

CMPE 491

PROJECT SPECIFICATION REPORT

for

“Sanctified Retribution”

By

***Metehan TÜTER
Elif Aysu KÜRŞAD
Ceren Bilge OYAR
Erkan SANCAK***

Date: 31.03.2023

Table of Contents

1. Introduction	3
1.1. Description:	3
1.2. Constraints:	4
1.3. Professional and Ethical Issues	4
2. Requirements	4
2.1. Menu Display	4
2.1.1. Stimulus/Response	5
2.2. In-Game Display	6
2.2.1.1. Stimulus:	6
2.2.1.2. Response:	6
2.2.2.1. Stimulus:	6
2.2.2.2. Response:	6
2.3. Map Display	6
2.3.1.1. Stimulus:	6
2.3.1.2. Response:	6
2.3.2.1. Stimulus:	7
2.3.2.2. Response:	7
2.4. Software System Attributes	7
2.4.1. Reliability:	7
2.4.2. Performance:	7
2.4.3. Usability:	7
2.4.4. Security:	7
2.4.5. Scalability:	7
2.4.6. Compatibility:	7
2.4.7. Maintainability:	7
2.4.8. Portability:	7
2.5. Design Constraints	8
2.6. Performance Requirements	8
3. Overall Description	8
3.1. Product Perspective	8
3.2. System Interfaces	9
3.3. User Interfaces	9
3.4. Hardware Interfaces	9
3.5. Software Interfaces	10

3.6. User Characteristics.....	10
3.7. Assumptions and Dependencies.....	10
4.Planning.....	10
4.1 .Team Structure	10
4.2. Estimation	10
4.3. Process Model	11

1. Introduction

1.1. Description:

The development that we started is a 2-D Roguelike action game with pixel art style and a story. The game engines became more and more useful and functional over time, and that's why we picked the "Unity" game engine for our project. Unity is a software framework primarily designed for the development of video games and generally includes relevant libraries and support programs with also using C#. We will also use the programs "Aseprite" and "Blender" for our map and overall character and item designs. The game focuses on some interactions with the side characters and the enemies, and with the map itself. The enemies will also be going to have an A.I. system that when the player gets close enough, it will trigger the enemy and the enemy will start what it needs to do in the story. There will be different types of enemies that will change the way the player plays. The game we are trying to create is specifically for over 15 years old players. We will also try to create an interesting map design that allows players to explore all types of content. If players do not understand what to do in some parts of the story, we will be going to put some clues in-game in order to clarify what to do in that situation. To give an example, if the mini boss that player needs to defeat has some gimmick that is the only way of defeating it, the boss will give the clue when it is speaking in-game.

1.2. Constraints:

Developing a 2D Roguelike game with a variety of features, mechanics, and an engaging storyline has imposed several constraints on us. Time constraints have made it difficult to meet project deadlines, while budget constraints have made it challenging to manage project costs. We have faced technical constraints when the development team lacked the skills to implement certain features or mechanics. Additionally, content creation has been time-consuming and at times required specialized skills. Quality assurance has been an ongoing process throughout development to ensure the game is stable and bug-free. Making the game work on different platforms and protecting user information has required us to take various security measures. In addition to the constraints mentioned above, the development team has faced other potential challenges, such as the need for careful planning and execution when designing a game that is both challenging and fun, and the requirement for strong writing and storytelling skills to ensure the game's plot is engaging.

1.3. Professional and Ethical Issues

It is obvious that a large amount of resources are required to make a game with many mechanics, features, maps, etc. Since we are not able to meet all these resources ourselves, we can use external resources to meet our needs. These things we use will be used in an ethical framework by obtaining permissions and giving the necessary references. These resources can be, for example, assets used for things like map and character design, and can be used to create a smoother and bug-free gameplay by using some source code. As there may be violence and blood in the game, proper warnings will be made to users to avoid potentially disturbing content. Moreover, age restriction can be considered.

2. Requirements

2.1. Menu Display

Player will be going to access the menu when they open the game. The system will be going to allow players to select or create a save data. They can see a list of saved data.

Menu display is user-friendly and easy to understand, as players will be interacting with it frequently throughout their gameplay.

Additionally, we provided clear instructions about how to use the menu to help avoid confusion.

2.1.1. Stimulus/Response

When the player enters the game, the system will display a list of available in-game options such as “New Game” that will going to allow the player to create a new save data for the game, “Load Game” for using the old save data of the player, Options in order to change language and adjust the sound settings, and a “Exit” button in order to close the game.

2.1.1.1. Stimulus:

When the player clicks on the “New Game” button.

2.1.1.2. Response:

A new game save will be created and the game will start

2.1.2.1. Stimulus:

When the player clicks on the “Load Game” button.

2.1.2.2. Response:

The game will be loaded from the save point that the user has created before.

2.1.3.1. Stimulus:

When the player clicks on the “Options” button.

2.1.3.2. Response:

A settings menu will open so players can change settings such as language and sound.

