Makoto Keyboard

This is iOS test application to demonstrate work with speech recognition in Keyboard extension.

App architecture

Overview. App consists of:

- **Makoto** - main app;
- **MakotoKeyboard** - Keyboard extension, which also contains KeyboardView
- **KeyboardView** - custom view of keyboard, can display different states (recognition is disabled, enabled, in progress, failed); contains only presentation (UI) logic but know nothing about recognition frameworks/logic;
- **SpeechRecognition** - framework for speech recognition; handle all logic related to speech recognition;

Principles used during development:

- **Modular architecture** - app consists of frameworks/modules. In current case all logic related to speech recognition is placed in separate framework **SpeechRecognition.**
- **MVVM** - pattern of presentation layer to separate data preparation/formatting from UI configuration. **Combine** was used for bindings and to organise data flow.
- **Unidirectional data flow** - to organise data flow between components.
- **POP** - protocol oriented programming aka Dependency inversion principle of SOLID
- **SRP** - single responsibility principle of SOLID
- Other SOLID, KISS, OOP principles

Libraries

- **Combine** - native lib for reactive programming
- **Speech, AVFoundation** - native libs for speech recognition
- **Lottie** - third-party lib for play animation
- **Reusable** - third-party small but effective library for nib loading (just to minimise lines of code)

SPM used for manage libraries;

Known issues/bugs

Default dictation button appears on custom keyboard after speech recognition. Need to research how to solve it

Improvements

Add themes support for **KeyboardView**
Add localisation
Use Combine also for **SpeechRecognition** framework

License