

Managing App Complexity

Chris Vasselli

@chrisvasselli

Malaga Mobile, March 2022

Agenda

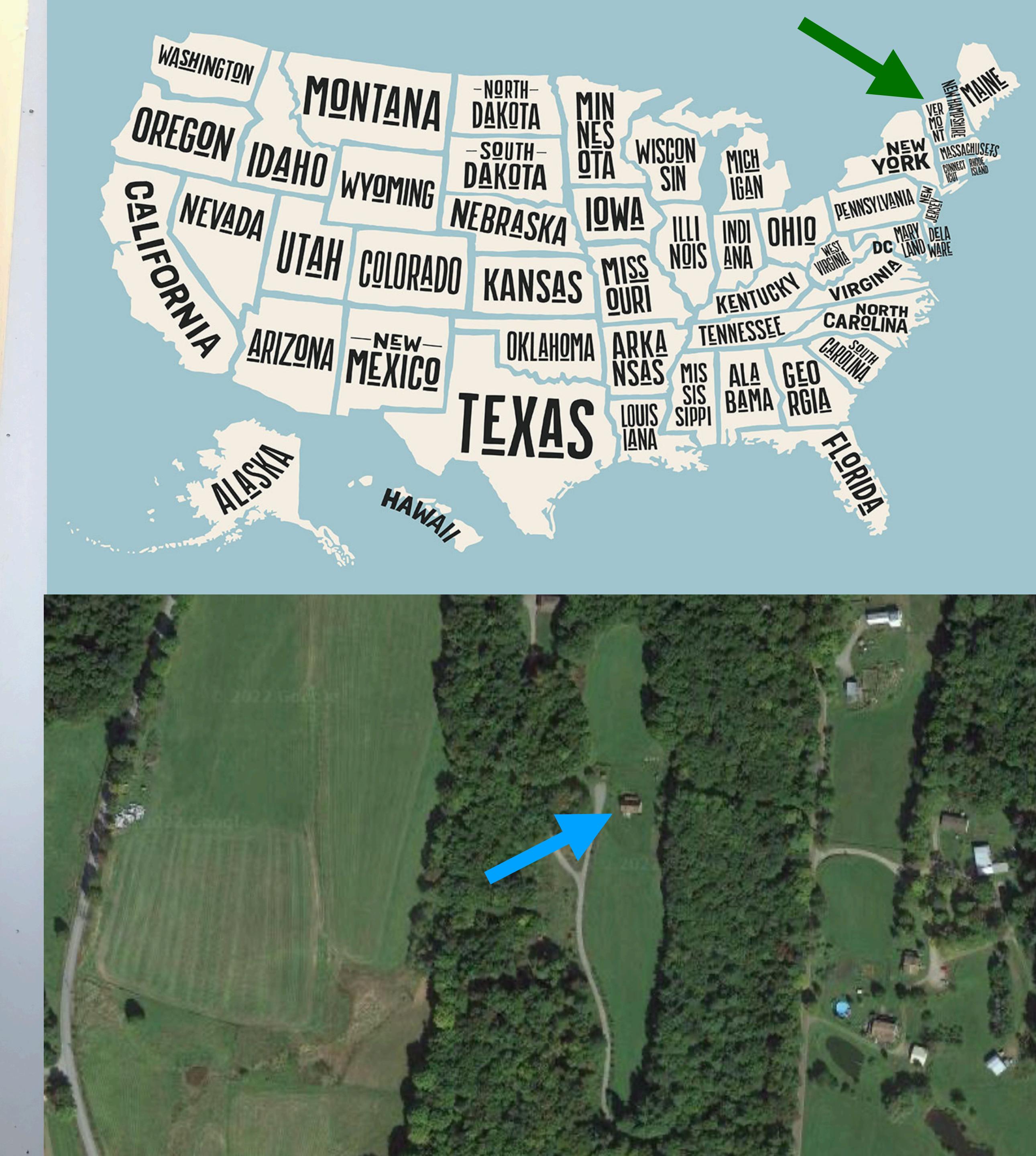
About Me

What do I mean by complexity?

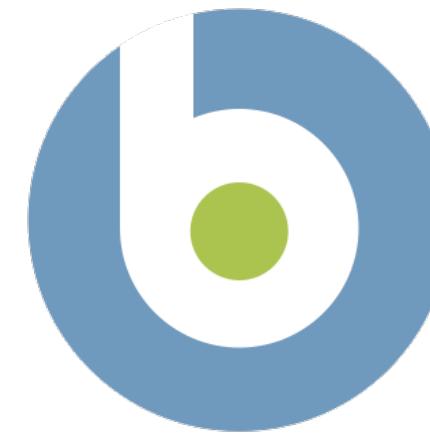
Managing complexity through testing

Tools I use

About Me







BIG FIX



box



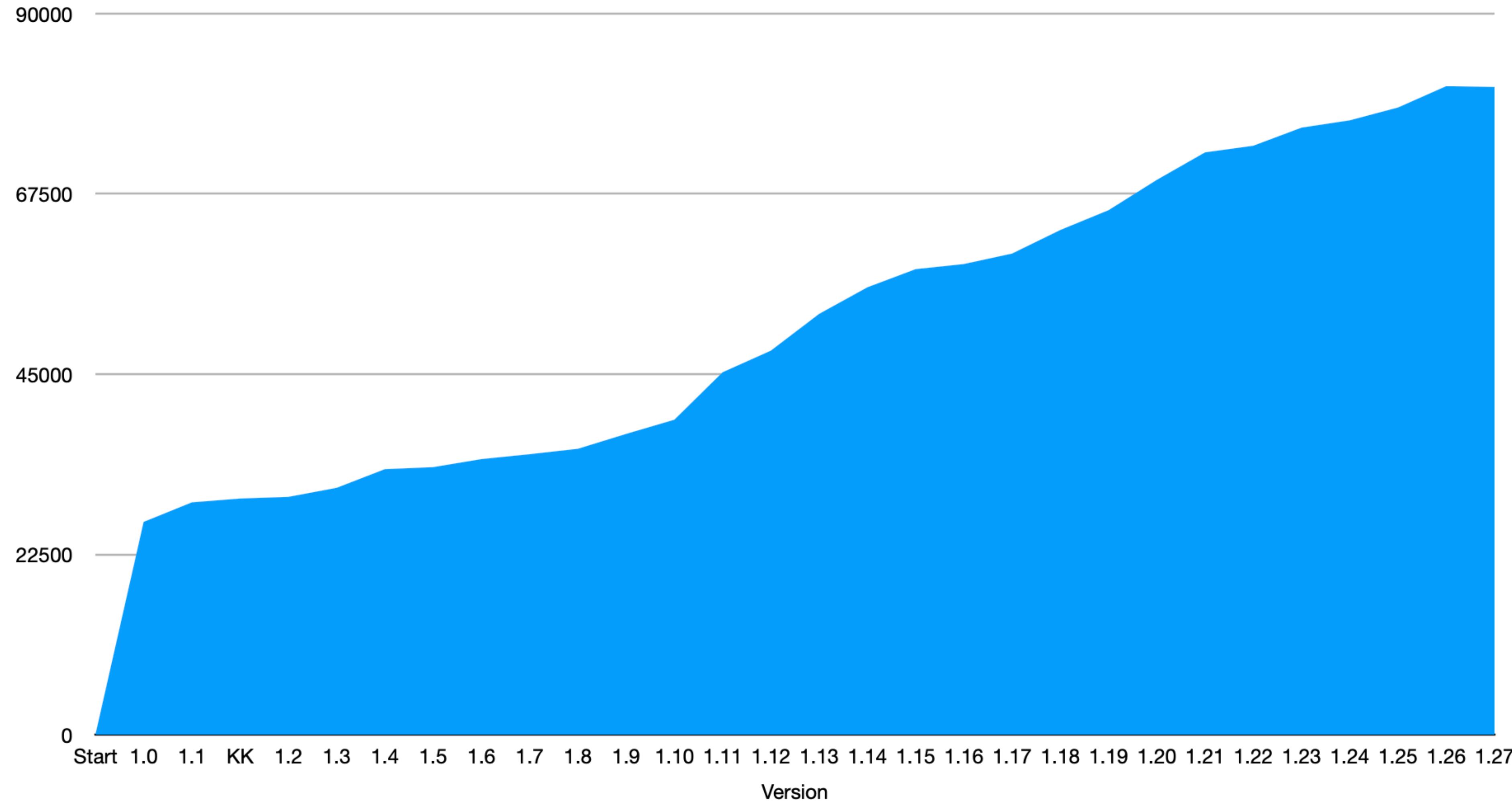
NIHONGO

A modern Japanese dictionary and study tool.

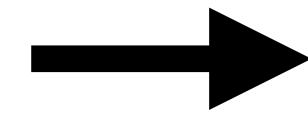
Managing App Complexity

What do I mean by complexity?

Nihongo Lines of Code



3x more code



3x more complex?

