‘OSRS Clone’

CIS 1512

Software Engineering

Software Requirements Specification

Software Project Management & Planning

Professor Hadi Nasser

‘OSRS Clone’ Team:

Kyle

Malik

Mitchell

Laith

Table of Contents

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

1.0 Introduction

In our project, we will be applying agile methodology to the extent to which we can. We will have sprints that are focused around delivering at least one of our functional requirements. Each functional requirement can be thought of as a user story without the need to necessarily detail and write out that story. Each iteration of the project will be shown to other group members for feedback. Testing before delivering is done by consistently testing while coding. We can show the current iteration through demonstration videos or by even sharing a download for others to play so we can play-test our game. The next sprint will incorporate any feedback gathered and our requirements and specifications will be adjusted accordingly. In a limited sense, interacting with other group members can count as both customer interaction and team interaction. Even the professor could be shown sprints and function as a stakeholder or customer representative.

Goals and Objectives

The main objective of this project is to create a software that is a playable game reminiscent of Old School Runescape. It is our goal to create as successful a software as possible while working within the constraints of our group. The requirements we have laid out will allow the end-user to move around, interact, and do combat within a fantasy world setting. The end-user will be able to get at least some enjoyment out of experiencing the novelty of a small-scale creation like this game.

Scope of The Project

Since the programming will be done by one or two group members and we have limited time, we are trying to be realistic in our scope. The features outlined in the use case diagram and the functional requirements will be used as guidelines for keeping on pace with the limited timing of this class. As time and skill permit, we plan to 'flesh out' the game as much as we possibly can.

Functional Requirements

- The player will be able to interact with items and enemies by clicking on them.
- Implementation of a combat system that includes melee, ranged, and magic damage options.
- The player will be able to pick up items to alter their stats.

- The player will be able to level up their character by gaining experience points through gameplay.
- The player will move by clicking.
- The player will be able to interact with NPCs by reading their text dialogue.
- Implementation of a clear and concise GUI that notify the player of their health, mana, gold, etc...
- Implementation of a menu system to allow the player to customize options.
- An application to start the game.

