KEN TORIMARU

310.994.0379
ken@torimaru.com

SKILLS

Swift • Objective-C • iOS • tvOS • watchOS • Cocoa Touch • Xcode • Interface Builder • Storyboards • IBDesignable Objects • REST • JSON • Git • SQLite • MySQL • Crashlytics • Test Flight • Alamofire • Eureka • CocoaPods • Java • Go • AWS Serverless • AWS S3 • AWS Rekognition • Dynamo DB

EXPERIENCE

BCGDV, July 2018 - April 2019
Contributed to the Drift car sharing venture. Helped develop an iOS app connected to a Node and Mongo driven backend. Part of the team that created the on device API and UX implementation. UX includes Eureka forms and custom IBDesignable objects. Contributed to a Twillio based custom messaging system.

Looks Like, September 2017 - on hold
iOS app connected to a AWS Serverless backend utilizing Dynamo DB and the AWS Rekognition object recognition services. iOS app is being written in Swift and the backend API in Go.

Anthem Media, July 2017 - September 2017
Created the RMXR (remixer) app, a video editing iMessage app. Utilized AVFoundation/AVPlayer to build an app that can edit downloaded web content, local video and audio content into a custom video.

ZiipTransit, June 2017
Add features to an existing app.

GE Power, October 2016 - April 2017
Part of the APM Mobile team building a native iOS app. Responsible for the implementation of app state restoration, multiple features and UI changes. Participated in the planning for a Core Data implementation, daily stand-ups and other agile activities.

Developed a working prototype iPad app for monitoring turbine performance in real time. The app itself was a UDP client which graphed incoming data using Quartz 2D. Built UDP server to simulate a data feed. Later converted the app to utilize an available http feed.

watchOS prototype for monitoring a power plant. Responsible for all development based on an evolving design. Used Quartz 2D graphics extensively. Developed in Swift.

Fully functional tvOS prototype for monitoring a power plant. Responsible for all development including working with the designer. Utilized MapKit using custom tiles and annotations. Utilized URLSession objects for login system and data procurement. Imported and utilized SJCL encryption library via JSContext object.

Construction Data Systems, July 2016
Upgraded the app's interface utilizing Storyboards and Apple's Human Interface Guidelines. iPad support was added along with an upload monitoring feature. The NSSession object and Swifty JSON was used for data communication. Fixed a photo issue for non-EXIF compliant browsers utilizing Core Graphics. Multiple bug fixes. Development was in Objective C.

HopSkipDrive, March-May 2016
Stabilized and added features to existing Swift codebase. The NSSession object and Swifty JSON was used for data communication. Managed two mission critical releases. Improved workflow and code management practices.

torimaru.com, 2005-Present
iOS development from the opening of the App Store. Experience includes use of the following frameworks: AddressBook, AudioToolbox, Contacts, CoreData, CoreGraphics, CoreLocation, Foundation, GameKit, iAd, libsqlite3, MapKit, MediaPlayer, MessageUI, QuartzCore, Social, SpriteKit, Twitter, UIKit, WatchConnectivity, and WatchKit.

Apps include AddingMachine, I'm Here, QuickLight, one touch dialers, Sale Price %, Icon Dial, Caw Of Doody (with Unity), Balloon Bounce, Balloon Pop, Hashtag Notes, and Jacks or Better Video Poker. Hashtag Notes and Jacks or Better Video Poker included an Apple Watch interface. There is an Apple TV version of Video Poker app. Both Video Poker games were written in Swift.

Director of Mobile Technology, SmallPlanet, Inc., 2004-2006
SmallPlanet was an early social networking company which attempted to differentiate itself by entering the soon to emerge mobile space.

Developed the friend finder software which enabled cell phones running the software to locate other proximate phones and then notify the cell's owner with an abbreviated profile. The software was based on J2ME and Bluetooth.

Developed a Google Map-based social networking website which the Google Latitude system would eventually resemble.

Independent Software Developer, 2001-2004
Developed casual gaming software in J2ME for the Symbian operating system. Software was distributed through Handango and the Nokia Software Market.

Technology Director, ClickZ, Inc., 1999-2001
ClickZ was a publisher of internet marketing newsletters utilizing a CGI-based content management and delivery system which was not scaling fast enough to accommodate the rapid growth. Management planned an expansion into conferences and physical print publishing. ClickZ was sold to Internet.com and new management planned a four-fold expansion.

Transitioned the content management system to java servlet technology. Created an e-commerce system for conference registration and payment, Developed a sales systems for book publishing. Updated the email newsletter publishing system. Recruited and managed a programming team to facilitate expansion. Integrated existing systems with the new parent company systems.

Independent Software Developer, 1996-1999
Developed websites and commerce systems on a contract basis. Specialized in dynamic data driven applications.

SurfSpot.com
Entrepreneurial web venture to provide real time images of surf conditions. Developed a robotic camera system to capture images, create time-lapse video, and upload data to a web server. Developed website for distribution of the image data and built a searchable database of surf related companies.

EDUCATION

Machine Learning with TensorFlow on Google Cloud Platform, a 5-course specialization by Google Cloud on Coursera. Specialization Certificate earned on May 8, 2019
Stanford University Coursera, Machine Learning Certificate, January 2016
California State University Northridge, B.S. Business Administration

OTHER INTERESTS AND PROJECTS

Electronic Hardware Development and Arduino Coding
Robotic camera system: Arduino controlled pan tilt head utilized on camera and remote GPS data to maintain in frame subject orientation. Communication utilized Zigbee radios.

Wifi sprinkler control system with iOS client control program: ESP8266 utilized for wifi communication, time updates and control of the relays.

WiFi Barbecue Temperature Gauge: ESP8266 utilized for wifi communication and processing of thermocouple data. Communicated with an iOS app over the local network.

Open Source Contributions
Contributed to the SwiftyPickerPopover open source project.
https://github.com/ktorimaru/SwiftyPickerPopover