This Flutter/MX Makeover

Episode I: The flutter_tts demo refactor

The f/mx knockoff of the Flutter_Tts demo

Here is where we started. A reasonably close copy of the official flutter_tts demo:

```
https://github.com/dlutton/flutter_tts/blob/
master/example/lib/main.dart
```

We noticed we did not understand having both play and pause visible at the same time.

We thought one button, as far as the user knew, should toggle between play and pause.

We were in for a great ride. And we think it will be a great introduction to Matrix and Flutter both.
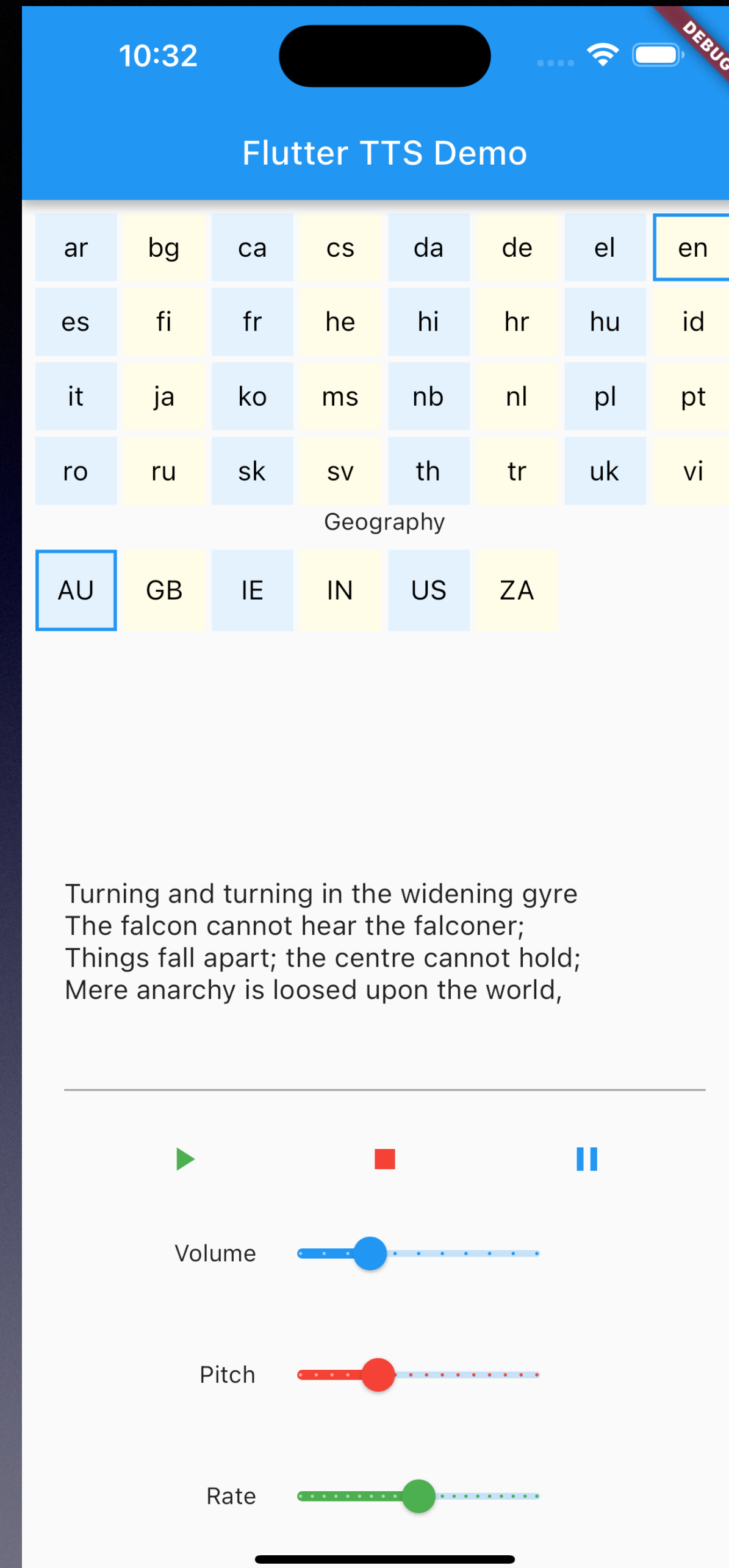
Hushing Auto-play

Auto-play at start-up is nice to dazzle the audience, but we will be developing iteratively, and it will get old.

And anyway, turning it off lets us take an early deep dive into the MX life cycle. Good to cover these as they come up, so the reader always has some realistic context.

The life cycle in brief: any new model such as our app gets immediately "awakened" simply by evaluating all its formulas.

Here, the `lg-current` rule takes the `lg-pref` as an initial value then becomes an input cell, to accept future user choices.

Awakening also runs all watch functions. This one calls 'do-speak', so we hear speech right at app start-up.

Next we will look at how we defeat that.

```clojure
:lg-pref        "en-AU"
; "current lang-geo"
; as a convenience, we "auto-play" when the user
selects a new one.
:lg-current (cF+n [:watch (fn [_ me new prior _]
                            (do-speak me))]
              (mget me :lg-pref))
```

Hushing Auto-play, Part II

We can use some internals
knowledge to detect the start-up
auto-play.

One might wonder what will be the
"prior" value when the *first* is
calculated. MX delivers a gensym:

```
(def unbound

  (gensym "unbound-cell-value"))
```

And thanks to how the app starts
up, we can use that to detect and
hush the initial app auto-play.

```
:lg-current (cF+n
              [:watch (fn [_ me new prior _]
                  (when (not= prior mx/unbound)
                    (do-speak me)))]
              ; the rule…
              (mget me :lg-pref))
```