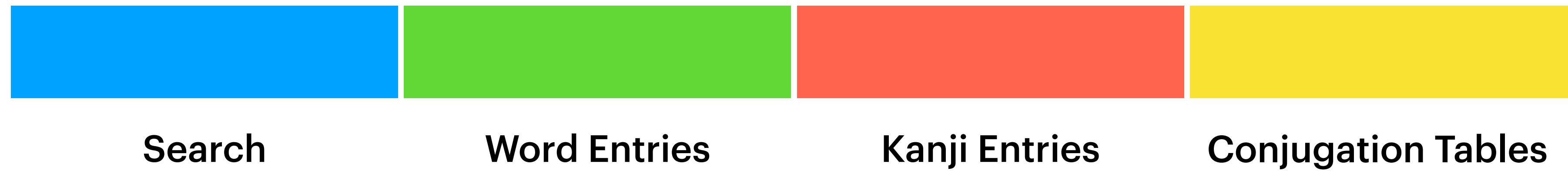
No.

Nihongo 1.0



Total Complexity: **3**

Nihongo 1.1



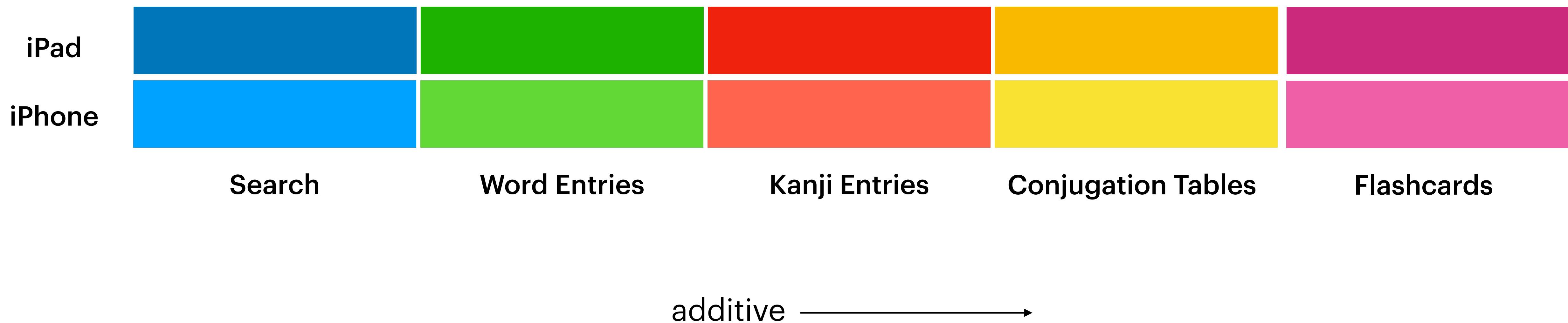
Total Complexity: $3 + 1 = \mathbf{4}$

Nihongo 1.2



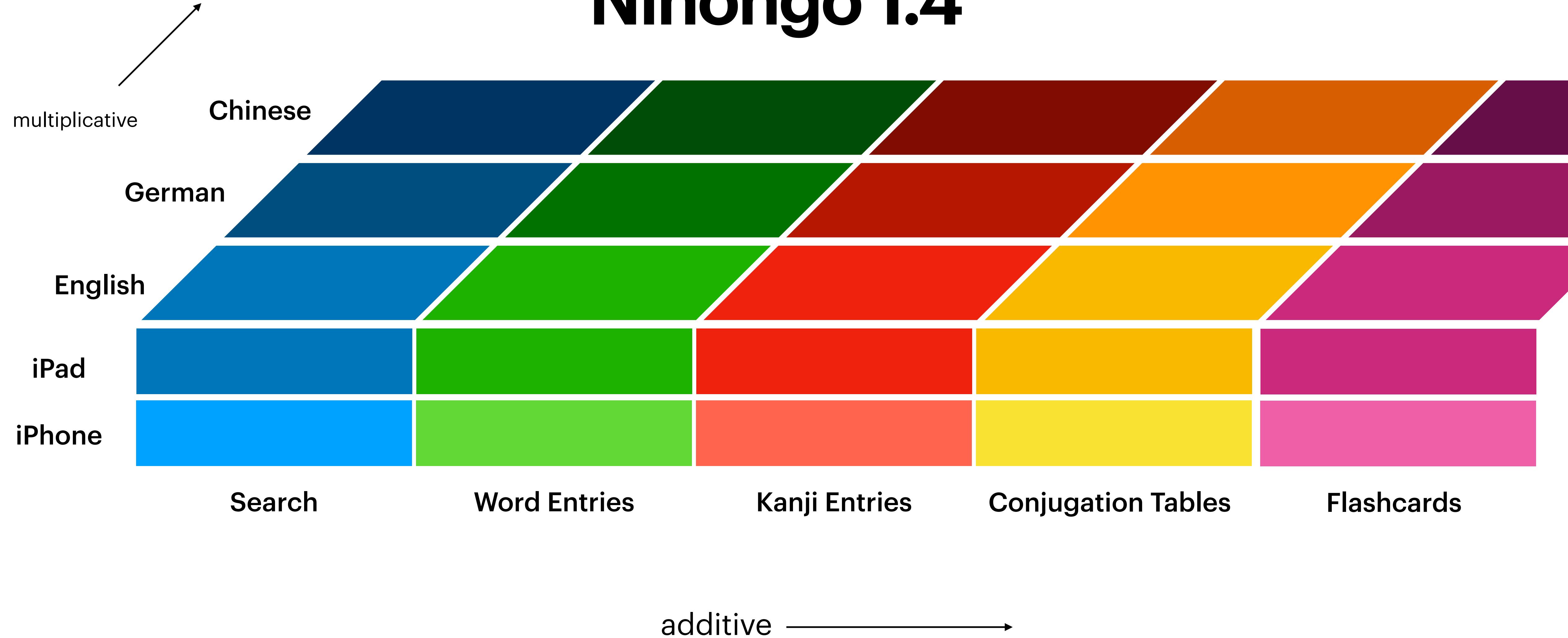
Total Complexity: $4 \times 2 = 8$

Nihongo 1.3



Total Complexity: $8 + 2 = \mathbf{10}$

Nihongo 1.4

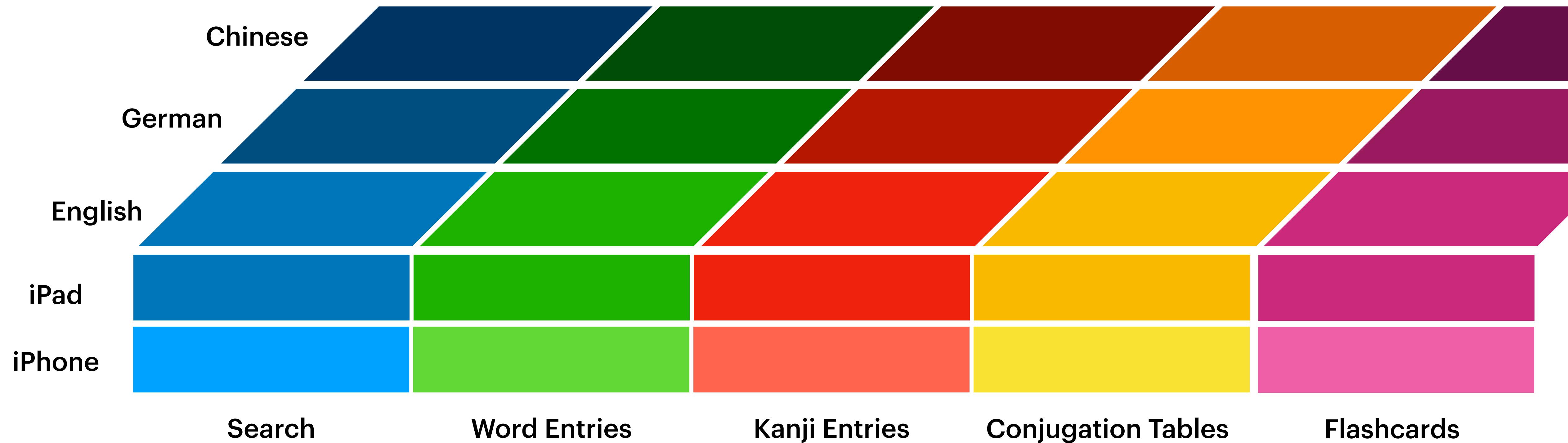


Total Complexity: $10 * 3 = \mathbf{30}$

Additive / Multiplicative Features

- Widgets - additive
- Handwriting input - additive
- Dark mode - multiplicative
- Dynamic type - multiplicative
- Support for multiple iOS versions - multiplicative

