Whack A Mole

Adding UI components (and 'outlets' for interacting with them) in a drag-and-drop manner using Interface Builder is convenient! But it's important to know what's happening when you do that:

> *When you add (for example) a UIButton to your app's view with IB, Xcode is creating a serialized version of the object and embedding it in a VERY messy (not easily human-readable) XML file. There is no easily readable 'code' for the button, in a manner of speaking.*

Therefore, it's very important to know that anything that can be done with IB can also be done with code; developers should be able to use either method, whichever is appropriate for the situation. In this project, you'll make a simple 'whack a mole' type game using a timer and a button, done entirely programmatically.

1. Set up the game board.
    a. Find the size of the screen.
    b. Set up the score box:
        i. Make a label and place it at the top left corner of the screen.
        ii. Make it 1/10th the height of the screen.
        iii. Initialize the score text with "0".
    c. Paint the field:
        i. Make a label and paint it light green.
        ii. Size the label to cover the entire screen below the score (except for the screen edge margin)
    d. **Test your program before continuing.** Simulate on a newer iPhone.
2. Put a mole on the screen.
    a. Make a dark colored button and place it near the top of the field.
    b. Add an event handler function; have it print to the console. Test your button before continuing!
    c. Increment the score when the button is tapped. Test this feature before continuing!
    d. Make the button disappear when tapped. Test this feature!
3. Make the mole move.
    a. Make the button appear at a random location. Test this feature!
    b. Add a 5 second timer to the app (you can use less time to make it more challenging). Make the button disappear when the timer runs out. Test this feature!
4. Put it all together.
    a. Refactor your code so the app runs as follows (test each feature as it's added):
        i. After launch, the field, the score label with zero, and the 'mole' button appears.
            1. The mole button should begin in the top-left corner of the field, with text telling the user to tap to start.
                a. Once the button is initially tapped, the game begins.
        ii. If the player hits the mole within 5 seconds, make the mole disappear and increment the score. Make a new mole appear in a new random location. Restart the timer.
            1. View (option + click) Timer's documentation to learn more about it; reading the official docs is a skill programmers must develop over time. If you get stuck, you can find more timer information here.

iii. If the player fails to hit the mole in time, make the mole disappear and decrement the score. Make a new mole appear in a new random location. Restart the timer.

5. Test your Whack-A-Mole thoroughly!

Extension Ideas

- Add messages for missing the mole.
- Add a mole image to the button.
- Choose a random size for the mole. Add more points for smaller targets.
- Choose a random countdown timer for the mole. Add more points for shorter timers.
- Add colors to the mole. E.g., use red for higher-point moles and green for easy targets.
- *Challenging*. Add a stopwatch to measure the time taken to hit the mole. Add more points for faster reaction time.
- *Challenging*. Add more moles to the screen at the same time. Each needs its own timer.