Software Requirements Specification

for

<Web Based Interactive 3d Narrative>

Version 1.0 approved

Prepared by <Ashna Baig and Hassaan Atif>

<Government College University, Lahore>

<13th November, 21>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This provides an overview of the entire project by listing out all the specifications for our Web-Based Interactive 3d narrative (game-like) project. It discusses things pertaining to the functionality, performance, user interface, and different mechanics involved.

1.2 Document Conventions

Not yet defined.

1.3 Intended Audience and Reading Suggestions

The document is intended for all kinds of readers. The developers may get a better gist by reading this document in sequence i.e starting from project scope all the way to the end. The users, however, may not be interested in learning about the performance goals and objectives, system features, or functional requirements. The testers, on the other hand, may be especially interested in those areas (for example, they would like to test out things like the performance of the application and how much load it can take).

1.4 Product Scope

General goals and objectives:

- To be able to render and simulate a 3d environment with game-like mechanics inside a web browser
- Because rendering 3d meshes involves the use of complex algorithms that are very resource intensive, optimal performance is desired.
- Simulation of physics involved with these meshes.
- Some meshes might incorporate custom, handwritten computer shaders.
- Providing game-like mechanics such as moving the character or driving a vehicle and interacting with the 3d world

1.5 References

None yet.

2. Overall Description

2.1 Product Perspective

This project is a web-based game-like 3d based narrative. It is supposed to be a WebApp that is intended to provide entertainment and instant fun through the use of unsophisticated mechanics and an easy interface. The software is freeware.

2.2 Product Functions

- To be able to explore the world. This may be done in either (or all) of the following ways:
  i) Through 3rd person character movement
  ii) Through 1st person character movement
  iii) Through some other means (such as a vehicle)

- To be able to interact with objects through user input (i.e through either mouse, keyboard or a controller).
- Simulating 3d animations inside the web browser
- Providing with a story-driven adventure

2.3 User Classes and Characteristics

TBD.

2.4 Operating Environment

Since this is a WebApp, it will run on almost any Operating System that supports a modern-day Web-Browser like “Google Chrome”. The hardware required to run this web app need not be very advanced. A typical decent machine will work just fine.

2.5 Design and Implementation Constraints

No constraints identified so far.

2.6 User Documentation

A documentation along with tutorial videos will be provided with the software package. These, too, will be delivered through a web page format.

2.7 Assumptions and Dependencies

This project is not really platform-dependent since almost all the modern-day operating systems support a web browser like “Google Chrome”. The only dependency here will be the availability of a browser such as Chrome. Along with that, your browser must also support “WebGL” (Google Chrome already does).
As far as the internal (project-based) dependencies go, the project uses a library called “Three.js” that takes care of a lot of boilerplate code for us (such as line drawing algorithms and other algorithms pertaining to the rendering of meshes).

3. External Interface Requirements

3.1 User Interfaces

3.1.1 Main Menu
There will be a GUI based main menu that the user will interact with through mouse input. This will trigger the actual playable scene from the project. The layout for the screen will be the entire browser window filled with the 3d scene. The theme and the overall style for the main menu remains subject to further discussion, so nothing has been finalized at the moment.

3.1.2 Playable Scene
This scene will be triggered via the input from the main menu. This is where the actual scene occurs and this is where the actual meshes will render. An option may be available for the player to seek help through one of the modes of user input.

3.2 Hardware Interfaces

TBD.

3.3 Software Interfaces

3.3.1 Three.Js library

This library is an abstraction and the user needs not to know anything about it. However, for the developers, this is a useful tool in the sense that it takes care of a lot of boilerplate code for WebGL coding for us (for example, taking care of the line drawing algorithms and matrix transformations for things like rotations and quaternions).

3.3.2 Blender3d Software

Open source 3d modeling tool. This is where the meshes will be modeled and then later on exported to be rendered inside a web browser.

3.3.3 HTML/CSS/Javascript

These are the backend languages that will be used to code the product.

3.3.4 webpack

It is a module bundler to bundle JS files and have them used inside a web browser.

3.4 Communications Interfaces

We will be using HTTP Data Exchange as a communication protocol. Details are to be decided later on.

4. System Features

4.1 Setting up a world

4.1.1 Rendering 3d meshes (Priority: High)

The camera that has been set up should be able to render and draw the meshes onto the screen and this can be done by calling complex line drawing algorithms that are a part of the three.js library.

4.1.2 Configuring camera and transformations (Priority: High)

Making an object of the camera, defining its attributes like (field of depth, layout x, layout y, and its x,y, and z axes).

4.1.3 Functional Requirements

REQ-1: Exploring the world. This is done by being able to roam around and have the camera follow you as you explore your surroundings.

REQ-2: Moving the character. This will be done through user input (keyboard, input, and possibly a controller). The character should respond accordingly to the key pressed. For example, pressing the up arrow key should make it move forward.

REQ-3: Interacting with the world. This can be done through user input. Upon the user input, the application should respond accordingly, and this response can be in the form of animations, text, or one of the meshes that should change their properties (such as their color or their position).

4.2 Simulating the physics (TBD)

TBD.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Seamless rendering

Because drawing 3d meshes onto the screen requires the use of complex line drawing algorithms, it is very important that some effort should be made in reducing the overhead that comes with it and achieving optimal performance by being able to render the 3d world and meshes seamlessly.

5.1.2 Lag-Free playability

While moving around and exploring the world, it is a desired objective to be able to have a lag-free playability or in other words to be able to put extra emphasis on achieving a good frame-rate.

5.2 Safety Requirements

None.

5.3 Security Requirements

None decided yet.

5.4 Software Quality Attributes

5.4.1 Realiability

We want to make sure that the application is reliable and is as error-free as possible. We want to be able to achieve this goal through a fault-tolerant design.

5.4.2 Reusability

We want to be able to make the components as reusable as possible. A good emphasis will be put on clean-code principles, as well as the Object Oriented Paradigm to solve complex problems and the use of the different Design Patterns to encourage this even more.

5.5 Business Rules

No business rules decided yet.

6. Other Requirements

6.1.1 Reusability of Classes

The application will be coded by employing an Object-Oriented Paradigm and at the same time, we will try to follow the best design patterns possible in order to motivate the reusability of various application components.

Appendix A: Glossary

None yet.

Appendix B: Analysis Models

To be determined.

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

- Appendix A: Glossary
- Appendix B: Analysis Models
- Appendix C: To Be Determined List
- Software Quality Attributes
- Security Requirements
- Business rules
- Security Requirements
- Simulating physics
- User Classes and characteristics
- Hardware Interfaces
- References