Nihongo 1.4



Nihongo 1.27

Search Screen

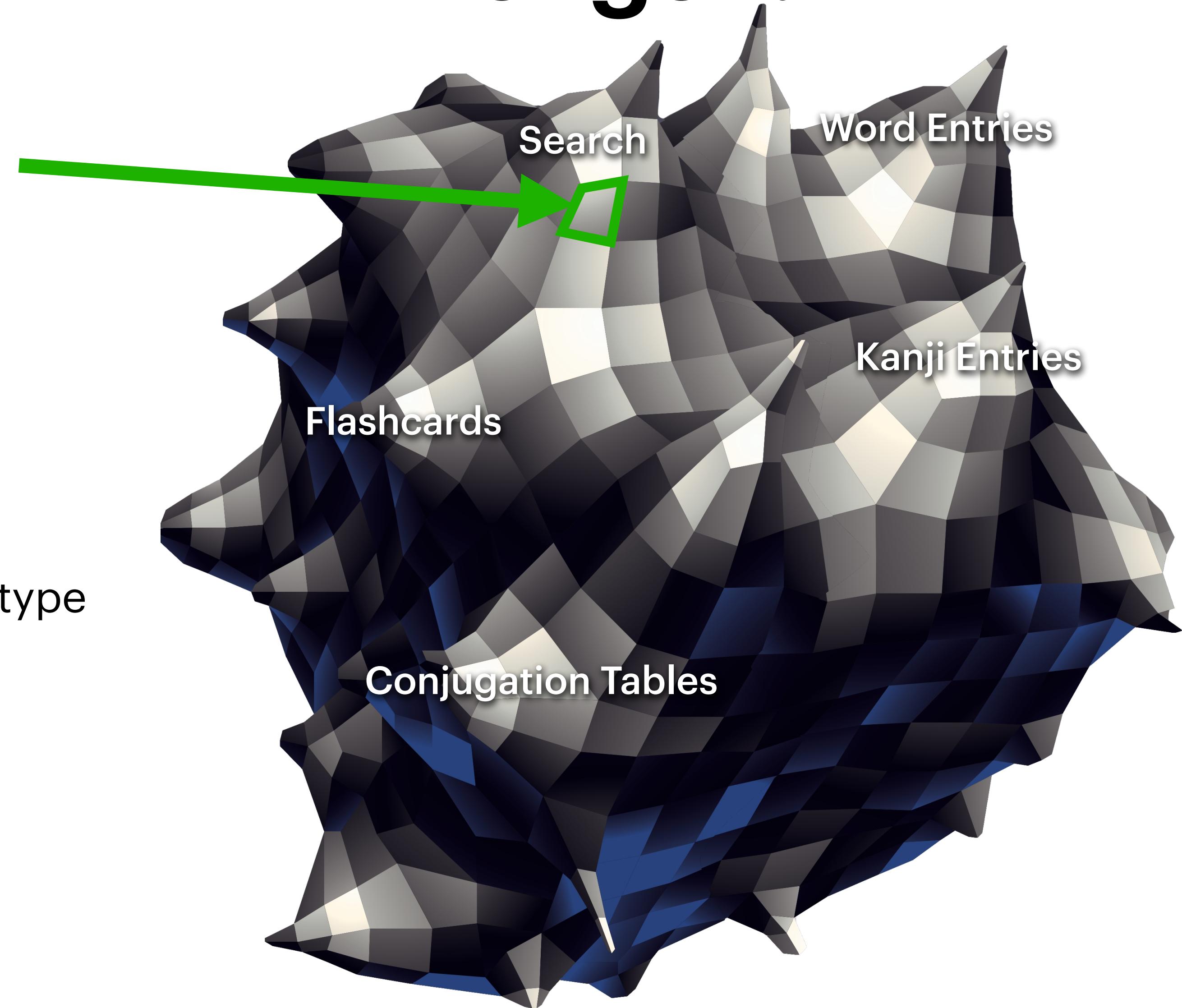
in German

on an iPad

in Dark Mode

on iOS 14

with extra large dynamic type



Managing Complexity

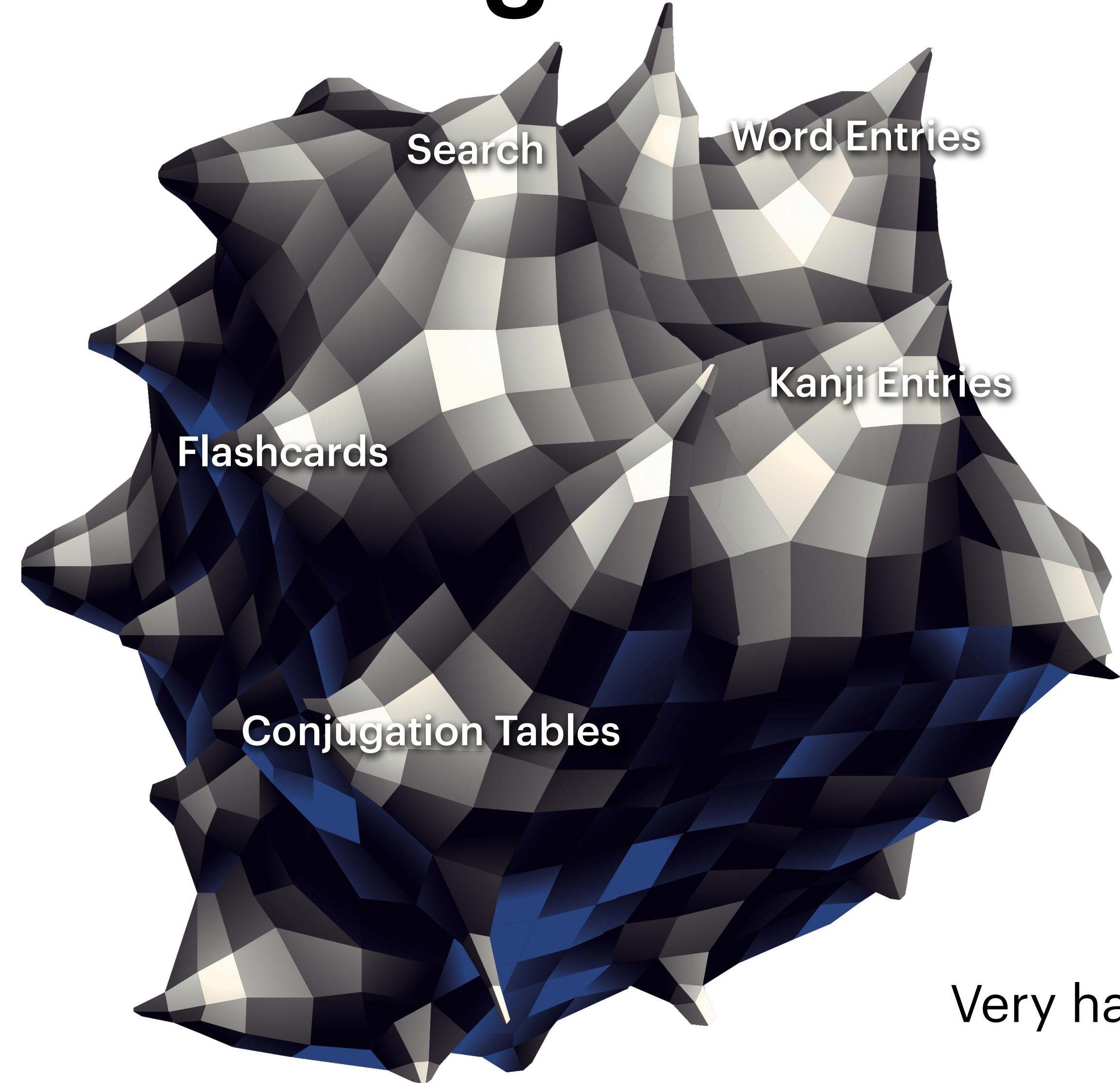
Difficulty of verifying correctness of an app scales with complexity

Nihongo 1.0



Easy to verify it's working

Nihongo 1.27



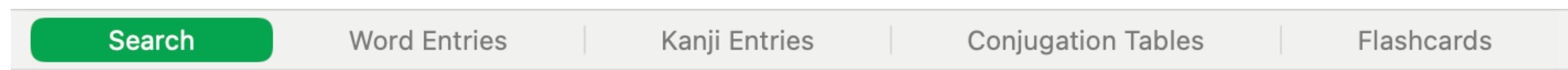
Very hard to verify it's working

Managing Complexity

Difficulty of verifying correctness of an app scales with complexity

Having confidence that your app works as it's supposed to, even as it grows.

Test Plan

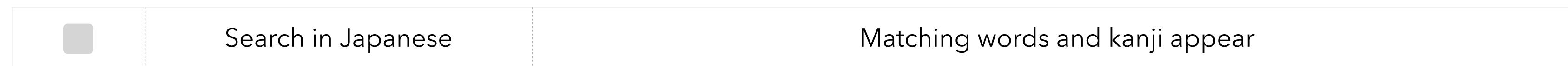


SEARCH

		Desired Behavior
	Basics	
<input type="checkbox"/>	Search in Japanese	Matching words and kanji appear
<input type="checkbox"/>	Search in English	Matching words appear
<input type="checkbox"/>	Search again	Results are replaced when you hit return
	Transitions	
<input type="checkbox"/>	Tap Kanji cell	Correct kanji entry appears
<input type="checkbox"/>	Tap Word cell	Correct word entry appears

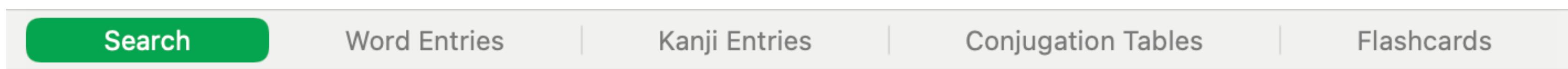
Automation (XCUI Test)

SEARCH



```
func testSearchInJapanese() {  
    let searchField = app.searchFields["dictionarySearchField"]  
    searchField.typeText("大")  
    XCTAssert(app.cells["大きい"].exists)  
    XCTAssert(app.cells["大 (KANJI)"].exists)  
}
```

