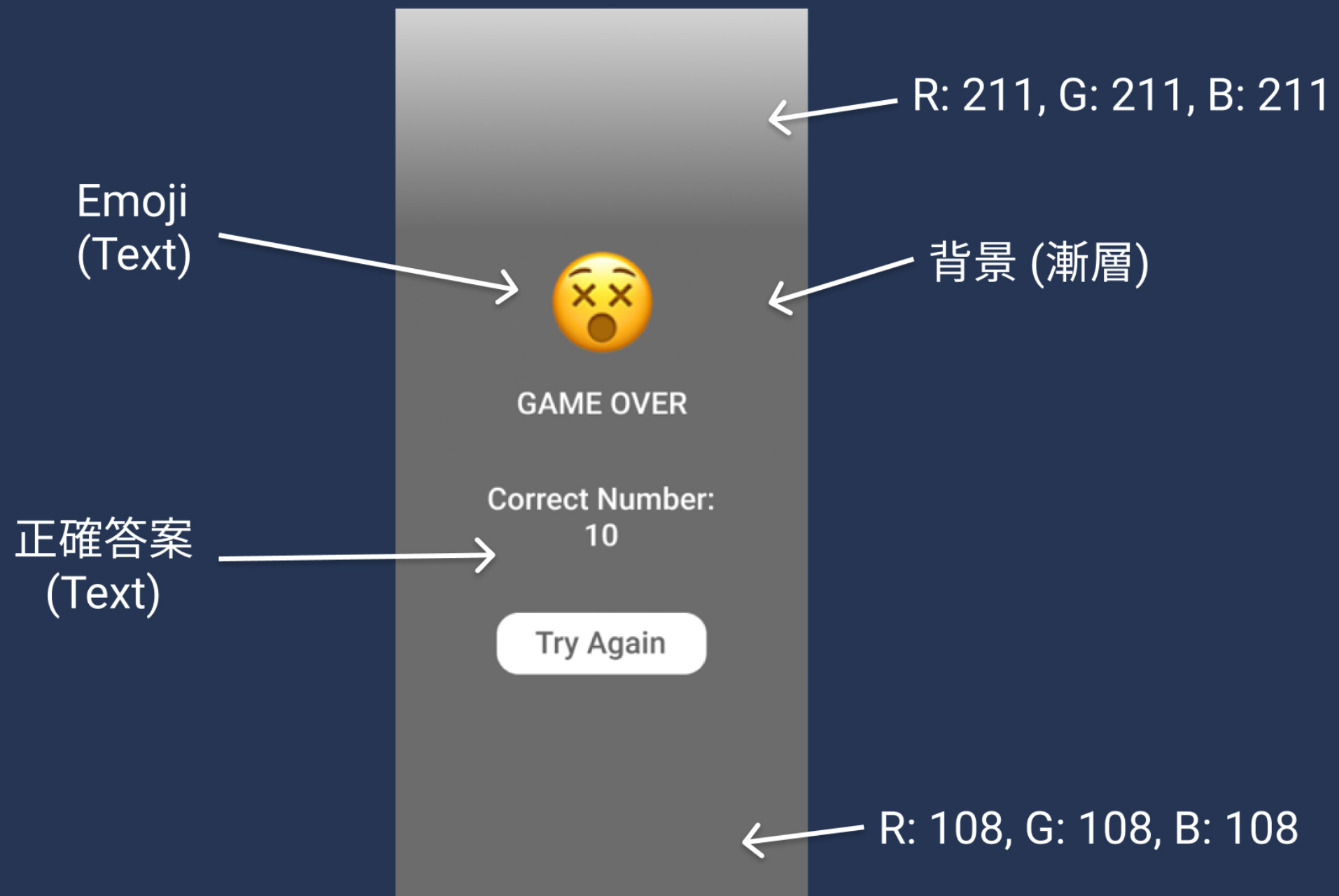
- 編寫一個簡單的遊戲 - 估數字

設計檔案

[https://www.figma.com/file/
RCNIbEk9Or5fsWqzBHNZDM/Game-Screen?
node-id=0%3A1](https://www.figma.com/file/RCNIbEk9Or5fsWqzBHNZDM/Game-Screen?node-id=0%3A1)







第一部分 - 遊戲畫面

Gradient

```
VStack {  
...  
}  
.background(  
    LinearGradient(  
        gradient: Gradient(colors: greenBackground),  
        startPoint: .top,  
        endPoint: .bottom  
    )  
)
```


