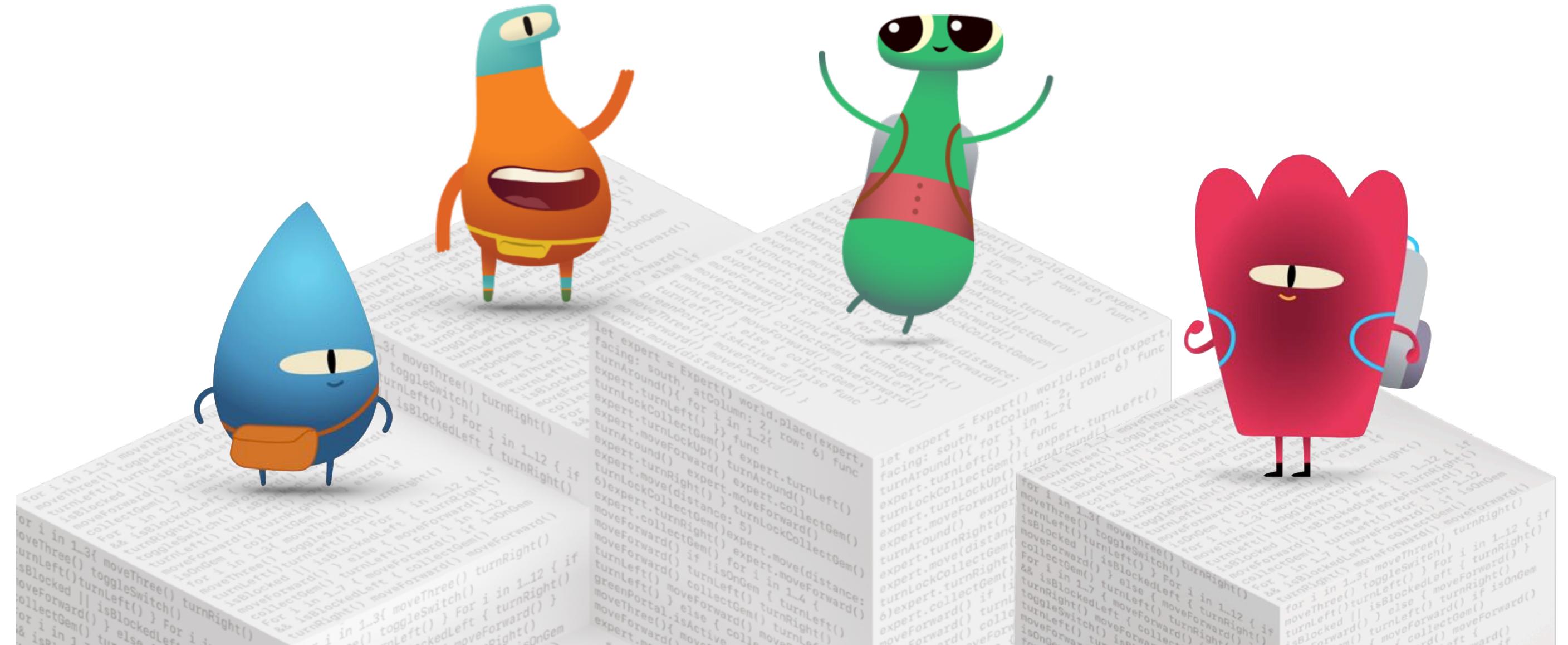




Swift Coding Club

Special Session



Session 4

Let's do Alerts for our Maths Game! + Consulting Session



```
for i in 1..3{ moveThree() toggleSwitch() turnLeft() isBlocked || turnLeft() && collectGem() moveForward() } else if i == 1 { moveForward() turnRight() } else { moveForward() turnLeft() } for i in 1..3{ moveThree() toggleSwitch() turnRight() isBlocked || turnRight() && collectGem() moveForward() } else if i == 1 { moveForward() turnLeft() } else { moveForward() turnRight() } for i in 1..3{ moveThree() toggleSwitch() turnLeft() isBlocked || turnLeft() && collectGem() moveForward() } else if i == 1 { moveForward() turnRight() } else { moveForward() turnLeft() } for i in 1..3{ moveThree() toggleSwitch() turnRight() isBlocked || turnRight() && collectGem() moveForward() } else if i == 1 { moveForward() turnLeft() } else { moveForward() turnRight() } for i in 1..3{ moveThree() toggleSwitch() turnLeft() isBlocked || turnLeft() && collectGem() moveForward() } else if i == 1 { moveForward() turnRight() } else { moveForward() turnLeft() } for i in 1..3{ moveThree() toggleSwitch() turnRight() isBlocked || turnRight() && collectGem() moveForward() } else if i == 1 { moveForward() turnLeft() } else { moveForward() turnRight() } for i in 1..3{ moveThree() toggleSwitch() turnLeft() isBlocked || turnLeft() && collectGem() moveForward() } else if i == 1 { moveForward() turnRight() } else { moveForward() turnLeft() } for i in 1..3{ moveThree() toggleSwitch() turnRight() isBlocked || turnRight() && collectGem() moveForward() } else if i == 1 { moveForward() turnLeft() } else { moveForward() turnRight() }
```



Math x game



Math Game - **ADVANCE**

Step 5

Alert

The alert shows "Game Over", the player's score, and asks if they want to play again with "Yes" and "No" buttons. "Yes" restarts the game, while "No" just closes the alert.