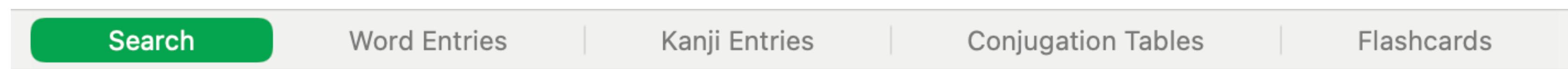
Test Plan



SEARCH

		Desired Behavior
Basics		
	Search in Japanese	Matching words and kanji appear
<input type="checkbox"/>	Search in English	Matching words appear
<input type="checkbox"/>	Search again	Results are replaced when you hit return
Transitions		
<input type="checkbox"/>	Tap Kanji cell	Correct kanji entry appears
<input type="checkbox"/>	Tap Word cell	Correct word entry appears

Test Plan



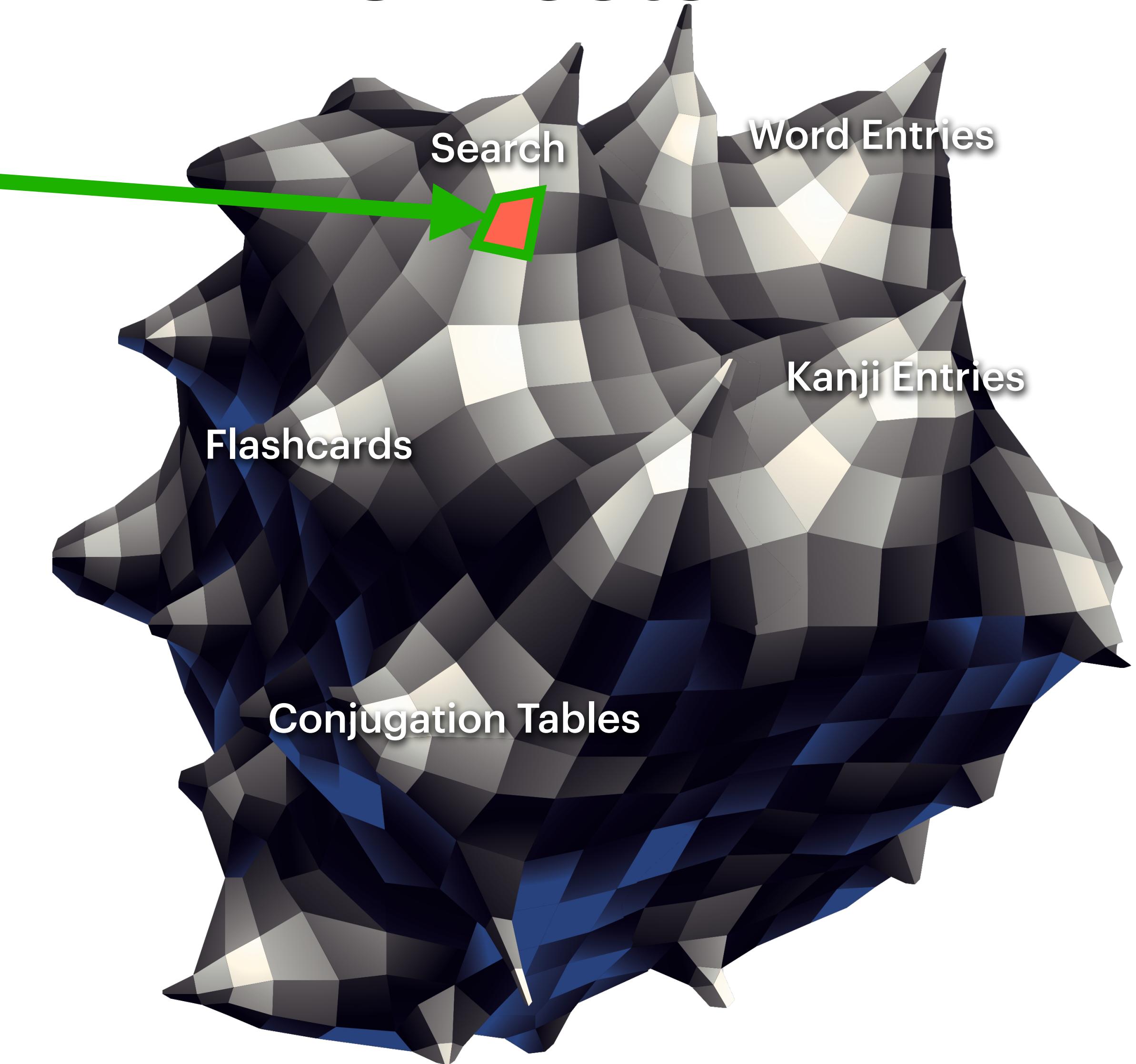
SEARCH

		Desired Behavior
	Basics	
	Search in Japanese	Matching words and kanji appear
	Search in English	Matching words appear
	Search again	Results are replaced when you hit return
	Transitions	
	Tap Kanji cell	Correct kanji entry appears
	Tap Word cell	Correct word entry appears

UI Tests

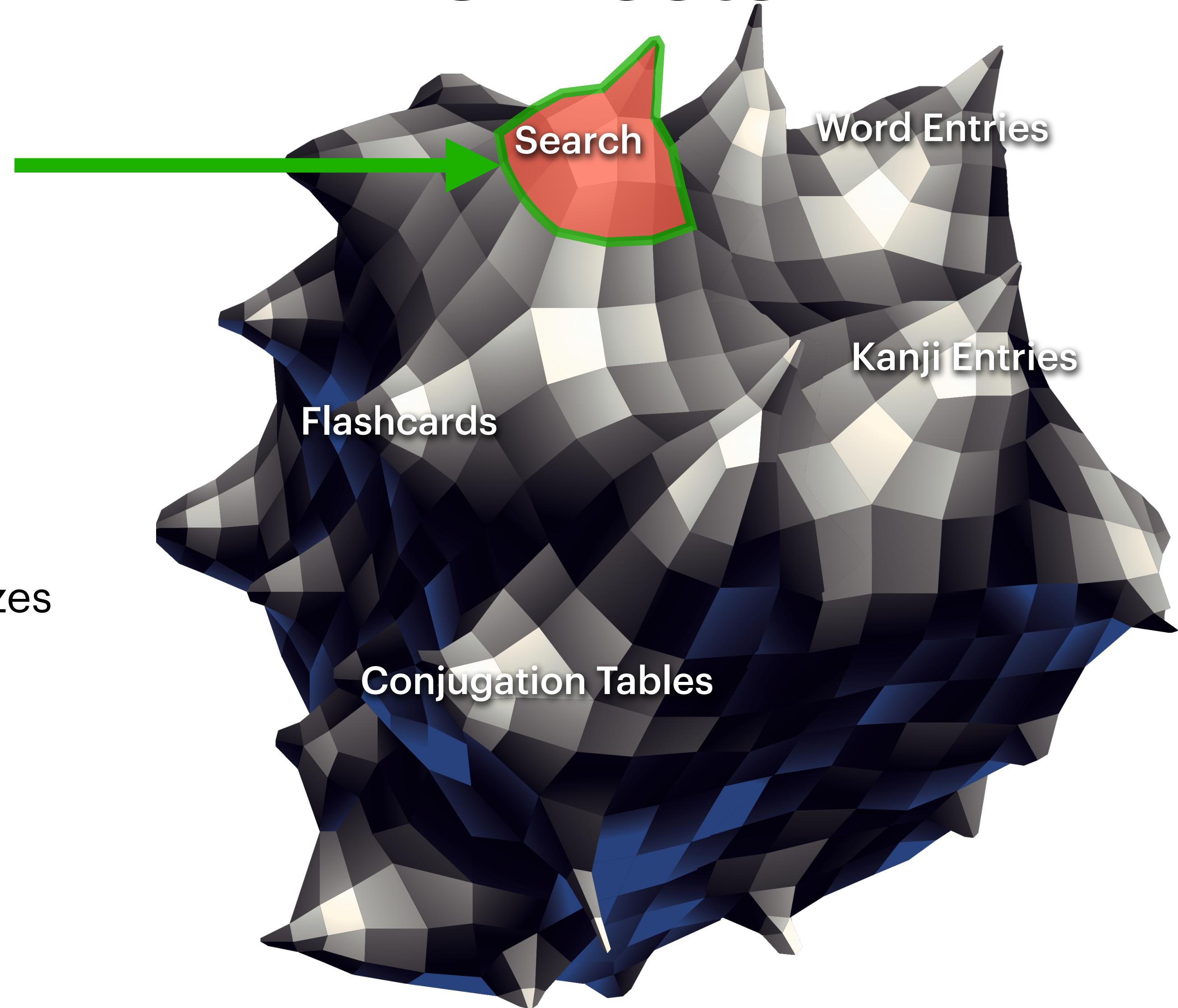
Search Screen
in German
on an iPad
in Dark Mode
on iOS 14

