



CASE WESTERN RESERVE
UNIVERSITY EST. 1826

think beyond the possibleSM

Launch Control System Software Development System Automated Testing

Student Name: Andrew Hwang
Academic Level: Junior
Academic Major: Computer Science
Academic Institution: Case Western Reserve University

Mentor Name: Jamie Szafran
Mentor Job Title: Computer Engineer
Org Code/Branch: NE-XS/Software Branch
Division: Electrical Engineering
Directorate: Engineering

Abstract

The Spaceport Command and Control System (SCCS) is the National Aeronautics and Space Administration's (NASA) launch control system for the Orion capsule and Space Launch System, the next generation manned rocket currently in development. This system requires high quality testing that will measure and test the capabilities of the system. For the past two years, the Exploration and Operations Division at Kennedy Space Center (KSC) has assigned a group including interns and full-time engineers to develop automated tests to save the project time and money.

The team worked on automating the testing process for the SCCS GUI that would use streamed simulated data from the testing servers to produce data, plots, statuses, etc. to the GUI. The software used to develop automated tests included an automated testing framework and an automation library. The automated testing framework has a tabular-style syntax, which means the functionality of a line of code must have the appropriate number of tabs for the line to function as intended. The header section contains either paths to custom resources or the names of libraries being used. The automation library contains functionality to automate anything that appears on a desired screen with the use of image recognition software to detect and control GUI components. The data section contains any data values strictly created for the current testing file. The body section holds the tests that are being run. The function section can include any number of functions that may be used by the current testing file or any other file that resources it. The resources and body section are required for all test files; the data and function sections can be left empty if the data values and functions being used are from a resourced library or another file.

To help equip the automation team with better tools, the Project Lead of the Automated Testing Team, Jason Kapusta, assigned the task to install and train an optical character recognition (OCR) tool to Brandon Echols, a fellow intern, and I. The purpose of the OCR tool is to analyze an image and find the coordinates of any group of text. Some issues that arose while installing the OCR tool included the absence of certain libraries needed to train the tool and an outdated software version. We eventually resolved the issues and successfully installed the OCR tool. Training the tool required many images and different fonts and sizes, but in the end the tool learned to accurately decipher the text in the images and their coordinates. The OCR tool produced a file that contained significant metadata for each section of text, but only the text and coordinates of the text was required for our purpose. The team made a script to parse the information we wanted from the

OCR file to a different file that would be used by automation functions within the automated framework.

Since a majority of development and testing for the automated test cases for the GUI in question has been done using live simulated data on the workstations at the Launch Control Center (LCC), a large amount of progress has been made. As of this writing, about 60% of all of automated testing has been implemented. Additionally, the OCR tool will help make our automated tests more robust due to the tool's text recognition being highly scalable to different text fonts and text sizes. Soon we will have the whole test system automated, allowing for more full-time engineers working on development projects.

Acknowledgements

I have a many great people to thank for all of the accomplishments I was able to achieve throughout this internship. I would like to first thank Jamie Szafran for continuously offering her most gracious generosity, assistance, and time, even if the help needed was minor assistance for simple project concerns or asking about programming advice and assistance. Next, I would like to thank Oscar Brooks and Caylyne Shelton for always thinking of the interns' safety and interests first. Also, I would like to thank Jason Kapusta for always being patient with the interns and answering every question that would be thrown at him with a smile. Lastly, I would like to thank all of the automated testing interns and full-time engineers: Brandon Echols, Caidi Phillips, Joseph Hastings, Jacob Moore, Shelby Pace, Mark Rodriguez, and Susan Pemble, for all working so well together as a team with great communication, patience, and support.