The screenshot shows the Xcode interface with the ContentView.swift file open in the main editor. The code defines a struct ContentView that contains a var body and an alert block. The alert displays "Game Over" and the player's score. It includes two buttons: "Yes" (which restarts the game) and "No" (which dismisses the alert). The right side of the interface shows the iPhone 15 Pro Max simulator running the game. The screen shows a math problem (8 × 7 = ?), a score of 0, and a text input field for the user's answer. A "Check Answer" button is at the bottom. The status bar at the top right indicates "Running MathGame on iPhone 15 Pro Max".

```
11 struct ContentView: View {
23     var body: some View {
46         .padding()
47         .background(Color.black)
48         .foregroundColor(.white)
49         .cornerRadius(8)
50     }
51     .padding()
52     .alert(isPresented: $showGameOverAlert, content: {
53         Alert(
54             title: Text("Game Over"),
55             message: Text("Your total score is \(score)."),
56             primaryButton: .default(Text("Yes"), action: startNewGame),
57             secondaryButton: .cancel(Text("No"))
58         )
59     })
60 }
61
62 func nextProblem() {
63     number1 = Int.random(in: 1...10)
64     number2 = Int.random(in: 1...10)
65     userAnswer = ""
}
```



Math x game



Math Game - **ADVANCE**

```
.alert(isPresented: $showGameOverAlert, content: {  
    Alert(  
        title: Text("Game Over"),  
        message: Text("Your total score is \$(score). Do you want to play again?"),  
        primaryButton: .default(Text("Yes"), action: startNewGame),  
        secondaryButton: .cancel(Text("No"))  
    )  
})
```

Step 5 - `.alert` - Explain

- **`.alert(isPresented: $showGameOverAlert, content: {...})`**: This is a modifier applied to a SwiftUI view. It displays an alert dialog when the condition specified by **showGameOverAlert** becomes true. The **\$** before **showGameOverAlert** indicates that this variable is a **@State** variable, meaning it's a source of truth for the view and changing its value will cause the view to update.

Follow the highlight



Math x game



Math Game - **ADVANCE**

```
.alert(isPresented: $showGameOverAlert, content: {  
    Alert(  
        title: Text("Game Over"),  
        message: Text("Your total score is \$(score). Do you want to play again?"),  
        primaryButton: .default(Text("Yes"), action: startNewGame),  
        secondaryButton: .cancel(Text("No"))  
    )  
})
```

Step 5 - Alert - Explain

- **Alert(...)**: Inside the **alert** modifier, an **Alert** structure is being created. This defines the content and behavior of the alert dialog.

Follow the highlight



Math x game



Math Game - **ADVANCE**

```
.alert(isPresented: $showGameOverAlert, content: {
    Alert(
        title: Text("Game Over"),
        message: Text("Your total score is \$(score). Do you want to play again?"),
        primaryButton: .default(Text("Yes"), action: startNewGame),
        secondaryButton: .cancel(Text("No"))
    )
})
```

Step 5 - title - Explain

- **title: Text("Game Over"):** This sets the title of the alert to "Game Over".



Follow the highlight



Math x game 🎮 ×

Math Game - **ADVANCE**

```
.alert(isPresented: $showGameOverAlert, content: {
    Alert(
        title: Text("Game Over"),
        message: Text("Your total score is \$(score). Do you want to play again?"),
        primaryButton: .default(Text("Yes"), action: startNewGame),
        secondaryButton: .cancel(Text("No"))
    )
})
```

Step 5 - message - Explain

- **message: Text("Your total score is (score). Do you want to play again?"):** This sets the message in the alert. It includes dynamic text showing the **score** variable, presumably representing the user's score in the game.

Follow the highlight



Math x game



Math Game - **ADVANCE**

```
.alert(isPresented: $showGameOverAlert, content: {
    Alert(
        title: Text("Game Over"),
        message: Text("Your total score is \$(score). Do you want to play again?"),
        primaryButton: .default(Text("Yes"), action: startNewGame),
        secondaryButton: .cancel(Text("No"))
    )
})
```

Step 5 - **primaryButton** - Explain

- **primaryButton: .default(Text("Yes"), action: startNewGame)**: This defines the primary button in the alert. It's labeled "Yes", and when tapped, it triggers the **startNewGame** function, which likely resets the game to its initial state.