Non-Functional Requirements

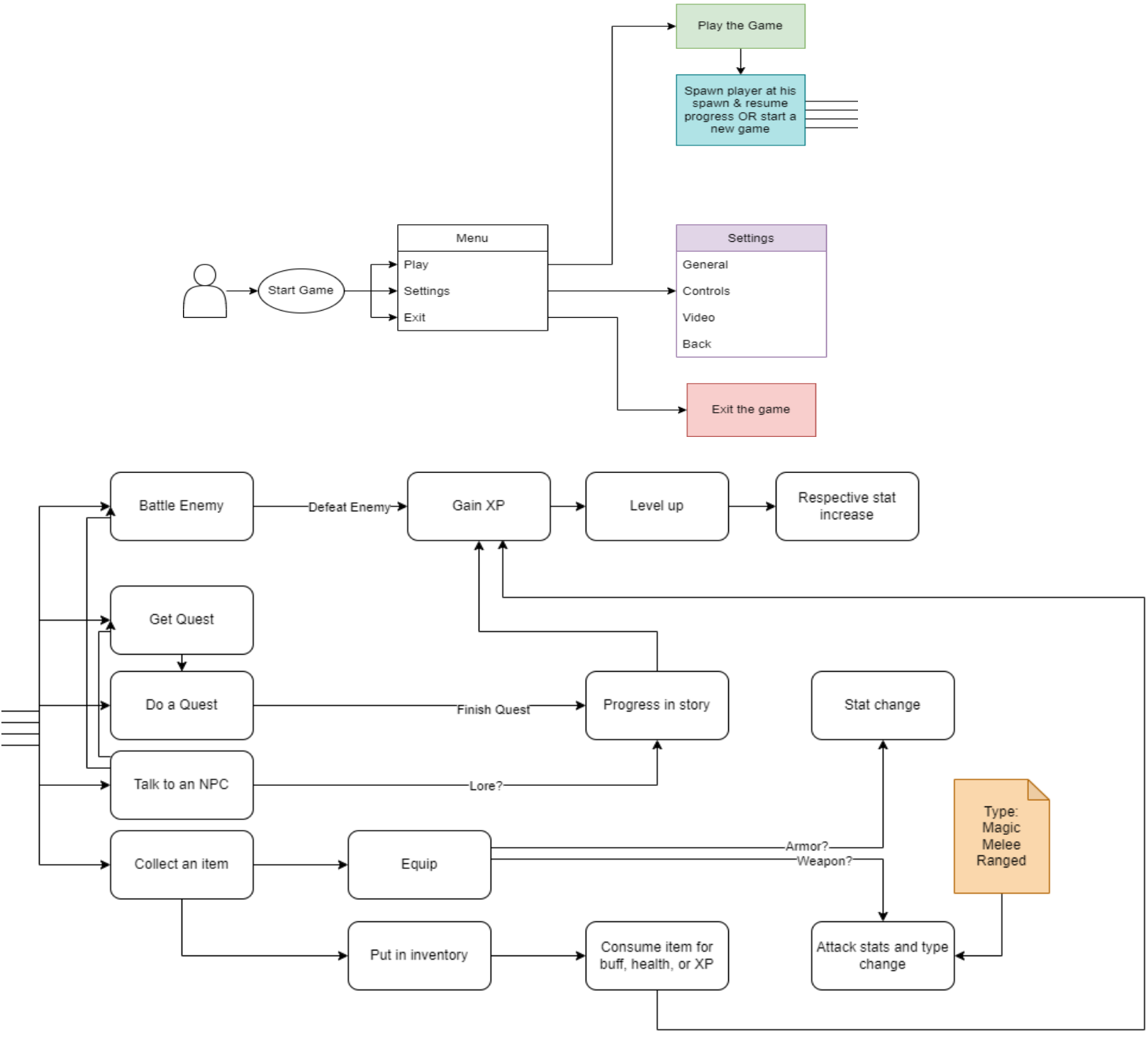
- The game should run normally without crashing.
- There should be no input or visual lag. If any lag exists, it should not notably impact user experience.
- Navigating the menu system should be intuitive..
- The game should be straight to the point, no fluff or cluttered in-game systems for the user to sift through.
- The game should be fun and have some replay value.

Inverse Requirements

- This game will not contain a full campaign, with a concrete beginning and end. While it is our intent to work towards having a full campaign, due to limitations we will discuss within this document, we may only have key parts of the broader vision ready to play.
- Due to our constraints as a team, this game may not be developed to the point where a multiplayer feature is ready by the deadline. Therefore, this game may not have an online component like the original Runescape.

2.0 Usage Scenario

Use Cases/Use Case Diagram


Play the Game
Spawn player at his spawn & resume progress OR start a new game
Menu
Play
Settings
Exit
Start Game
Settings
General
Controls
Video
Back
Exit the game
Battle Enemy
Defeat Enemy
Gain XP
Level up
Respective stat increase
Get Quest
Do a Quest
Finish Quest
Progress in story
Stat change
Talk to an NPC
Lore?
Type:
Magic
Melee
Ranged
Collect an item
Equip
Armor?
Weapon?
Put in inventory
Consume item for buff, health, or XP
Attack stats and type change

4.2 Software Interface Description

External machine interfaces

This game is single player as of now, and as such it will not have any direct interfaces to other machines outside of internal PC hardware.