Gradient

```
VStack {  
...  
}  
.background(  
    LinearGradient(  
        gradient: Gradient(colors: greenBackground),  
        startPoint: .top,  
        endPoint: .bottom  
    )  
)
```

Gradient

```
VStack {  
...  
}  
.background(  
    LinearGradient(  
        gradient: Gradient(colors: greenBackground),  
        startPoint: .top,  
        endPoint: .bottom  
    )  
)
```

Gradient

```
let greenBackground = [  
    color1,  
    color2,  
    color3  
]
```

Gradient

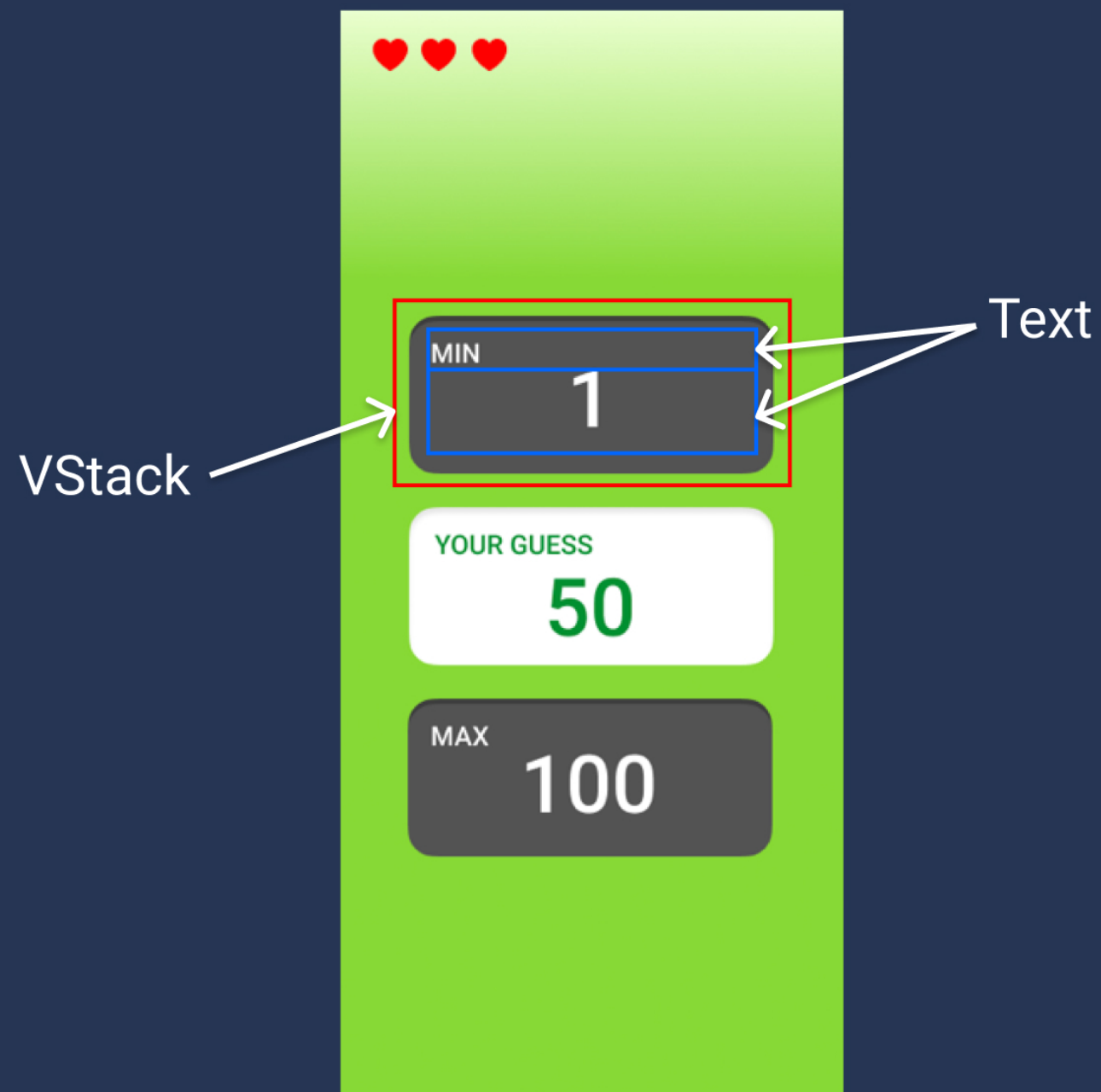
```
let greenBackground = [  
    color1,  
    color2,  
    color3  
]
```

