Software Architecture Report

SIGSEGV

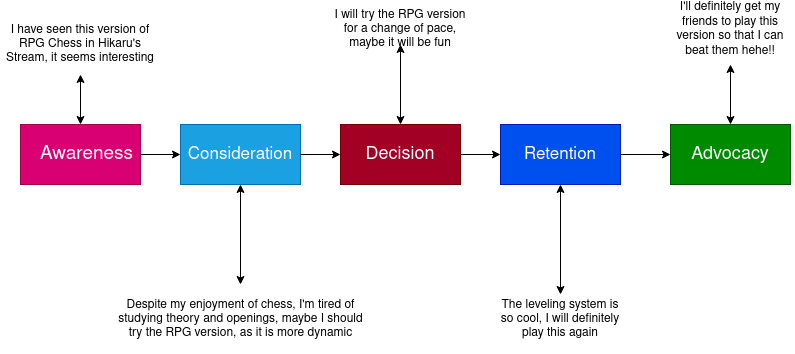
RPG CHESS

This document aims to describe to a potentially new member of the development team what does the software project do at the time of redaction. It doesn’t have to be a monolith document, the information can be stored in READ.MEs.

The intended audience for this document is a technical one. Different sections may target different stakeholders.   
  
The document must be:

* Dated: 29th of January 2024
* Versioned – First and final version

Contents:

1. What the purpose of the software project is
   1. This may be a summary based on the planning documentation.
      1. The project’s purpose is to implement an upgrade to the game of chess by giving it an RPG feeling, in the form of leveling up and abilities
   2. Fulfilled capabilities – what can the project do at this point, and what is to be done until project fulfillment.
      1. The project has been finalized successfully for a beta version launch and has all of the RPG functionalities intended
2. Guides on how to:
   1. Run the project locally.
   2. Build the project.
      1. 1. docker build -t sigsegv/chessrpg:1.0 .
      2. 2. docker run -p 3000:3000 stefantudose/chessrpg:1.0
   3. Deploy the project (either locally, or how is it hosted, where it is the case)
      1. Locally for development and testing
      2. With ngrok (at the moment) for multiplayer play.
   4. Contribution guide
      1. Patterns used in your application
         1. Singleton for database
3. Application entry points
   1. Data sources
   2. Data inputs
   3. Configuration files:
      1. There are 3 main configuration files:
         1. package.json: containing all of the NPM dependencies
         2. tsconfig.json: containing configuration instructions on how the Typescript compiler must run
         3. webpack.config.js: containing instruction for our web compiler - webpack
4. High level diagrams of the architecture
   1. User/data journeys
   2. Most valuable output
5. Deployment plan
   1. Where is the application deployed?
      1. The app is deployed in a docker container that builds it nicely and contains all of the necessary dependencies for fluid functionality
   2. How the CI/CD pipeline works.
      1. A custom image was built that contains the operating systems, the platform that was used (Node, Typescript, NPM) and all of the required modules.
6. Description of the QA process
   1. Test suites – what do they test. - test suites were not used in the project

1. External dependencies included in the project
   1. APIs used
      1. No APIs were used in this project
   2. Libraries:
   3. The project runs in Node version 21:06:00 and uses typescript.
   4. The following npm modules were used:
      1. dependencies": {
      2. "async-mutex": "^0.4.0",
      3. "typescript": "^5.0.4",
      4. "cookie-parser": "^1.4.6",
      5. "ejs": "^3.1.9",
      6. "express": "^4.18.2",
      7. "mongodb": "^5.5.0",
      8. "readline-sync": "^1.4.10",
      9. "socket.io": "^4.6.1",
      10. "uuid": "^9.0.0"
      11. },
      12. "devDependencies": {
      13. "@types/express": "^4.17.17",
      14. "@types/node": "^20.2.5",
      15. "@types/uuid": "^9.0.1",
      16. "webpack": "^5.85.0",
      17. "webpack-cli": "^5.1.1"
      18. }

DEMO: https://youtu.be/khzuol5ZFJc