External system interfaces

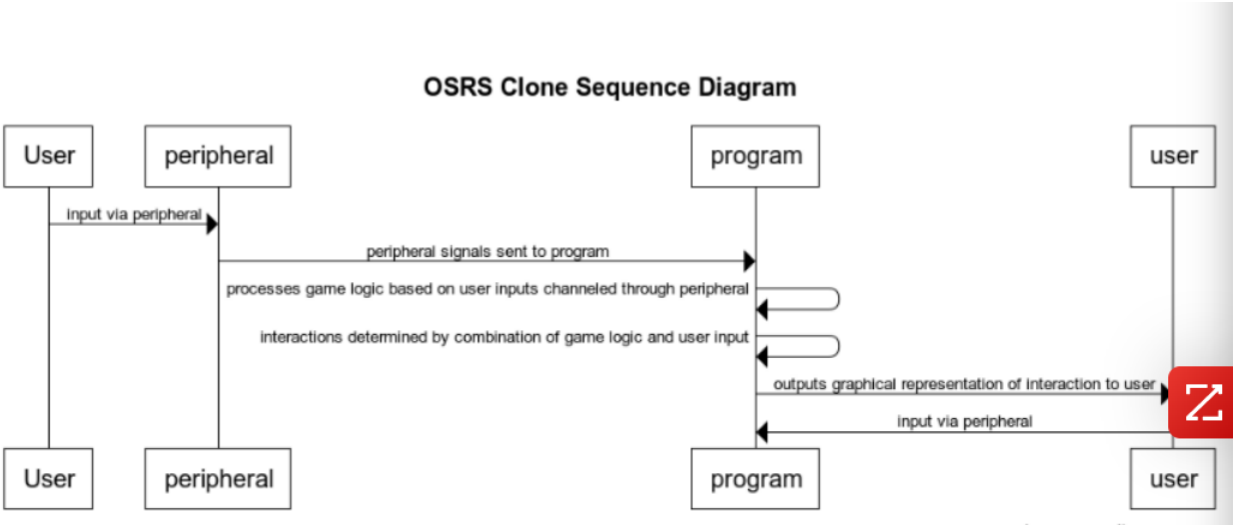
Without a multiplayer component, this game will not have any direct interactions with networks or other systems.

Human Interfaces

Multiple human interface components will be in use, primarily in the form of PC peripherals.

- A monitor will be in use to view the game world and GUI.
- A mouse and keyboard will be in use to register user input.
- Speakers, headphones, soundbar, etc. will be used to play the noises the game makes out to the player.

5.0 Behavioral Model and Description



(Ignore the red square on the right of the diagram - did this on my work laptop and that's an extension I cannot turn off)

6.0 Restrictions, Limitations, and Constraints

Not all team members are able to contribute to the programming section of the project. Because making a video game is tasking and requires hard work, having this section of the work dedicated to just one or two team members will limit the extent to which we can develop our software. We aim to assist each other as much as we can while working together to create good software.

We do not have advanced knowledge of video game design, so much of this project will require learning as we go. This can cause bumps down the road in our project as some functional requirements may take longer than others. We aim to resolve this through collaboration and communication.

There is limited time available to complete this project. All group members have their own schedule, including other classes, work, and personal commitments. Therefore, we will focus on creating the core of the game, something that works as is playable according to what we specified, while saving more ambitious goals for later.

7.0 Validation Criteria

Test Case Categories

- Functionality testing: Make sure all criteria as specified in the use case diagram and functional requirements work properly and without 'jank.'
- User interface testing: Make sure all menus work properly and quickly so as to not inhibit the experience of the game.
- Performance tests: Make sure the game runs without lag at a level sufficient for player enjoyment.
- Software/QA tests: Make sure there are little/zero bugs in the game that outright affect player enjoyment.
- Black box testing: Non-programming members can do black box testing of software requirements due to their lack of knowledge of the code. Tests can include play-testing for bugs, ensuring positive user experience, and general quality assurance. Play-testing can also be a part of the agile method of testing, where the user/customer is part of the development team.
- White box testing: Programming members can review code and create test cases for it. Examples can include testing various aspects of the code to create errors or "break" and see how the software reacts.

Expected Software Response

We expect our software to run according to all requirements specified by iterating on the build after testing. At the moment, we cannot be very specific about inputs and outputs of tests as we are still in early stages of development. We do expect there to be a lot of errors and that the testing we run will help us resolve issues with the software.

Performance bounds

No app crashes are to be expected unless user lies outside of Unity System Requirements

Having a sufficient CPU, GPU, and enough RAM/storage needed for low end 3d gaming will also prevent unwanted crashes from occurrence.

8.0 Appendices

Configuration Management

We have Github for version control. We also use discord for communication and meetings. We have a meeting at least once per week, which functions as our daily standup.