with extra large dynamic type

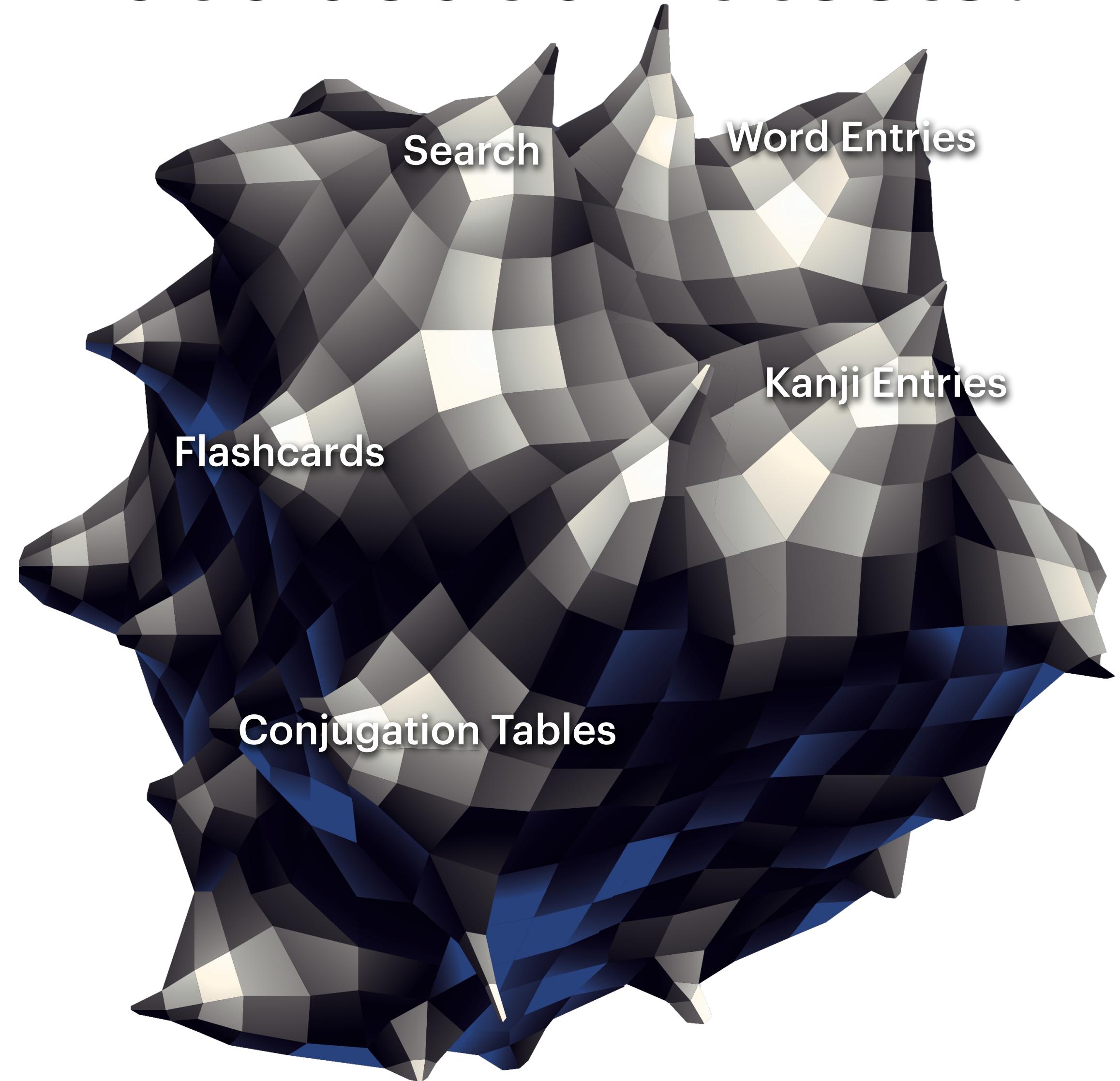


UI Tests

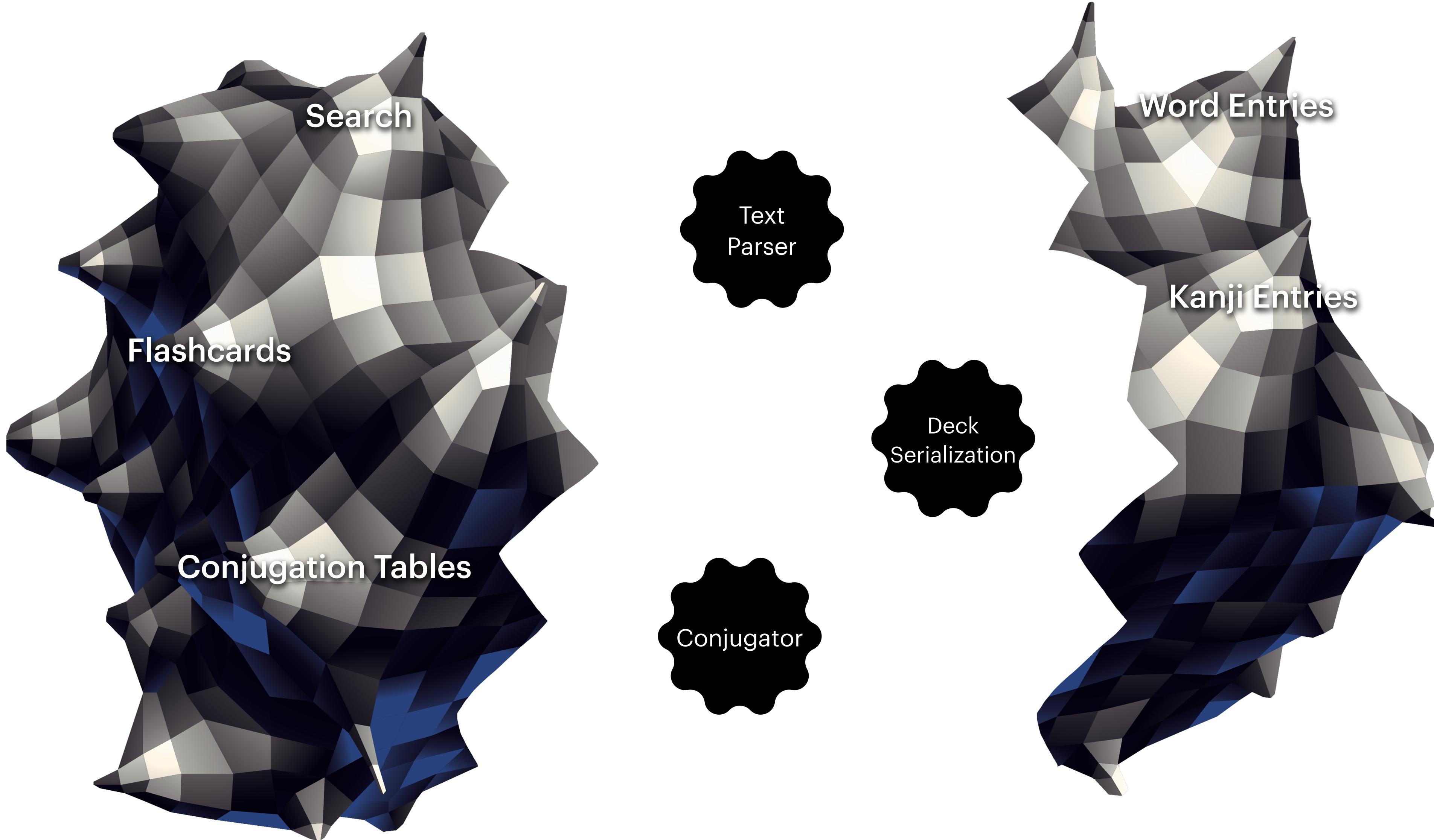
Search Screen
in all languages
on all devices
in light/dark mode
on all iOS versions
with all dynamic type sizes