Gradient

```
let greenBackground = [  
    Color(red: 234.0/255.0, green: 255.0/255.0, blue: 207.0/255.0),  
    Color(red: 135.0/255.0, green: 217.0/255.0, blue: 52.0/255.0),  
    Color(red: 135.0/255.0, green: 217.0/255.0, blue: 52.0/255.0)  
]
```

背景佔滿整個版面

```
VStack {  
...  
}  
.background(  
    LinearGradient(  
        gradient: Gradient(colors: greenBackground),  
        startPoint: .top,  
        endPoint: .bottom  
    )  
)  
.frame(maxWidth: .infinity, maxHeight: .infinity)  
.edgesIgnoringSafeArea(.all)
```

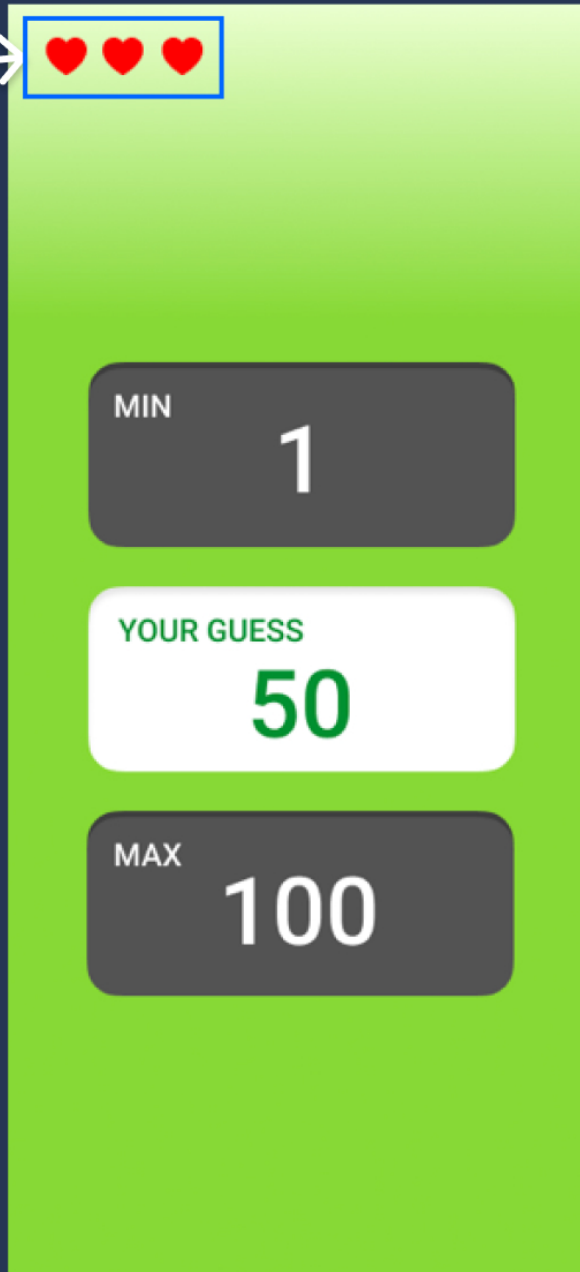


圓角版面

```
VStack() {  
    ...  
}  
    .padding()  
    .frame(maxWidth: 250)  
    .background(darkGray)  
    .cornerRadius(10.0)  
    .foregroundColor(.white)
```




HStack →



剩餘次數

```
let maxTries = 5 // 定義最多重試次數
```

```
~~
```

```
HStack {  
    ForEach(0 ..< maxTries) { index in  
        Image(systemName: "heart.fill")  
            .foregroundColor(.red)  
    }  
}
```

ForEach

```
ForEach(0...10) { index in // 0-10 (共11次), index 為每次執行的數字
    Text("\(index)")
}
// ~~~

ForEach(0..<10) { index in // 0-9 (共10次), index 為每次執行的數字
    Text("\(index)")
}
// ~~~

ForEach(0...10) { // 沒有 index, 使用 $0 代表第一個參數
    print("\( $0)")
}
```

?: 語法 (Ternary Operator)



如果使用變量控制 **ForEach** 內的組件數量...

@State let tries = 5 // 通過程式將 tries 改成 3, 畫面是否有改變呢?

