

## Software Development Plan

### Statement of purpose and scope

The objective of this application is to create a small-scale, short arena-themed text-based fighting game in the Ruby programming language, that runs on the user's terminal.

The fighting in the game will utilise a simple combat logic pulled from the classic hand-based game 'Scissors-Paper-Rock'. The game will see the player choose a name for their fighter in the terminal before being thrown into the arena, within the code file, to compete against a number of foes with various pre-defined characteristics, including their names and their starting health points. The fighting itself will then be conducted through the player choice of one of multiple 'moves' that they can perform – the options of which will be displayed to the player before being able to be entered through the regular keyboard input. This player 'move' selection will then be logically compared against their opponent's choice of move using the aforementioned fighting logic, and health points will then be deducted appropriately from each party depending on the outcome of the comparison of each parties' move.

The intended target audience of this application will be people who possess a computer of some sort, a bit of free time to spare and a desire for some form of entertainment whilst also not being opposed to the idea of the fighting game genre, or alternatively, those wanting to sample a small code project built in the Ruby programming language and possibly check the project for areas that can be improved as a means of furthering their own comprehension of Ruby knowledge via improving on someone else's work at a beginner level.

The application will be able to be used by having the Ruby code file in a known directory, and then opening up that file via the users' terminal and passing in a name argument in order to boot up the game.

## Features

- 1) The user has a choice in username: The user within the arena terminal application is able to designate a username for their usage during their time within the game. This enables the user to feel a certain level of familiarity with their in-game character, who otherwise doesn't really have many identifying characteristics. There are also length constraints on this username to make sure that any usernames run with the program are of a reasonable length to repeatedly display. Users also have the option to check the username history log in the program's primary directory if they wish to take inspiration for their username choice from previous users of the application – whose chosen usernames are stored in this log in a neat format.
- 2) The user is able to engage in combat via a Scissors, Paper, Rock logic-influenced combat system: The combat system of this program combines the 'three conflicting elements' combat style of the traditional children's game Scissors, Paper, Rock, with a 'health points' mechanic pulled from role-playing videogames that represents the life total of both the player and their opponent (and that upon reaching zero – results in a game over state for that side). The combat also differs from traditional Scissors, Paper, Rock logic in that it is turn-based – with a set attacker and defender per turn, or phase of the match. Combat is carried out via user input directed by the prompts of the in-application text engine.
- 3) The user is able to enjoy the differing characteristics of characters appearing within the application: This application features, exclusive of the user themselves, four different characters appearing in sequence. These characters all have their own characteristics – encompassing a separate colour text for each characters' communications, distinct catchphrases/speaking styles, and differing health points (for combat-related characters). These differing characteristics are used to supplement the user experience by offering an element of diversity within in-game back-and-forth interactions.

## User interaction and experience outline

Before booting up the application-proper, the user is able to browse the contents of the applications repository; including a username history file documenting the usernames that all past users of the application on the same terminal have chosen for decision reference, and this software development plan informing the user the general purpose of and acting as a general how-to guide for the application, amongst other files. The user will begin using the application by calling it from their terminal with a single argument that is to be used as their username within the application. The application will then begin as users are greeted with their provided username by an announcer. The announcer will ask a couple of questions to the user – one of which the users response is meaningless, and the other of which requires an affirmative response from the user in order to proceed, as it explains the rules of the game in-depth. Then, the first battle opponent will emerge – and the user, now knowing the game rules, will commence combat with that opponent. The combat experience sees the user typing in one of their moves out of “scissors”, “paper” and “rock” in both their attack phase (attacking) and the opponent’s attack phase (defending). This is carried out sequentially until either the player or the opponent run out of health points. If the player runs out of health points before the opponent does, the screen will announce a game over state and exit the application. However, if the player manages to eliminate all of the opponent’s health before they lose all of theirs – the goal of the game, they are allowed to proceed to the next opponent. Following the first and second battle phases, the announcer will briefly return – announcing the full restoration of the users health points and giving a brief introduction of the next opponent. After the third opponent is beaten in combat, the announcer briefly congratulates the player for completing the game, and then the application exits. The application is designed to be and is inherently simple enough to be void of most errors, but a notable error is when the user fails to successfully enter a single, three to eight character username in their command line argument when booting the main file. This is handled by automatically assigning a default username to the player character for that game session – not saved in the username history file, but fine for a tentative playthrough of the application, and convenient for allowing the user not to have to deal with the application exiting and necessitating a relaunch.

## Control flow diagram