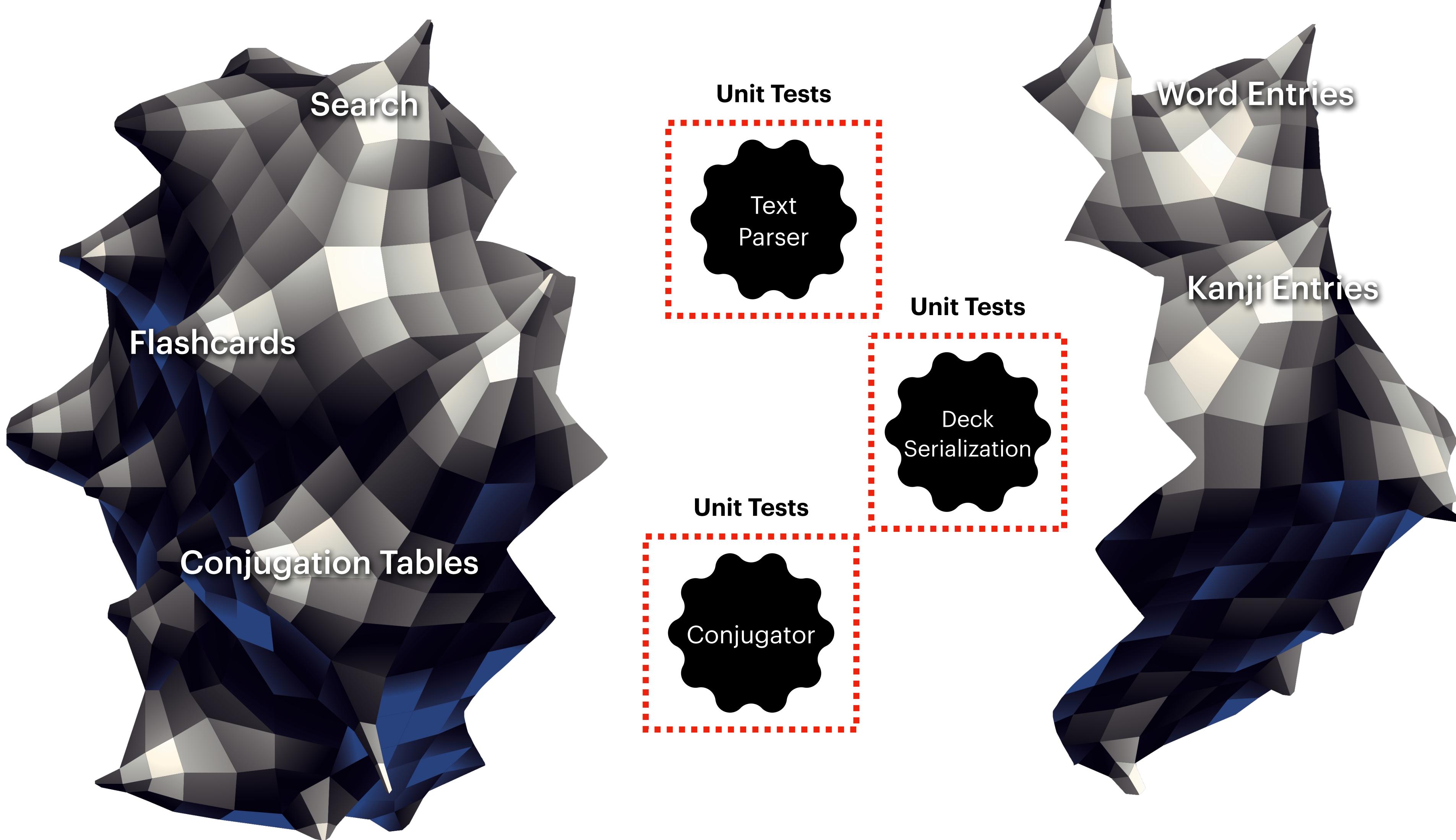
What about unit tests?



What about unit tests?



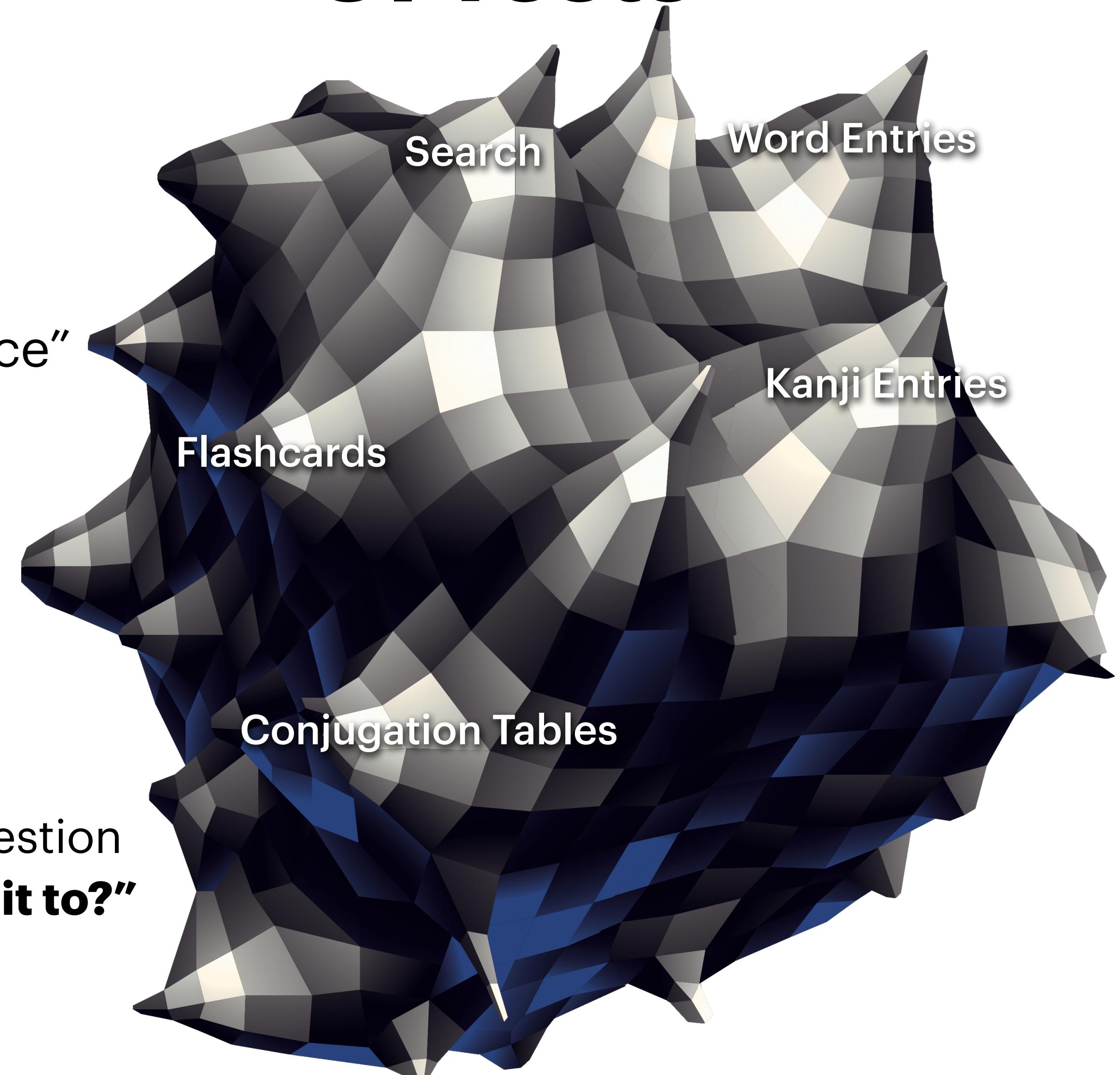
What about unit tests?



UI Tests

App Development

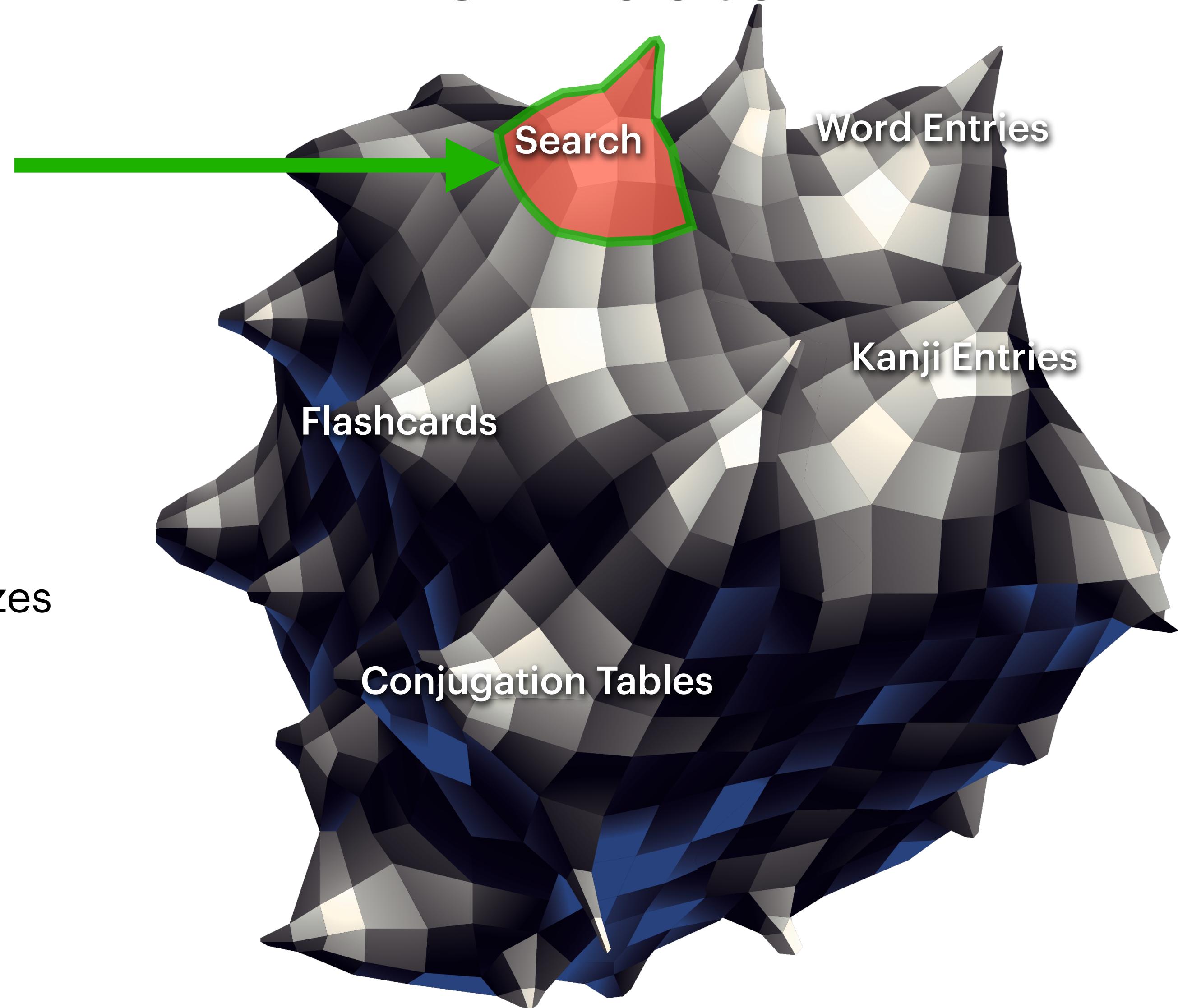
- More UI Bugs
- Disruptive OS Updates
- Code “closer to the surface”



UI Tests directly answer the question
“does my app work as I expect it to?”

