Christian Nassif-Haynes | SWE

Sydney – Australia

📱 0414 95 33 90   •   ✉ christian@nassif-haynes.com   •   🌐 nassif-haynes.com
🄾 DoxasticFox

Experience

FAIMS Project

Software Developer   *2015–2018 & 2020–Now*

Built Android-based information-gathering tools for archaeologists which won several grants, including $400,000 from the NSW Department of Industry:

- Implemented software used by Harvard and CSIRO researchers.
- Achieved a 3-fold reduction in handwritten lines of code by automating development.
- Created a stress testing framework and software profiler for our libraries, letting us measure and optimise their performance.
- Yielded a 30% reduction in our programs' startup times.
- Used: Python, BeanShell (i.e. Java with type inference and an interpreter), Bash, SQLite + SpatiaLite, GNU/Linux, `git`, `vim`.

Commonwealth Bank of Australia

Systems Engineer   *2018–2020*

Uplifted CBA's release engineering practices:

- Maintained and developed a large (>100,000 LOC) suite of static- and dynamic-analysis tools, allowing bugs to be caught early.
- Convinced my team to use unit tests and linters in our CI/CD pipeline.
- Implemented a web app, allowing us to securely collect and store metrics about release pipelines.
- Dockerised a suite of Apache tools (e.g. Hive, Hadoop, Spark) allowing other developers to test code locally, subsequently eliminating the associated several-day deployment time previously required for testing.
- Nominated as a CBA "Legend of Can" and won "due to exemplary performance and productivity".
- Used: Python, Scala, Docker, Bash, macOS, GNU/Linux, `git`, `make`, `vim`, Jenkins, TeamCity, Artifactory, Ansible, Kibana, REST, Flask, NumPy.

Projects

res-rnn: Achieved a new state-of-the-art on a machine learning problem (98.45% accuracy on pMNIST, an image classification task considered hard because pixels are shuffled before classification). Used: PyTorch.

google-research/batch-ppo & blei-lab/edward: Contributed bugfixes and documentation to Google researchers' machine learning repositories. Used: TensorFlow.

a-calculator.com: Amassed 3,000,000 page views and $16,000 AUD by implementing specialised calculators and explaining how to do the math by hand. Used: JavaScript, jQuery, MathJax, HTML, CSS.

haskell-train-carriage-game: Used memoisation and parallelisation in a purely functional language to solve the train carriage game. Used: Haskell.

odd-one-out-game: Created an artificially intelligent which uses natural language processing to play the common "name the odd one out" game. Used: Natural Language Toolkit (NLTK).

Education

Macquarie University   **Sydney**

Bachelor of Engineering (Software Engineering)   *2018*

Certifications

Amazon Web Services (AWS)

AWS Certified Cloud Practitioner   *Valid until: 21 September 2023*

Validation Number: DLZCF24BKBF1113J