Colosseum Duel

Requirements and Test Document

Adrian Quintero

December 2nd, 2022

CS 225, Fall 2022

Embry-Riddle Aeronautical University

Daytona Beach campus

1 Aerospace Boulevard

Daytona Beach, FL 32114

INTRODUCTION:

BACKGROUND INFORMATION:

Requirement 1 exists to ensure that the software shall have the ability to have classes which can be successfully extended from Weapons. It also exists to ensure that the player can willingly select a specific extended class. The primary methods being utilized in this requirement are weaponSelect(), makeSelection(), and printInfo(). Test Case 1 checks that software selects the first switch case when the user inputs a value of 1. Test Case 2 checks that the software catches an out-of-bound number and prompts the user to reinput a valid number. Finally, Test Case 3 checks that the software can determine that an input is non-numerical and prompts the user to reinput something numerical.

Requirement 3 exists to ensure that the software shall have the ability to read user input. It proves that the software can determine whether a specific value input by the user is out of the desired range, and as a result forces the user to re-enter all values until all the requirements are met. Methods primarily being utilized in this requirement are statAllocation(), inputHealth(), inputAttack(), inputSpeed(), and inputDefense(). Test Case 1 checks that the software accepts the outer bounds of multiple stats and proceeds as normal. Test Case 2 checks that the software can determine when the player combatant's BST is too high or too low, then prompting the player to reinput their values. Finally, Test Case 3 checks that the software can determine that an input is non-numerical and prompts the user to reinput something numerical. This test case and Test Case 2 of Requirement 1 show that the software can detect errors in input.

Requirement 4 exists to ensure that the software shall have the ability to have classes which can be successfully extended from Armor. Methods primarily being utilized in this requirement are armorSelect(), makeSelection(), and printInfo(). The test cases for Requirement 4 are identical to the ones conducted for Requirement 1.

Requirement 8 exists to ensure that the software shall keep track of the number of duels that have occurred since the software was (re)ran. Test Case 1 checks that the software can determine when the player's combatant has run out of HP, in turn printing out a "GAME OVER" statement. Test Case 2 checks for the opposite, being that the software can determine when the player's combatant successfully completes five duels without running out of HP. The software then proceeds with a congratulatory statement. In both test cases, the software allows the user to save their combatant data.

Requirement 10 exists to ensure that the software shall print out battle forecasts when applicable. Test Case 1 checks that the software understands when the player has completed

their combatant customization, printing out the battle forecast as a result. Test Case 2 checks that the software understands when a duel is complete with the player combatant still alive, printing out a battle forecast as a result. Test Case 3 checks that the software understands when the maximum number of duels has been reached, thus not printing out a battle forecast.

Requirement 12 exists to ensure that the software shall read and write information from and to a file, showcasing file I/O. Test case 1 ensures that the software shall determine when the file does not exist yet is being called, letting the user know that it does not exist and moving on with manual stat allocation. Test case 2 ensures that the software shall be able to read the data from the text file correctly. Test case 3 ensures that the software shall be able to write specific data to a text file at the request of a user.

REQUIREMENTS:

Below, Table 1 outlines the ideas and requirements that shall be implemented and achieved, respectively.

Table 1: Requirement Specifications

Reqt ID	Requirement Specification
As a developer, I want multiple types of weapons with varying behaviors to be available to choose from.	
1	The software shall allow the player to input a value that selects a weapon from a switch case when the player has finished selecting their combatant's stats.
As a developer, I want enemy combatants to be automatically generated with varying behaviors.	
2	The software shall auto-assign a BST to an auto-generated duelist that is satisfactory when called by the GameController class.
As a developer, I want stat allocation for the player's combatant to be determined by the player.	
3	The software shall allow the player to input five different stat values that are satisfactory when the player chooses not to load from a save file.
As a developer, I want gear for the player's combatant to be determined by the player.	
4	The software shall allow the player to input a value that selects an armor set from a switch case when the player has finished selecting their combatant's weapon.
As a developer, I want two combatants to complete a duel autonomously.	
5	The software shall have two combatants take turns dealing damage to one another when the game has determined which combatant initiates combat.
As a developer, I want combatant move order to be determined by specific stat calculations.	
6	The software shall determine whether the player or enemy combatant deals damage first by comparing multiple stats yet to be selected.

