AVSpeechSynthesis
Making iOS talk
Session 236

Chris Fleizach, iOS Accessibility
Agenda

What and why
AVSpeechSynthesis basics
Choosing the right voice
Rate, pitch, and volume
Attributed strings
AVSpeechSynthesis

API for computer synthesized speech

Uses

• Announcements
• Non-sighted interfaces
• Education apps
• Many more...

Example—Audio updates during a workout
AVSpeechSynthesis and Accessibility

Powerful tool for helping many users

• Cognitive
• Speech vocalization
• Non-sighted

Not a replacement for VoiceOver

• Speech can overlap with VoiceOver's
• Won't be available to Braille devices
• Make your app accessible instead using UIAccessibility
Basics

Create AVSpeechSynthesizer

Ensure it's retained until speech is done
• (Speech will be cancelled in the synthesizer is deallocated)

```swift
let synthesizer = AVSpeechSynthesizer()
```
Basics

Create an utterance

Dispatch to synthesizer

```swift
let utterance = AVSpeechUtterance(string: "Hello")
synthesizer.speak(utterance)
```
AVSpeechSynthesis and Audio Sessions

AVAudioSession automatically activated on speak()

To mix with other audio, use

```swift
AVAudioSession.sharedInstance().setCategory(AVAudioSessionCategoryPlayback, with: .mixWithOthers)
```

To duck other audio, use

```swift
AVAudioSession.sharedInstance().setCategory(AVAudioSessionCategoryPlayback, with: .duckOthers)
```
Callbacks

Delegate methods inform about the life cycle of an utterance

AVSpeechSynthesizerDelegate defines optional methods

• Speech started
• Speech finished
• Character range will be spoken
• Speech paused
• Speech continued
synthesizer.delegate = self

func speechSynthesizer(_ synthesizer: AVSpeechSynthesizer, didStart utterance: AVSpeechUtterance)
{
    print("Speech started")
}

func speechSynthesizer(_ synthesizer: AVSpeechSynthesizer, didFinish utterance: AVSpeechUtterance) {
    print("Speech finished")
}

func speechSynthesizer(_ synthesizer: AVSpeechSynthesizer, willSpeakRangeOfSpeechString characterRange: NSRange, utterance: AVSpeechUtterance) {
    guard let rangeInString = Range(characterRange, in: utterance.speechString()) else { return }
    print("Will speak: \
    \(utterance.speechString_stringCharAt(rangeInString))")
}
Demo
Speak
Choosing the Right Voice

Built in voices (one for each supported language)

Siri voices not available through API

Users can download higher quality voices
Choosing the Right Voice

Select with an identifier or a language

Selecting by language will select the users default voice

```swift
let utterance = AVSpeechUtterance(string: "Hello")
// Select an English (US) voice preferred by user (if no preference, default is used)
utterance.voice = AVSpeechSynthesisVoice(language: "en-US")

// Select the first voice
let allVoices = AVSpeechSynthesisVoice.speechVoices() 
utterance.voice = AVSpeechSynthesisVoice(identifier: allVoices[0].identifier)
```
Choosing the Right Voice

Languages supported

<table>
<thead>
<tr>
<th>Arabic</th>
<th>Cantonese (Hong Kong)</th>
<th>Czech</th>
<th>Danish</th>
<th>Dutch</th>
<th>English (US)</th>
</tr>
</thead>
<tbody>
<tr>
<td>English (UK)</td>
<td>English (Australia)</td>
<td>English (Ireland)</td>
<td>English (South Africa)</td>
<td>Finnish</td>
<td>Flemish (Belgium)</td>
</tr>
<tr>
<td>French (France)</td>
<td>French (Canada)</td>
<td>German</td>
<td>Greek</td>
<td>Hebrew</td>
<td>Hindi</td>
</tr>
<tr>
<td>Hungarian</td>
<td>Indonesian</td>
<td>Italian</td>
<td>Japanese</td>
<td>Korean</td>
<td>Mandarin (Mainland China)</td>
</tr>
<tr>
<td>Mandarin (Taiwan)</td>
<td>Norwegian</td>
<td>Polish</td>
<td>Portuguese</td>
<td>Portuguese (Brazil)</td>
<td>Romanian</td>
</tr>
<tr>
<td>Russian</td>
<td>Slovak</td>
<td>Spanish (Mexico)</td>
<td>Spanish (Spain)</td>
<td>Swedish</td>
<td>Thai</td>
</tr>
<tr>
<td>Turkish</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Speech Rate

Controls the rate of speech [0-1]

- Speaking Rate is scaled from ≈0x -> 1x with values [0, .5]
- Speaking Rate is scaled from 1x -> 4x with values [.5, 1]

```swift
let utterance = AVSpeechUtterance(string: "Hello")
utterance.rate = 0.75 // faster
utterance.rate = 0.25 // slower
utterance.rate = AVSpeechUtteranceDefaultSpeechRate
utterance.rate = AVSpeechUtteranceMaximumSpeechRate
```
Pitch and Volume

Set properties on AVSpeechUtterance

- Pitch [0-1]
- Volume [0-1]

```swift
let utterance = AVSpeechUtterance(string: "Hello")
utterance.pitch = 1 // high pitched voice
utterance.volume = 0.25 // lower speech volume, does not affect system volume
```
Attributed Strings

Customize behavior of generated speech with attributes

IPA Notation

• International Phonetic Alphabet
• Uses—Specialized names, proper nouns, business names...
• Example—iPhone = ˈaɪˌfoʊn
IPA Notation

How do you generate IPA notation?

Settings>General>Accessibility>Speech>Pronunciations
Phrase
iPhone

Substitution

Dictate or spell out how you want the phrase to be pronounced.

Languages
English (US)
Phrase
iPhone

Substitution

Dictate or spell out how you want the phrase to be pronounced.

Languages
English (US)
IPA Notation

Use attributed string API with IPA pronunciation

```swift
let attributedString = NSMutableAttributedString(string: "Hello iPhone")
attributedString.addAttribute(.accessibilitySpeechIPANotation, value: "ˈɑɪ.ˈfʊn", range: NSRange(location: 6, length: 6)))

let utterance = AVSpeechUtterance(attributedString: attributedString)
```
Summary

Augment your app experience by adding speech at the right time

Multiple languages and voices are available

Customize pronunciation with IPA notation
More Information

https://developer.apple.com/wwdc18/236