Project Resources

The minimal hardware and software requirements should be nothing out of the ordinary. A regular consumer laptop may or may not be able to run the game, we are not sure at the moment. The user may need a graphics card, but not any software.

Resources we use for the project include any information available online that is helpful for learning software and game development, such as Unity tutorials.

SOFTWARE PROJECT PLAN (SPMP)

2.0 Project Estimates

COCOMO Calculation

Software Project	a_b	b_b	c_b	d_b	Select
Organic	2.4	1.05	2.5	0.38	◉
Semi-detached	3.0	1.12	2.5	0.35	○
Embedded	3.6	1.20	2.5	0.32	○

Calculate Effort and Duration

Effort (E) = $a_b(KLOC)^{b}_{b}$ = 5.45 **Duration (D) = $c_b(E)^{d}_{b}$ =** 4.76

3.0 Risk Management

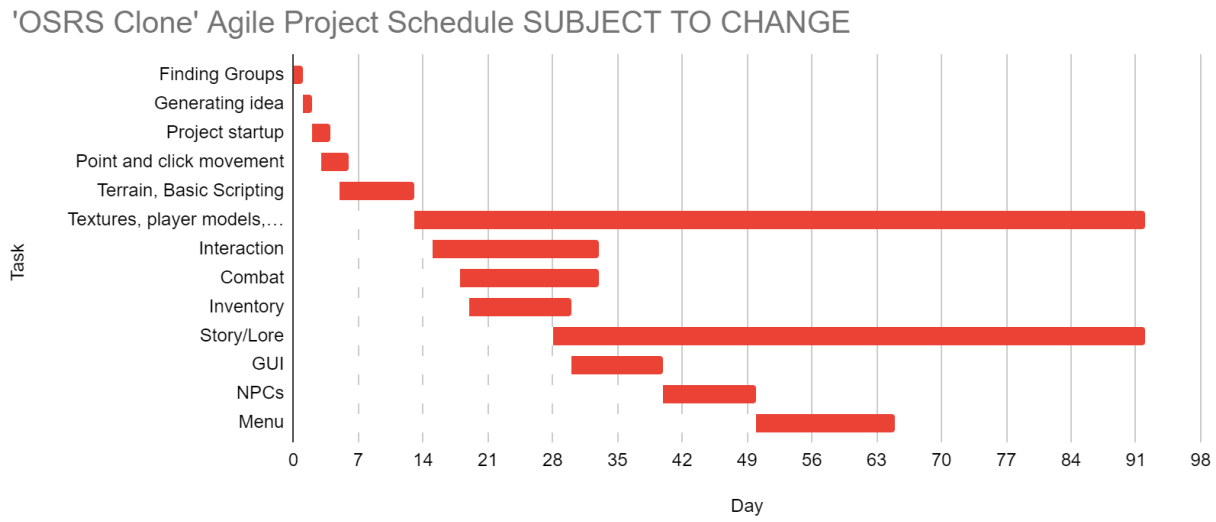
Risk Management Table

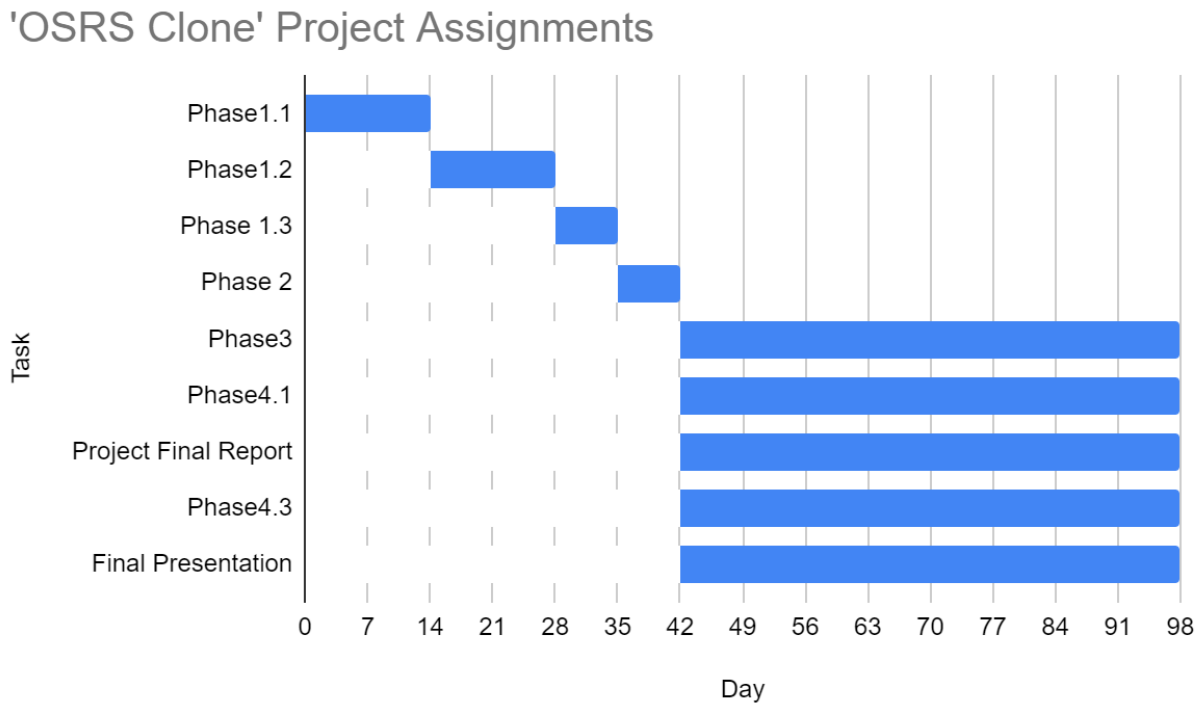
Risk Name	**Risk Description**	**Impact (High, Medium, Low)**	**Probability of occurrence (High, Medium, Low)**	**(Contingency) Plan of Execution**
Time constraints	Team members have their own schedule, limited time due to the project deadline.	High	Medium	Distributing workload accordingly, planning ahead, strong communication, motivating each other
Hardware malfunction	Unforeseen technological malfunctions	High	Low	It would depend on the hardware malfunction, but group members would have to use other computers if needed.
Skill assessment	Not all members of the team can do the same tasks. Only some can program.	Medium	Low	Addressing each members' strengths & weaknesses and assigning them applicable tasks. High level of motivation from key members. Kyle is the one who really wanted to make a video game, and has been excellent and progressing in

				the development on his own, despite that it is mostly him working on the programming for now.