Follow the highlight



Math x game



Math Game - **ADVANCE**

```
.alert(isPresented: $showGameOverAlert, content: {  
    Alert(  
        title: Text("Game Over"),  
        message: Text("Your total score is \$(score). Do you want to play again?"),  
        primaryButton: .default(Text("Yes"), action: startNewGame),  
        secondaryButton: .cancel(Text("No"))  
    )  
})
```

Step 5 - **secondaryButton** - Explain

- **secondaryButton: .cancel(Text("No")):** This defines a secondary, or cancel button, with the label "No". Tapping this button will simply dismiss the alert without performing any additional action.

Follow the highlight



Swift Coding Club *Thailand*

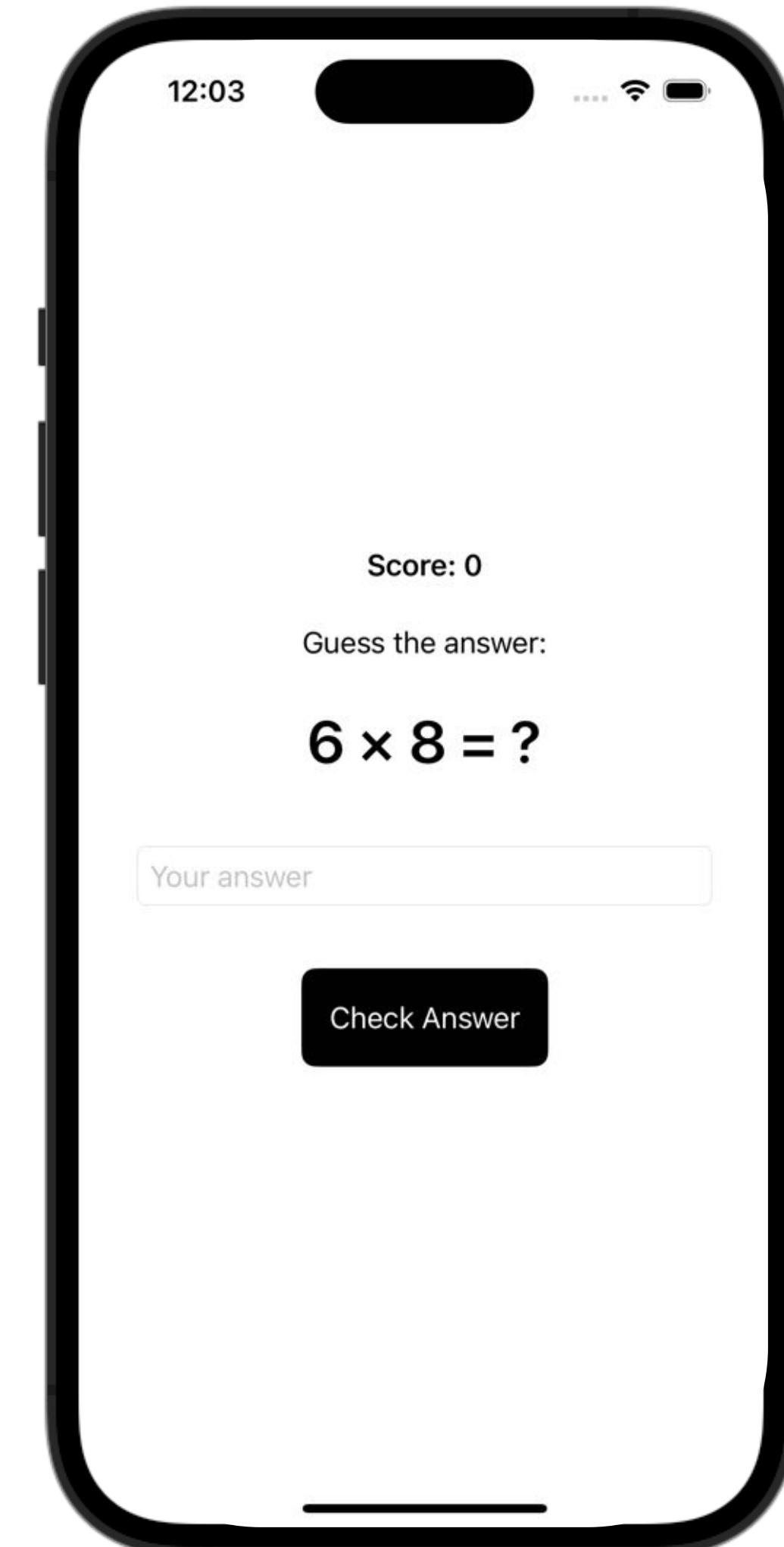
Math Game - **ADVANCE**

Done



-

Math x game





Swift Coding Club *Thailand*

© 2024 Swift Coding Club Thailand, Apipoom Chuenchompoo.

This work is licensed by Swift Coding Club Thailand. under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International license