UI Tests

Search Screen
in all languages
on all devices
in light/dark mode
on all iOS versions
with all dynamic type sizes



Automation (Jenkins)

- Device
 - OS Version
 - Dark/Light Mode
 - Language
 - Dynamic Type Size

Automation (Jenkins)

```
xcrun simctl create "My Simulator" "iPhone 8" "iOS15.4"
```

- **Device**
- **OS Version**
- Dark/Light Mode
- Language
- Dynamic Type Size

Automation (Jenkins)

```
udid=$(xcrun simctl create "My Simulator" "iPhone 8" "iOS15.4")
```

- **Device** `xcodebuild -destination id=$udid test ...`
 - **OS Version**
 - Dark/Light Mode
 - Language
 - Dynamic Type Size

Automation (Jenkins)

```
udid=$(xcrun simctl create "My Simulator" "iPhone 8" "iOS15.4")
```

- Device
- OS Version
- **Dark/Light Mode**
- Language
- Dynamic Type Size

Automation (Jenkins)

```
udid=$(xcrun simctl create "My Simulator" "iPhone 8" "iOS15.4")
```

- Device
- OS Version
- **Dark/Light Mode**
- Language
- Dynamic Type Size

```
/usr/libexec/PlistBuddy "...userInterfaceStyleMode.plist"  
-c "add UserInterfaceStyleMode integer 2"
```

Automation (Jenkins)

- Device
 - OS Version
 - Dark/Light Mode
 - **Language**
 - Dynamic Type Size
- /usr/libexec/PlistBuddy “....GlobalPreferences.plist” -c
“add AppleLanguages array” -c “add AppleLanguages:
string es”

Automation (Jenkins)

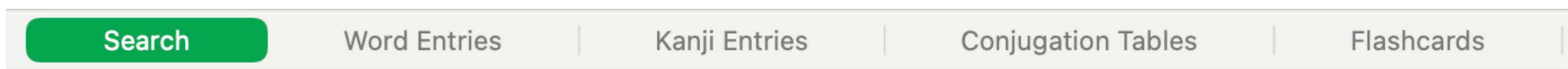
- Device
- OS Version
- Dark/Light Mode
- Language
- **Dynamic Type Size**

```
/usr/libexec/PlistBuddy "...com.appleUIKit.plist" -c -c  
"add UIPREFERREDCONTENTSIZECATEGORYNAME string  
UICTContentSizeCategoryXXXL"
```

Automation (Jenkins)

- Device
 - OS Version
 - Dark/Light Mode
 - Language
 - Dynamic Type Size

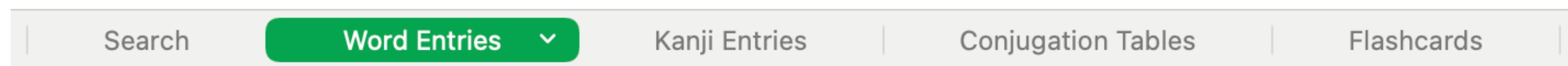
Test Plan



SEARCH

		Desired Behavior
	Basics	
	Search in Japanese	Matching words and kanji appear
	Search in English	Matching words appear
	Search again	Results are replaced when you hit return
	Transitions	
	Tap Kanji cell	Correct kanji entry appears
	Tap Word cell	Correct word entry appears

Test Plan



WORD ENTRIES

		Desired Behavior
	Basics	
	Tap Speaker Icon	Audio plays

Some tests
cannot be →
automated

Test Plan

Search	Word Entries	Kanji Entries	Conjugation Tables	Flashcards
WORD ENTRIES				
Basics				
Some tests must use screenshots →				
	Tap Speaker Icon		Audio plays	
	Localization		All strings localized and layout looks good	

Some tests
must use
screenshots →

Automation (Screenshot Test)

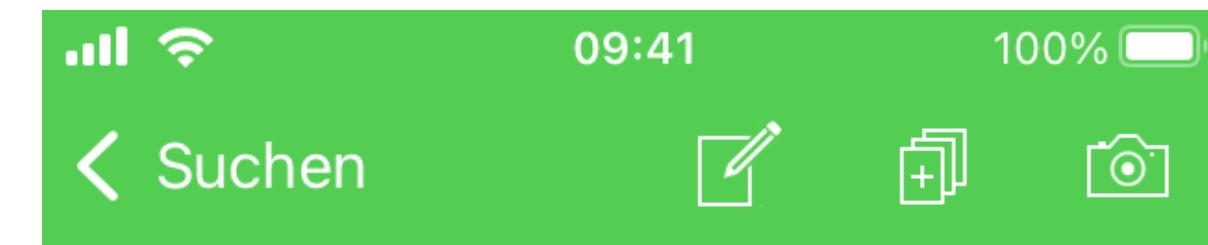
WORD ENTRY

	Localization	All strings localized and layout looks good
--	--------------	---

```
class WordEntryTests: XCUITest {  
    func testLocalization() {  
        navigateToWordEntry(for: "話す")  
        takeScreenshot(name: "Verify localization and layout")  
    }  
}
```

Output:

WordEntryTests - testLocalization - Verify localization and layout.png



はな
話す (audio icon)

GELÄUFIG

Transitives Verb

Konjugationen >

ALTERNATIVEN

咄す

BEDEUTUNG

sagen, sprechen, reden

erzählen, äußern

sich unterhalten, plaudern

beschreiben, schildern, berichten



WordEntryTests - testLocalization - Verify localization and layout.png

Screenshot Tests

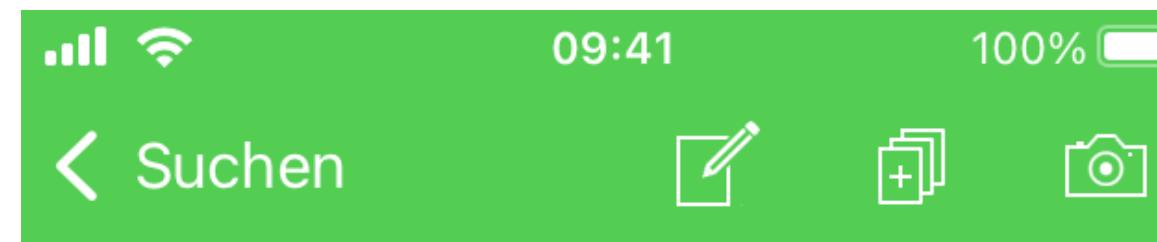
```
func takeScreenshot(_ description: String) {  
    ...  
    try! XCUIScreen.main.screenshot()  
        .image.pngData()!.write(to: outputURL)  
}
```

1. Make sure it looks good
2. Check it into git



WordEntryTests - testLocalization - Verify localization and layout.png

Next time - verify any changes are expected



はな
話す

Transitives Verb Konjugationen >

ALTERNATIVEN

咄す

BEDEUTUNG

sagen, sprechen, reden

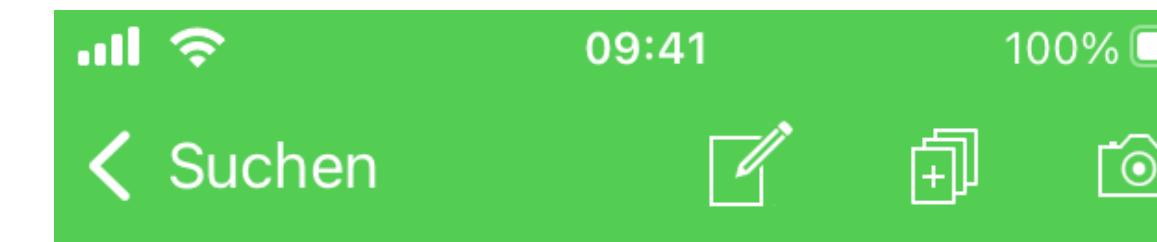
erzählen, äußern

sich unterhalten, plaudern

beschreiben, schildern, berichten



Old (from git)



