# Preface

|  |  |  |
| --- | --- | --- |
| Version | Changes / State | Contributor(s) |
| 1.0 | Initial Draft Document | \* |
| 1.1 | Introduction and Elicitation outlined | Robert, Danny, Ceiran |
| 1.2 | User Requirements Draft | Luke, Bruno |
| 1.3 | Functional Requirements | Ben, Bruno, Luke |
| 1.4 | Non-functional Requirements | Ben, Bruno, Luke |
| 1.5 | Updates to Elicitation section | Ceiran |
| 1.6 | Glossary Definitions | Ben |
| 1.7 | Final Updates - Finished Specification | \* |

# Introduction

The system that we are making is a visual, interactive game with 3 levels. The need for the game is entertainment and competitiveness. Most games are developed for entertainment, the thrill comes from overcoming obstacles and defeating enemies as you progress through the levels, ours is no different. We will also have the feature of high scores, this gives the game that competitive edge so that users can compete to beat each other and increase engagement with the system. The storage for high scores will be created using a database.

The system will be programmed in Java by our experienced programmers. The visual graphics will be created by a graphical animator on our team. As for the game itself, as the player progresses, it will become more difficult and challenging. In the start the player will arrive in a room filled with enemies which he will have to defeat to progress to the next level.

Similar systems that our game can be compared to is an old 90’s game called DOOM. Although DOOM was a 3D game and ours is not, they are very similar in which you have to defeat enemies to progress through levels and are given a progress report as you complete each level. In addition, they are both shooters where the player has several weapons of choice to defeat these enemies and they both implement a high scores system.

The functionality of our system overlaps through entertaining our users in several ways at once. For example, a user may complete the game in a certain amount of time but may not be happy with how the game went for them. Therefore, the competitive edge comes in hand here because the user may want to play the game again to beat their previous record. This also gives it an addictive functionality as it keeps the user intrigued in our game.

# 

# 

# Requirements Elicitation

Gathering the requirements for our complex game system called for us to interview some of our stakeholders, people who enjoy playing video games and competitive score-based systems.

What we discovered were the features our users would like, how certain functions should work and the constraints we will have to work with when designing and developing the system.

## Scenarios

The following scenarios were put forth by our stakeholders in interviews, they explore what the possible actions for our system might be and the issues involved. We conducted a wide range of interviews and used the most commonly given scenarios.

|  |  |
| --- | --- |
| ***Scenario*** | The user opens the program ready to play the game, having never seen the program before. |
| **How it will be implemented** | *Would be done with prominent buttons as soon as the program is started. This will be shown in the form of a main menu with two buttons stating ‘start game’ and ‘controls’. This means that if the user hasn’t started the game before they can familiarise themselves with the controls and then start and also be able to play the game straight away if they already have done before.* |
| **Issues** | *The user might feel overwhelmed with lots of buttons on screen, they may not know where to click and also what they are doing. This could result in them not being able to start and play the game.* |
| **Resolution** | *We would make the main menu very simple and easily understandable, this would be done in a simple graphical format with easy to read text and clear directional instructions. We will only be using buttons that are necessary for the user and their needs and will not be adding anything that may ultimately hinder the user playing the game efficiently. As the user wants a not overwhelming experience we feel as this is the most adequate way to do so.* |
| **Activities operating at the same time** | *The game will be fully loaded and ready to play, with the clear instructions on how to either start the game straight away or go to the controls menu. This means that there is no loading delay from when the user enters the controls menu and starts the game.* |
| **System state when scenario is finished** | *The system state will be different once that scenario has finished, the user will have either decided to go into the controls menu to learn the controls of the game or will have either started the game and is playing. Both of these states are different to the Scenario state.* |

|  |  |
| --- | --- |
| ***Scenario*** | User wants to interact with game, I.E move and shoot. |
| **How it will be implemented** | *The user would use the W,A,S,D keys or arrow keys and the mouse to control character movements and interactions. W,A,S,D AND arrows would be used for moving the in game character; the mouse for aiming; left click for shooting.* |
| **Issues** | *Users may not know key bindings or may want to change the current key bindings for different ones.* |
| **Resolution** | *To check key bindings the user can use the escape key to pause the game, then select controls. From here the controls are listed and can be looked up by the user.*  *To change key bindings the game is paused using the escape key, selecting controls, then selecting the control they want to change. The user will then be prompted to press the key they want to be used for the selected control.* |
| **Activities operating at the same time** | *The database will be loaded in the background ready to save new controls if the user has selected different controls. The same stable version of the game will be ready to be played once the controls have been edited or kept the same. Also images will have been loaded by the software displaying the game.* |
| **System state when scenario is finished** | *The system state will be the main menu screen once again, the user will have options to either customize controls or start the game. This is because for the user to be interacting with the game (move and shoot) the game will have started.* |

# User Requirements

This information is based on interviews and scenarios with the stakeholders of our system. We interviewed a number people within the target audience and this information was useful in acquiring an understanding of what they want from our product.

