Boyuan_Wang

sky.stevenwang@gmail.com • (647) 741 – 6417 • Toronto, ON, Canada
https://www.linkedin.com/in/boyuan-wang-steven/ • https://github.com/b232wang/

EDUCATION

University of Waterloo Waterloo, ON, Canada
HONOURS MATHEMATICS (Computer Science Major) 09/2014 – 01/2019

EXPERIENCE

ALLO. Inc 10/2020 - ~
Tech Lead Toronto, On, Canada

- Scheduling tools ensure the prioritized tasks that assigned to team members were monitored and processed as planned.
- Used **AWS_API_GATEWAY** to build an interface can be accessed by applications on any platform.
- Used Amazon **DynamoDB** to store all user's information.
- Create a demo of a social app that allows user to share text picture and short video.

Ventureum. Inc 05/2018 – 01/2019
Solidity Engineer (Blockchain project full-stack website system) Toronto, BC, Canada

- Developed an On-Chain intermediate system that has crypto-crowdfunding protocol to perform self-regulation for blockchain projects in a decentralized approach.
- Developed an open source smart contract (CarbonVoteX) which can conduct vote in a secure fashion with the feature that the voting conducted did not require coins to leave voters' wallets
- Used **solidity** with **truffle** to develop several contracts for Ventureum System that can be used to evaluate and audit individual virtual currencies.
- Used **AWS API-Gateway** and **E2** to develop the backend of system to store the main-base contract address, so that the contract is capable of running all designed fundamental functions within system.
- Used **React** to develop a front-end GUI to clearly show every project's milestone and timeline to let users to bit or refund their tokens.
- Main contributor of company's major project completed 1/3 coding in 16 days ahead of scheduled timeline.

Amazon. Inc 06/2017 – 09/2017
Software Developer Engineer Intern (Simulation Data Generator) Vancouver, BC, Canada

Built a datatype(language) with json-like format that allows user to customize multi-dimensional simulation data.

- Used **AWS-EMR distributed computing** to efficiently generate big mount of customized data in batches via **MapReduce**, stored in **AWS-S3**, the simulation data is used for testing an internal system.
- Used **Lexical analysis** and **Syntactic analysis** to define a json-like datatype which allowed user to create any number of objects with random length of any known datatype, as well as to set constants when user wants a fixed data.
- A trigger was developed using AWS_API_GATEWAY, where users can flexibly call and adjust the parameters of the system.
- Optimized the performance of the generator and improved the unit testing efficiency by 30%.

WLP4 Compiler (with Assembler) 01/2016 – 04/2016

- Developed a compiler (and also an Assembler) for WLP4(waterloo language plus points plus procedures) which is a programming language contains a strict subset of the features of C++.
- Used **Lexical analysis** to tokenize the input file (WLP4 code), then used **Syntactic analysis** and **Semantic analysis** to identify grammatical errors and semantic.
- For compiled code (Mips and object code) efferently, used **constant folding**, **common subexpression elimination**, **register elimination**, **dead-code elimination** and **strength reduction** to make compact code with memory space less occupied by 40%.

CC3K-IO (Online Nethack RPG game) 6/2019 – 10/2019

- CC3k-IO is a small MMORPG game like online version of Nethack, that player can choose race and finish each challenge in multiple dungeons. The game has Attack/Defense/Potion/Leveling/Classes/PVP modules. The RNG module will generate monster and treasure randomly.
- Used **decorator pattern** to implement player status, increased attack or defense and multiple buffs.
- Used **observer pattern** to split the logical core, controller and UI, that extend the capability of GUI development in long-time cycle.
- Used **visitor pattern** to implement the interaction between different classes and monsters.
- Used **Socket.IO** with **NodeJs** to implement **WebSocket** platform to enable multiple-player only mode in a synchronized map.

SKILLS

- **Language:** Java, Python, C/C++, Racket, Bash, Mips, MATLAB, SQL, Solidity
- **Web:** JavaScript, HTML, CSS, Node.js
- **Compiler:** Lexical analysis, Syntactic analysis, Semantic analysis, Code generation
- **Tools**: Git/GitHub, Bash Scripting, Homebrew, Apache Spark, AWS Tools (S3, Lambda, API Gateway, EMR), NPM, Truffle