

Ryan Gonzales

Software Developer | Computer Engineer

+1-310-913-0243

rjgonza1@uci.edu

/in/ryan-josh-gonzales

www.ryanjoshgonzales.com

EDUCATION

University of California, Irvine

B.S. Computer Science and Engineering
Expected June 2018
GPA: 3.41

SKILLS

Languages

Java, C++, Python, C, System Verilog, Mathematica/MATLAB

Design/IDE Tools

Visual Studio, Eclipse, Cadence/Mentor Graphics

Office Tools

Microsoft Word, PowerPoint, Excel, SQL Server Management Studio, Team Foundation Server

COURSEWORK

Computer Science

Data Structure Implementation
Analysis of Algorithms
Principles of Operating Systems
Compilers & Interpreters
Computer Networks
Computer Vision
Machine Learning

Computer Engineering

Org. of Digital Computers
Digital Signal Processing
Embedded Systems Software
Electronic Devices & Circuits

HONORS

Dean's Honor List

March 2016 – March 2017,
September 2017 – Present

PROFESSIONAL EXPERIENCE

TechnipFMC, UCOS Products Group | Software Dev Intern

June 2017 – Present | Irvine, CA

- Collaborated with a team of engineers to develop and assure the quality of the next generation of HMI development software
- Experienced with working with C++ GUI projects and libraries, like MFC, to create powerful Windows applications
- Developed and designed the HMI and control logic for many control system applications, from subsea to surface-level systems

PROJECTS

Multi-cycle ARM Processor

January 2017 – March 2017

- Worked with a team of engineers to design and simulate a pipelined ARM processor
- Verified block designs by writing test cases and viewing the waveforms in Mentor Graphics
- Synthesized the design to measure power and slack, and made improvements accordingly

“CRUX” Language Compiler and Interpreter

January 2017 – March 2017

- Built the scanner, parser, and symbol table according to fictional CRUX language specifications
- Created an abstract syntax tree (AST) to represent code structure, effectively taking care of problems like scope and type checking
- Generated MIPS assembly code by traversing the AST to build a text file of commands to run on the MIPS simulator, SPIM

Image Demosaicing

January 2018 – January 2018

- Used linear and bilinear interpolation on pixels in a 16-bit PGM image to reconstruct it into a full color image
- Computed missing values using array indexing instead of for loops to minimized the complexity of the program

Gesture Controlled Universal Remote

In Progress

- Employed Kinect SDK libraries to get skeleton data points, which are used to track movement across the sensor's plane
- Applied image processing libraries like OpenCV and EmguCV to classify our gesture data and give more accurate predictions
- Created a simple infrared circuit to work like a universal remote, in order to send command signals from the Kinect to the TV