// ~~~

```
HStack {  
    ForEach(0 ..< tries) { index in  
        Image(systemName: "heart.fill")  
            .foregroundColor(.red)  
    }  
}
```

答：不可以，因 ForEach 內的數量在 View 建立時已固定

If

```
@State var showWinning = false
@State var showGameOver = false

// ~~~

if !showWinning && !showGameOver {
    // 遊戲畫面
}
if showWinning {
    // 勝出畫面
}
if showGameOver {
    // 結束畫面
}
```

第二部分 - 遊戲畫面

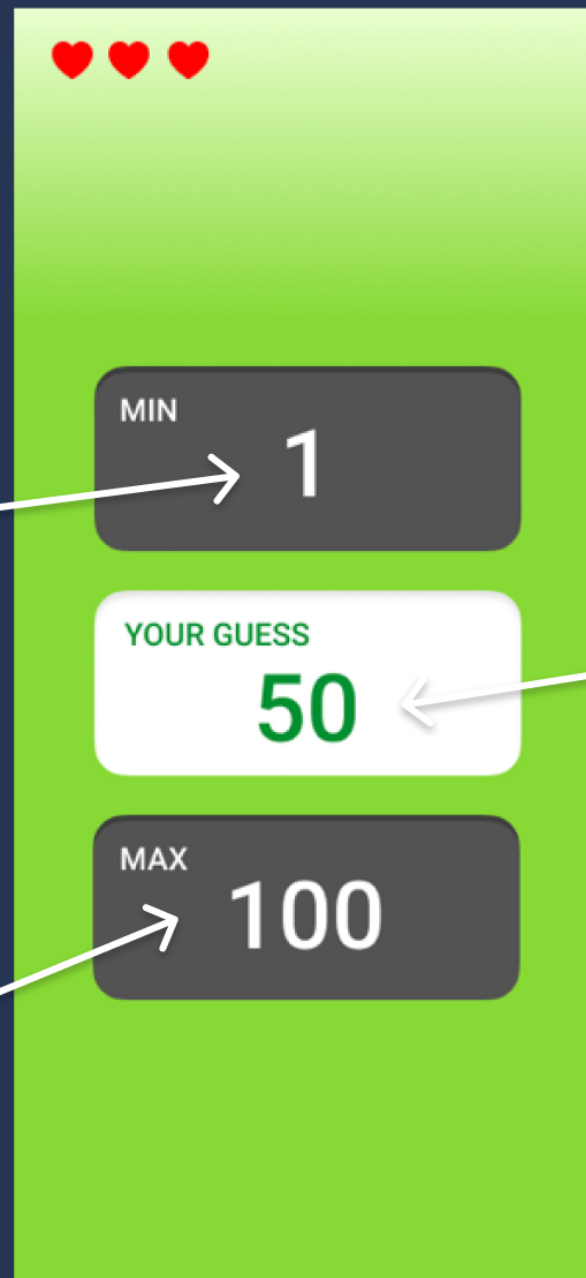
初始變量

```
@State var startingNumber = 1
@State var endingNumber = 100
@State var correctNumber = 25
@State var guessNumber = ""
```

correctNumber = ?
(不顯示出來)

startingNumber

endingNumber



guessNumber

按 **return** 時檢查

```
TextField("Guess a Number", text: $guessNumber, onCommit: {  
    // 檢查數字  
})
```

將功能成寫一個 **function**

```
TextField("Guess a Number", text: $guessNumber, onCommit: {  
    checkNumber()  
})
```

Func

```
func checkNumber() -> Void { // Void 表示不返回任何数值  
  
// ...  
  
}
```

檢查的步驟

1. 檢查是否數字 (用 if let)
2. 檢查是否在範圍內 1 - 100
3. 檢查數字
 - 檢查 [輸入內容] 是否 [小於] [正確數字]
 - 檢查 [輸入內容] 是否 [大於] [正確數字]
 - 檢查 [輸入內容] 是否 [等於] [正確數字] 🎉

