

Software Requirements Specification

FurReal Engine

Team GulDukat

Ethan Wright
Zachary Fenton
Rachel Koenig
Daniel Harding

Table of Contents

1. Introduction.....	2
1.1 Scope.....	2
1.2 Definitions, Acronyms, and Abbreviations.....	2
2. Interfaces.....	3
2.1 System Interfaces.....	3
2.2 User Interfaces.....	3
2.3 Constraints.....	3
3. Specific Requirements.....	4
3.1 Functional Requirements.....	4
3.2 Non-Functional Requirements.....	6

1. Introduction

The introduction section of the System Requirements Specification (SRS) provides information in regards to the programs scope and to list any definitions, acronyms, and abbreviations.

1.1 Scope

Our company has been contacted to create software for Cougar Gaming. Cougar gaming is a company that primarily focuses on simulating many gaming mechanics including the combat system. Our team plans to create a game engine, named {NAME}, to allow for users to have monsters and characters fight while we record combat stats (STR, INT, WIS, DEX, CON, HTH) and report many other useful results. Our program will also allow for PC's and NPC's to interact with each other.

1.2 Definitions, Acronyms, and Abbreviations

Term	Definition
User/Player	Refers to the user of the program
stats	Refers to combat statistics
STR	Refers to the character's attribute Strength
INT	Refers to the character's attribute Intelligence
WIS	Refers to the character's attribute Wisdom
DEX	Refers to the character's attribute Dexterity
CON	Refers to the character's attribute Constitution
HTH	Refers to the character's attribute Health
DL	Refers to a monster's dungeon level
PC	Refers to the term Playable Character
NPC	Refers to the term Non-Playable Character
D20	Refers to the action of a user rolling a 20-sided die
SRS	System Requirements Specification

2. Interfaces

The interfaces section of the System Requirements Specification (SRS) provides information in regards to system interfaces, user interfaces and constraints.

2.1 System Interfaces

Other systems that could potentially interface with this software are other games. Many other games could potentially have the ability to use our combat system, user modules and other functions in their game. Doing this will allow them to calculate many combat stats like health loss, damage done and treasure gained.

2.2 User Interfaces

There are many user interfaces that the software will possess. We plan to provide and update this section with UI mock-ups to give a visual of this.

We plan to provide many boxes to the primary scene, one of them being the attributes section with tons of useful stats visually displayed in and easily readable format. In addition to this, a treasure tracker to display the total amount of treasure gained by the character. Lastly, the scene itself, characters and monsters.

2.3 Constraints

There are not many constraints with this software. However, writing the software in C# would be considered one. Writing the software in C# may potentially limit us to using certain utilities, functions and libraries which may not be as helpful to us as other languages may be. Lastly, using C# may generate slightly different results than other languages would.

3. Specific Requirements

The specific requirements section of the System Requirements Specification (SRS) provides information in regards to functional and non-functional requirements. It does this with help from the requirements elicitation section.

3.1 Functional Requirements

REQ #	PRIORITY	REQUIREMENT	DEPENDENCIES
FR-01		Characters gain bonuses correlating to their profession with each increase in level	
FR-02		Specific environments can affect a PC or NPC/Monster's ability to fight	
FR-02A		Scene level, number of characters, and the character's profession levels are used to determine the CD	FR-02
FR-03		Treasure can only be accessed after all monsters are defeated	
FR-04		Success of all character and monster actions are determined randomly	
FR-02B		CD is determined by the sum of all monster DLs in current scene	FR-02
FR-05		Character initiative is modified by PC's intelligence	FR-09
FR-06		Monster initiative is modified by monster's DL	FR-09
FR-07		Sum of three D6 rolls is equal to the statistic score (i.e. a PC stat)	
FR-08		A D20 roll determines level of success	
FR-09		Character initiative at start of combat is determined randomly (can be modified)	
FR-10		PCs and Monsters with health less than or equal to zero are considered dead	
FR-07A		Characters have six types of statistics (STR, INT, WIS, DEX, CON, HTH)	
FR-11		Scene levels start at one and increases each scene	
FR-12		Monsters and NPCs choose attacking targets randomly	
FR-13		PCs are given a list of targets to attack	
FR-14		At start of simulation, tester is to choose a scene or let one be chosen randomly	

FR-15		Tester must specify max and starting scene level	
FR-16		Tester must specify number of characters and monsters for scene	
FR-17		Tester is allowed to let number of characters and monsters, either or both, be randomly selected	
FR-18		User is allowed option to select specific monsters for scene (if not random)	FR-17
FR-19		Tester must choose if a character is a PC or NPC	
FR-20		Tester may choose profession for character, or it is chosen randomly	
FR-21		NPC combat actions are simulation-controlled	
FR-22		For each PC, tester must select a valid option	
FR-23		Tester chooses for each character to heal "Each level" or "Upon Death"	
FR-24		Tester will specify how many times a scenario will be run of the same scene	
FR-25		Data values are reset at start of each new scenario	FR-34
FR-26		Users (testers/admin) run simulations and generate report functional	
FR-27		Administration (admin ONLY) can modify base data (insert, delete, etc) functional	
FR-28		Administration (admin ONLY) can create/delete accounts functional	
FR-29		Character stats modify (positively and or negatively) it's ability to fight functional	
FR-30		CD may be higher or lower depending on if user chose Novice, Apprentice, or Master functional	FR-02B
FR-31		As scene level increases, so do monster levels functional	FR-11
FR-32		Any remaining CD after tester selection is randomly met by putting random monsters on the scene (???) functional	
FR-33		A round ends when all monsters are defeated functional	
FR-34		Raw data of all values are collected for each scenario functional	
FR-35		Report is generated at end of software containing: means, standard deviations over entire population, and medians for each value functional	

3.2 Non-Functional Requirements

REQ #	PRIORITY	REQUIREMENT	DEPENDENCIES
NFR-04		Must be coded in C#	
NFR-05		Simulation results are generated via the "Monte Carlo method"	
NFR-13		Raw data values should be stored separately from the analysis values	