はな
話す

Transitives Verb Konjugationen >

ALTERNATIVEN

咄す

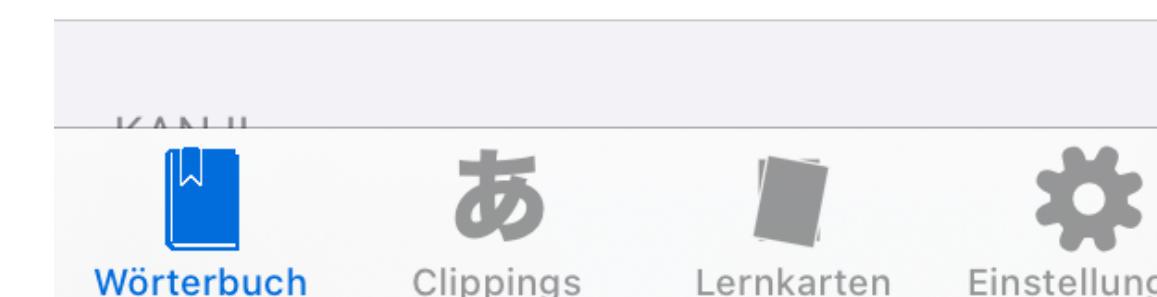
BEDEUTUNG

sagen, sprechen, reden

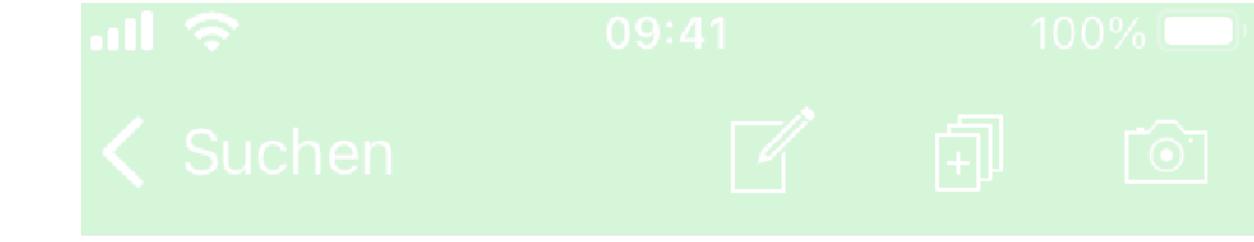
erzählen, äußern

sich unterhalten, plaudern

beschreiben, schildern, berichten



New



はな
話す

Transitives Verb Konjugationen >

ALTERNATIVEN

咄す

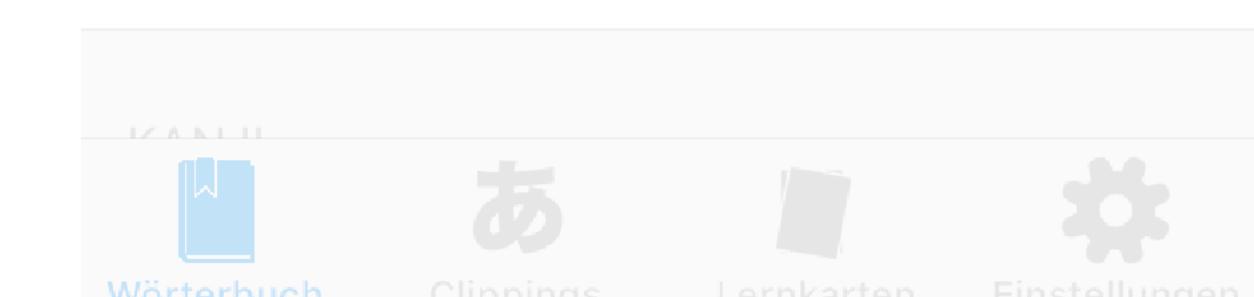
BEDEUTUNG

sagen, sprechen, reden

erzählen, äußern

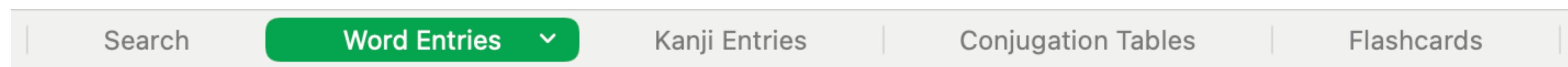
sich unterhalten, plaudern

beschreiben, schildern, berichten



Diff

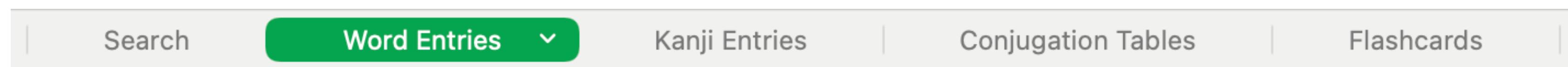
Test Plan



WORD ENTRIES

		Desired Behavior
Basics		
	Tap Speaker Icon	Audio plays
	Localization	All strings localized and layout looks good

Test Plan



WORD ENTRIES

		Desired Behavior
	Basics	
	Tap Speaker Icon	Audio plays
	Localization	All strings localized and layout looks good

How do I manage complexity

...and feel confident in each app release?

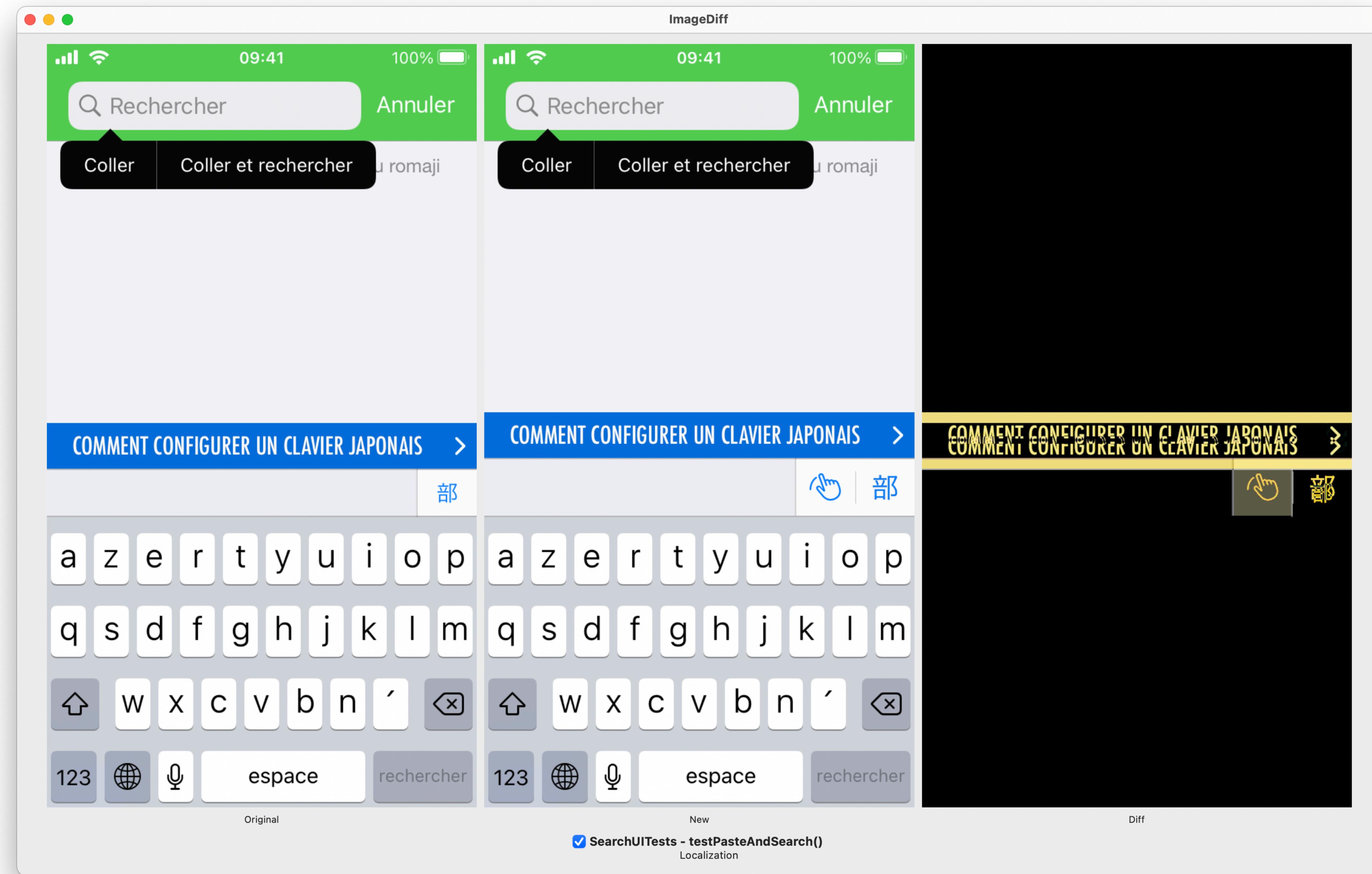
- Unit tests for internal components
- UI tests for additive features
- Screenshot tests when UI tests can't verify functionality on their own.
- Run across a variety of configurations via CI
- Test plan to verify entire app has been covered.

Thank You!

Extra Slides

Checking if Image Has Changed

- Shrink to max 6 pixels in one dimension:
 - `sips -Z 6 fullsize.png --out shrunk.png`
- Compare files:
 - `cmp --silent old-shrunk.png new-shrunk.png`



Custom Tools

Improving XCTest

- XCTAssertEqualSoon
- Massively parallel test runners
- Retrying on failures

Runtime Errors

- Firebase/Crashlytics
- expectNotNil