Kampa is an experimental general-purpose imperative programming language influenced by functional programming. While Kampa offers several potentially useful features, its usability is limited by some properties of its current implementation as well as of the language itself. We add support for coroutines and parameter inference, making the language more expressive. We also make changes to the syntax that improve code readability, in particular reducing the tendency to "line noise". We refine the implementation to remove several arbitrary restrictions regarding generic and dependent types, enabling generic type definitions (among others). Lastly, we demonstrate the practicality of the result by writing a library providing collections, optional, asynchronous programming, generators, and string utilities.