

Assignment 1 - Mastermind

Ali Abod - 201267800

How to Use

The app is quite simple to use. There is a view at the bottom of the screen in order to enter your combination guess. In order to see your input, there is a view that shows your input so far. You can add pegs to your guess by clicking the respective colour and remove pegs from your typed input by clicking the delete button on the keyboard. Once you are happy with your guess click the green tick in the keyboard to confirm your guess. This button is only activated once your typed input is 4 pegs long.

There is a table view that displays your guesses. This has been implemented using a single cell view which is the template used for all 10 cells. This shows your guess as well as its correctness (using black and white pegs).

At the top there is information about your remaining attempts, a history button and a restart button.

The remaining attempts label uses a computed property to calculate the number of attempts remaining.

The history button shows a pop-up message with your wins and losses over time. This is a stored property using persistent storage.

The restart button is to restart the game with a new combination in case you want to restart the game half-way through a game.

Notable Features

I've implemented a pop-up function that displays a message and an action button. This is used for when the game ends, and shows a restart button. This is also used when user taps "History" to show the number of total wins and losses.

I've also redone the UI to clean cleaner, more modern and flatter.

I've implemented a restart button at the top to restart the game with a new combination in case you want to restart the game half-way through a game.

I've made the structure of my code organised and consistent, using separate classes and files for each feature. This means if I ever wanted to improve on my project or change some details it would not require much work.

For testing, I've printed the secret code combination in the output.