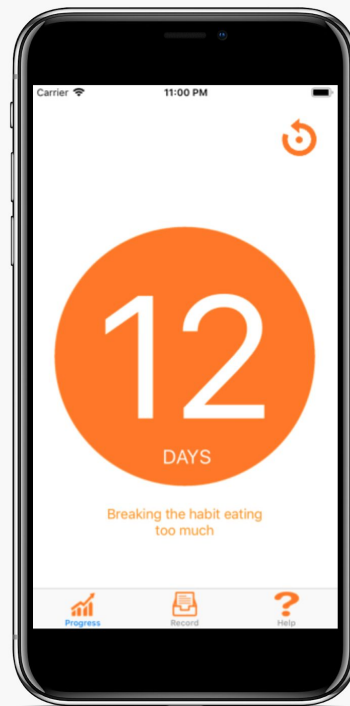


Habit Breaker



Mike Carbone
IDM362-001

The Overview

HabitBreaker is an app made for the iPhone to help you break bad habits. The idea behind it was simple: I needed an easy way to keep track of a habit I was actively trying to break. So, I created HabitBreaker, focusing on a simple interface with clear functionality.

The Context and Challenge

1. This app was designed to be very simple, for a wide audience, therefore:
 - a. Wanted to keep it easy to use
 - b. Easy to understand
 - c. Quick to pickup
2. The goal was to make it simple and keep scope small, especially considering this was a platform I've never developed on
3. Success can be determined by one simple measurement: it is able to keep track of your current habit. Everything else is considered extra.

The Process and Insight

Development of this application was, in my opinion, very difficult. It was unlike any programming language I've ever used before and a bulk of my time was spent problem solving and trying to

figure out the correct way to do it in Swift. Many of the problems I faced, I could've easily figured out in JavaScript, making the language itself the source of many of my problems.

New symbols like the question mark, were used in many different places, in many different contexts, performing different functions each time. Many elements of Swift proved to be similar to the question mark, slowing down development as I figured out these issues.

Transitioning from JavaScript development, where seemingly anything goes, to Swift, the strictest language I've ever coded in, made things very troubling. I faced many new problems that I wasn't aware even existed in programming. After this development process, I am eternally grateful for the ease of JavaScript and can confidently say building apps in Swift is the worst.

As for asset development, that was probably the easiest part of creating this application. Since simplicity was key here, the assets were simple and quick to create.

The Results

All in all, I consider the project a success. While there are many bugs, memory leaks, and several different problems, the core functionality is there and it looks pretty nice. I also learned a lot throughout the process, giving me a wider understanding of what it means to be a front-end developer, and more context to what is needed on the backend for an iPhone app. Development is slow, tricky, and there are many ways to do it wrong. For my first Swift project, I think it's okay and I am proud of it.