Specification delays	Unable to deliver a sprint containing a functional requirement on time. Can occur due to underestimation of task difficulty.	Medium	High	Being unable to deliver a functional requirement on schedule can heavily impact the future of the project. In this case, the schedule can be adjusted, and requirements can also be changed if needed.
Requirements change	Due to use of agile method, and the complexity of the project, it is likely that the requirements change throughout the development process.	High	Low	This is a low impact risk because we are adaptable and capable of conforming to whatever changes are necessary to continue development. Our strong communication and planning skills will allow us to create a plan to overcome

				obstacles and implement changes when necessary.

4.0 Project Schedule





5.0 Staff Organization

Kyle

Kyle will be doing most of the programming for the game using the Unity game engine and c#. He is also helping with group communication and whatever problems are left outright. Kyle is also the subject matter expert.

Laith

Laith will be creating narrative elements, such as the story and world-building, as well as assisting with any non-programming aspects of the project. Throughout the term, Laith will try to learn the programming needed to assist on that end but may not be able to get far enough from their beginner position to help much. Laith will also function as a theoretical customer/end-user and help give feedback on sprints from that perspective.

Malik

Minimal experience with Java & python, will mostly be providing assistance with non in game graphical assets, and documentation. Malik will also function as a theoretical customer/end-user.

Mitchel

Mitch will be helping with graphic design for things such as the world map. Further, he is available to assist with programming, coordination, and documentation.

Business Owner/Stakeholder: Is the instructor, Professor Hadi. He could also play the role of a customer and give feedback on iterations.