As a developer, I want combatants' attack accuracy to be based on a percentage calculated by stat allocation and gear.	
7	The software shall determine whether a combatant successfully deals damage based on their weapon's HIT, armor's WGT, and enemy's AVOID when it is the combatant's turn using Equation 1.
As a developer, I want the software to end/loop after a set number of duels won by the player in a row.	
8	The software shall run up to five duels when the player combatant does not get eliminated beforehand.
As a player, I want to be able to see how combat is progressing in the duel.	
9	The software shall print out dialogue detailing every event that occurs per duel when combatants begin taking turns.
As a player, I want to be able to see the enemy combatant's information after locking in my combatant.	
10	The software shall print out dialogue depicting a battle forecast before each duel when either player combatant customization is complete or proceeds to the next duel.
As a player, I want to be able to have control over when the next duel begins based on whether I can heal my combatant.	
11	The software shall allow the player to select whether to heal their combatant when they have finished a duel and if they are eligible.
As a player, I want to be able to save and load my combatant's stat allocation and gear loadout.	
12	The software shall allow the player to save their combatant's data to a file named "PlayerStats.txt" with the format shown in Figure 2 and load it when they restart of the software.

TEST CASES:

Table 2: Test Cases and Results

Req't ID	Test Case ID	Initial Conditions And Input	Expected Behavior Or Output	Actual Behavior Or Output	Pass Fail
1	1	Weapon selection = 1	Software selects Sword and asks for confirmation.	Software selects Sword and asks for confirmation.	Pass
1	2	Weapon selection = 0	Software detects out-of-bound value and prompts user to reinput value.	Software detects out-of-bound value and prompts user to reinput value.	Pass

1	3	Weapon selection = “test”	Software detects non-numerical value and prompts user to reinput value.	Software detects non-numerical value and prompts user to reinput value.	Pass
3	1	health = 50, attack = 40, speed = 15, defense = 20	Software proceeds as normal.	Software proceeds as normal.	Pass
3	2	health = 50, attack = 40, speed = 40, defense = 35	Software detects a BST value over 125 and prompts the user to reinput values.	Software detects a BST value over 125 and prompts the user to reinput values.	Pass
3	3	health = attack = speed = defense = “test”	Software detects non-numerical value and prompts user to reinput value.	Software detects non-numerical value and prompts user to reinput value.	Pass
4	1	Armor selection = 1	Software selects Leather Armor and asks for confirmation.	Software selects Leather Armor and asks for confirmation.	Pass
4	2	Armor selection = 0	Software detects out-of-bound value and prompts user to reinput value.	Software detects out-of-bound value and prompts user to reinput value.	Pass
4	3	Armor selection = “test”	Software detects non-numerical value and prompts user to reinput value.	Software detects non-numerical value and prompts user to reinput value.	Pass
6	1	Player speed > Enemy speed	Player combatant attacks first	Player combatant attacks first	Pass
6	2	Enemy speed > Player speed	Enemy combatant attacks first	Enemy combatant attacks first	Pass
6	3	Player speed = enemy speed	Enemy combatant attacks first	Enemy combatant attacks first	Pass
7	1	Player selected Sword and Leather Armor, enemy did not select bow	Total HIT = 90 (90HIT – 0 WGT – 0 AVOID)	Total HIT = 90	Pass

7	2	Player selected Sword and Rogue's Attire, enemy did not select bow	Total HIT = 80 (90HIT – 10 WGT – 0 AVOID)	Total HIT = 80	Pass
7	3	Player selected Sword and Soldier's Gear, enemy did select bow	Total HIT = 60 (90HIT – 20 WGT – 10 AVOID)	Total HIT = 60	Pass
8	1	Player combatant dies before completing all five duels	Software prints "GAME OVER" and prompts to save data.	Software prints "GAME OVER" and prompts to save data.	Pass
8	2	Player combatant is alive after completing all five duels	Software congratulates player and prompts to save data.	Software congratulates player and prompts to save data.	Pass
10	1	Player combatant customization is completed	Software prints out a battle forecast showing both combatants' information.	Software prints out a battle forecast showing both combatants' information.	Pass
10	2	Player combatant completes a duel	Software prints out another battle forecast showing both combatants' information (using next enemy combatant).	Software prints out another battle forecast showing both combatants' information (using next enemy combatant).	Pass
10	3	Player combatant completes its fifth duel	Software does not print another battle forecast.	Software does not print another battle forecast.	Pass
11	1	Player selects to heal their combatant	Combatant is healed.	Combatant is healed.	Pass
11	2	Player selects to not heal their combatant	Software proceeds to the next duel.	Software proceeds to the next duel.	Pass
11	3	All heals have been used by player	Software does not offer the option to heal.	Software does not offer the option to heal.	Pass
12	1	Player selects to load a save file when it has not been previously created	Software lets player know that the file does not exist and proceeds	Software lets player know that the file does not exist and proceeds as if the player selected no.	Pass

			as if the player selected no.		
12	2	Player selects to load a save file and it has been created	Software reads information from file.	Software reads information from file.	Pass
12	3	Player selects to create a save file	Software writes information to file.	Software writes information to file.	Pass

REFERENCES:

No references are needed for this document.

APPENDICES:

No appendices are needed for this document.