Some of the questions we asked included:

* What are your needs from a game in general?
* (After explaining concept of game) Would you play a game like this?
* How often would you play it?
* What would you need from a game like this?
* What is most important to you when choosing to play a game? (e.g. story, graphics etc.)
* What features do you dislike in games in general?
* What features do you dislike in playing games that are within the same genre as ours?

The results we found were as follows:

* The system shall provide a way to open and exit the program
* The system shall provide simple controls that are user friendly and intuitive
* The system shall provide a dynamic scoreboard so scores will be trackable for the user
* The system shall include gameplay entertainment, similar to what is found in other systems of the same type
* The system shall use clear and understandable language
* The system shall provide background audio to supplement the graphical entertainment
* The system shall implement clear objectives for users, so they have a clear goal to get through the levels
* The system shall make navigation easy to understand, and it shouldn’t take too long to navigate to go where the user is trying to go e.g. should not take more than five clicks to start playing the game
* The system shall provide instructions for users

# System Requirements

The requirements for our system have been broken down and specified further. We have categorised them into *functional* and *non-functional.*

## Functional

These are actual, defined components necessary to the program.

|  |  |
| --- | --- |
| **Name** | **Priority** |
| Input | High |
| **Notes:**  System must be interacted with using keyboard and mouse. | |

|  |  |
| --- | --- |
| **Name** | **Priority** |
| Controls | High |
| **Notes:**  System must allow for user to be able to control the character in game.  System shouldn’t feature too many inputs/controls. | |

|  |  |
| --- | --- |
| **Name** | **Priority** |
| SystemDisplay | High |
| **Notes:**  System must be displayed in a window visually, on a Windows computer. | |

|  |  |
| --- | --- |
| **Name** | **Priority** |
| Scores | Medium |
| **Notes:**  System must save inputs as a score with name. (Data manipulation)  Game should have a system to judge how well the person playing the game is doing. | |

|  |  |
| --- | --- |
| **Name** | **Priority** |
| Sounds | Low |
| **Notes:**  System must include background music and sound effects, appropriate sound effects will be allocated to each action. | |

|  |  |
| --- | --- |
| **Name** | **Priority** |
| Instructions | Low |
| **Notes:**  System must provide an instructions as guidance. | |

|  |  |
| --- | --- |
| **Name** | **Priority** |
| Store | Low |
| **Notes:**  Should allow user to buy items in game, such as weapons, upgrades and accessories. | |

|  |  |
| --- | --- |
| **Name** | **Priority** |
| Interactions | High |
| **Notes:**  Should be able to engage with the environment elements, such as enemies. | |

|  |  |
| --- | --- |
| **Name** | **Priority** |
| HUD | Low |
| **Notes:**  System should show a HUD for things like ammo, lives, Difficulty etc. | |

|  |  |
| --- | --- |
| **Name** | **Priority** |
| Support | High |
| **Notes:**  System Should provide a method that lets the user Email the developer if any problems happen. | |

## Non-functional

These are requirements that aren’t absolute but would be expected of the system.

* Game should load up reasonably quick.
* System shall have a fast frame rate to allow for smooth performance.
* Navigation should not take more than a certain amount of clicks. For example navigating from the main menu to gameplay should not take more than five clicks. This helps with the usability of the system.
* Overall game size should not be too large, so it doesn’t take up too much space of the user's computer.
* Game should be able run with the small amounts of RAM, for flexibility of users.
* Systems response time of clicking should not be too long, as this is an inconvenience for users.
* Game should be able to run on most Windows platforms.
* System shall be maintainable, to be able to adjust for bugs and glitches.
* System should run and process data smoothly and should not affect gameplay.

# Glossary

Terms and their meanings.

**WASD** - This is a keyboard control system used in many PC games where the up/forward, left, right and down/backward character movement being assigned to the W, A, S and D keys.

**HUD** - Heads Up Display, this is typically the information presented on-screen for the player.