RUIJIA(RITA) ZHANG

825-986-5052 | ruijia17@gmail.com | Linkedin Profile | Portfolio

A competent candidate with solid analytical and coding skills. Currently have hands-on experience in Full Stack Development and Front End Development. Seek a Full-Time SDE Position.

SKILLS

- Solid knowledge of JavaScript, ES6, CSS, HTML5, Java, C++, data structures, and algorithms.
- One year of web full stack development experience.
- Strong experience in web Front-end development using jQuery, AJAX, routing, React.js, Redux, Vue.js, and Vuex.
- Experience developing microservices with Spring Boot, Spring Security, Spring Cloud, JSON Web Token, WebSocket, and MySQL from scratch.
- Working knowledge of Git, GitHub, SSH, tmux, Vim, Postman, ECS, Docker, Maven, RESTful API, and Bootstrap.

PROJECTS

King of Bots (Vue.js, Vuex, Spring Boot, Spring Security, Spring Cloud, JWT, WebSocket, MySQL)

An online real-time battle platform also features game replay, rankings and more. Code your bot and let it fight.

- *Project Link* *Github Repo* *Project Demos*
- Implemented responsive design for eight front-end views using multiple-column layout, flexbox, grid, responsive images and typography.
- Designed superclass GameObject to reprint web pages by frame and unify Canvas animation refresh mechanism using requestAnimationFrame to avoid dropped frames for subclasses(Snake and GameMap).
- Used the flood-fill algorithm to ensure the connectivity of the adaptive scaling map with centrosymmetric randomly distributed obstacles to keep game fairness.
- Improved security and scalability of the web platform by using Spring Security to achieve permission control, JSON Web Token (JWT) instead of Session for authentication to avoid CORS, and WebSocket to realize asynchronous two-way communication.
- Realized MatchingPool(in MatchSystem microservice) following the readers-writers pattern and designed a matching algorithm based on players' rating and waiting time (simplified Elo rating system), shortening the player matching time.
- Implemented the BotRunningSystem microservice, put the user's bot code into the BotPool (designed based on the producer-consumer model) and manually realized a message queue to manage the code to be processed. The consumer dynamically compiles and executes the code in the queue by Joor API to achieve bot go forth.

Calculator (CSS, React Redux)

A calculator with frequent interactions between components, using Redux for state management.

- *Project Link* *Github Repo* *Project Demo*
- Used Grid Layout and Flexbox to implement button section and numeric display area, respectively, imitating Windows calculator.
- Designed four states and five actions that cover the regular operations of the calculator.
- Implement the digitButton and operationButton components, which abstract the functionality of the same type of buttons to achieve reuse.
- Used Redux to query state, change state, and propagate state changes in a single store to better manage the four states that are interrelated and frequently updated, avoiding multiple levels of components passing data and reducing re-renders.

King of Fighters (CSS, jQuery, JavaScript ES6)

A 2D horizontal fighting game. Two players share a keyboard to fight against each other.

- *Project Link* *Github Repo* *Project Demo*
- Build a game page with HP bars and countdown using flexbox for layout, float to adjust position, multiple type selectors and calc() for dynamic calculations.
- Constructed hierarchical structure classes related to each other by inheritance. GameMap and Player inherit from the base class, which renders all game objects. Different character classes inherit generic character actions, states and state transitions from the Player. This organized structure gives code consistency and easy management.
- Designed a finite-state machine, including a state collection, a set of state transitions and the current state variable, and by setting events listeners and character variables such as initial position, direction, speed and gravity etc., make characters have seven reasonable and smooth animations.
- Used Axis-Aligned Bounding Box for collision detection to achieve attack, attacked and death effects.
- Simplified development by using jQuery to filter/add/find elements, jQuery event methods, and jQuery animate() to effect the fading animation of HP bars after being attacked.

EDUCATION

University of Alberta, Edmonton, Canada *09/2020 - 04/2022*

Master of Engineering in Electrical and Computer Engineering, GPA: 3.9/4.0

University of Electronic Science and Technology of China (UESTC), China *09/2016 - 06/2020*

Bachelor of Engineering in Software Engineering, GPA: 3.63/4.0