檢查的步驟

當遇到 [小於] 或 [大於] 的情況時，需要扣除可嘗試次數

檢查的步驟

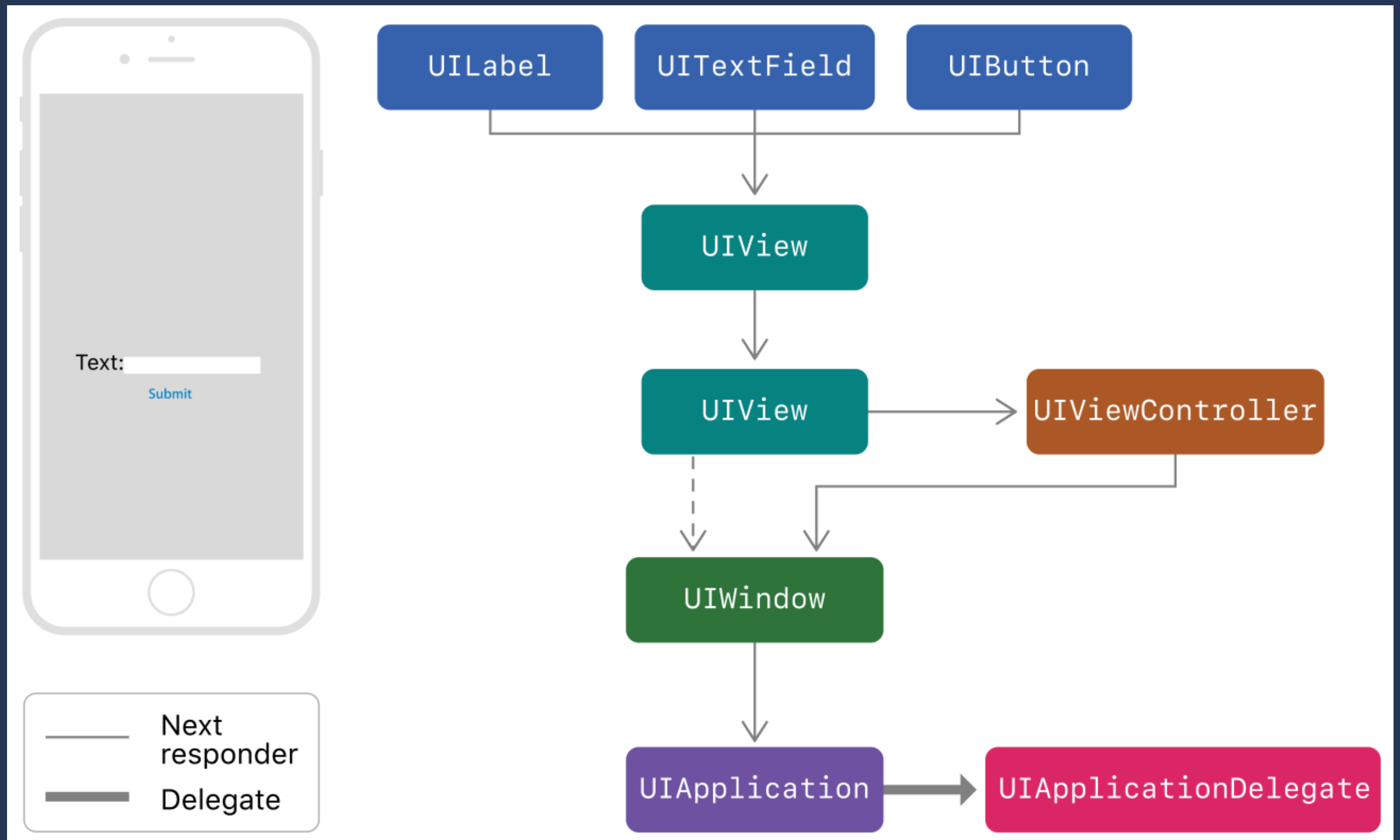
1. 檢查是否數字 (用 if let)
2. 檢查是否在範圍內 1 - 100
3. 檢查數字
 - 檢查 [輸入內容] 是否 [小於] [正確數字]
 - 檢查 [輸入內容] 是否 [大於] [正確數字]
 - 檢查 [輸入內容] 是否 [等於] [正確數字] 🎉
 - 如果已經沒有嘗試次數 - Game Over 😵

收起 Keyboard

```
// 收起 Keyboard
private func endEditing() -> Void {
    UIApplication.shared.sendAction(
        #selector(UIResponder.resignFirstResponder),
        to: nil,
        from: nil,
        for: nil
    )
}
```

收起 Keyboard

```
Button(action: {
    editing()
    checkNumber()
},
label: {
    Image(systemName: "arrow.right.square.fill")
})
.font(.largeTitle)
```



作業要求

1. 檔案命名：[你的全名].week4.zip
2. 檔案格式：必須將 Project 壓縮為 ZIP 檔後才上傳
3. 上傳至連結：<https://www.dropbox.com/request/Hh6CLWtcJdjuno2wf8UT>
4. 到期日：2021 年 7 月 28 日 (下下週三) 晚上 11 時前
5. 評分標準：基本完成要求滿分、空白 Project 或欠交沒有分數。