2.1.4.1. Stimulus:

When the player clicks on the “Exit” button.

2.1.4.2. Response:

Game will shut down.

2.2. In-Game Display

Player will be going to access the menu with a specific key that is assigned for it. Also the player will access the items that they found inside the game in their inventory with an assigned key. Also player will be able to see their usable items in the game with a small area that has the item's icons.

2.2.1.1. Stimulus:

When the player clicks on the "menu" key that is assigned for it.

2.2.1.2. Response:

The menu of the game will open.

2.2.2.1. Stimulus:

When the player clicks on the "Item Menu" key that is assigned for it.

2.2.2.2. Response:

An item menu will open where the player can access the items they have.

2.3. Map Display

Player will be going to access the map with a specific key that is assigned for it. Player will see the available parts of the map when opens the map. If the player has not found a piece of map, the game will show the area as a shadow in order to give information to the player.

2.3.1.1. Stimulus:

When the player clicks on the "Map" key that is assigned for it.

2.3.1.2. Response:

The map of the game will open.

2.3.2.1. Stimulus:

If the player does not obtain the specific map piece.

2.3.2.2. Response:

The game will show that part of the map as shaded

2.4. Software System Attributes

2.4.1. Reliability:

The game is stable and free from crashes or game-breaking bugs.

2.4.2. Performance:

The game runs smoothly without significant lag or frame rate drops.

2.4.3. Usability:

The game is intuitive and easy to understand for the players of all skill levels.

2.4.4. Security:

The game protects user data and it prevents cheating.

2.4.5. Scalability:

The game is designed with future updates and expansions in mind.

2.4.6. Compatibility:

The game is compatible with a wide range of hardware and software configurations.

2.4.7. Maintainability:

The game developed by our team is easy to maintain and update.

2.4.8. Portability:

The game is portable for use on different devices.

2.5. Design Constraints

Reports during the development phase of the project will be prepared according to IEEEStd830. UML design and flow charts are going to take part in that phase.

2.6. Performance Requirements

Inside the game, the enemies must respond in a fixed time period in order to give the player a chance to attack and defeat the enemy but also can punish the player if the player makes a mistake in order to make the game a bit hard. The number of enemies will be limited, and once defeated, they will not respawn unless the player chooses to load the game from the last save file, choose to rest in safe areas or returns to the checkpoint area after dying in game.

3. Overall Description

3.1. Product Perspective

Our game is aimed toward players who like stories with some challenges. The product is independent and mostly self-contained. Sanctified Retribution is a rogue-like game whose all internal designs belong to our project group. However the project process contains development of interfaces to use an AI system for the enemies for making the game experience a bit more entertaining . This means at the end of the year we will have one complete game with a storyline, some puzzles to solve in order to give players a bit time to relax from the action and various enemies to defeat.

In addition, we are also aiming to provide an immersive and visually appealing experience for players through the use of pixel art and 2D graphics. The game's soundtrack and sound effects will also be carefully designed to enhance the overall atmosphere of the game.

3.2. System Interfaces

Sanctified Retribution will be developed in Unity game engine, and all components will be executed on Windows.

In terms of system interface, the game will have a user friendly and intuitive interface design. The game menu has clear and concise options for players to select, including options for starting a new game, loading a saved game, adjusting game setting, and exiting the game. The system interface has a design to enhance the player's experience and make it easy for them to fully immerse themselves in the game world.

3.3. User Interfaces

The user interface of the game is going to be user friendly and very easy to get used to. Game screen will be embedded into the game. There will be information about how to play the game and some tutorials in order to teach the game to the player. And the game screen displays all necessary information about the game, such as the player's health, available items and weapons and progress through the game.

The user interface also includes interactive elements such as buttons and menus. In addition, the game provides helpful hints and tutorials to teach players how to play the game and how to navigate the various challenges they will encounter. The overall design of the user interface is intuitive and easy to use, even for players who are new to the game or to gaming in general.

3.4. Hardware Interfaces

There is no constraint on which kind of hardware must be used. There are common hardware devices that are enough to interact with the game. These are:

- Monitor screen: Screen provides visual information to the user.
- Keyboard: Keyboard provides the player to play the game.
- Controller: If the customer does not want to experience the game with a keyboard, can use a controller instead of it.

3.5. Software Interfaces

The required software products for Sanctified Retribution project are Aseprite, Blender, Unity and C#.

3.6. User Characteristics

The adult players will be a potential user for the product. There will be an age limit.

3.7. Assumptions and Dependencies

The game can be a bit challenging from time to time in order to give the player a “success” feeling. There will be some gimmicks for some specific enemies that players will need to use in order to defeat them. If the player does not understand what to do in that situation, sometimes the game can be given some sort of clues in order to help the player.

4.Planning

4.1 .Team Structure

Metehan Tüter, Elif Aysu Kürşad, Erkan Sancak, and Ceren Bilge Oyar make up our crew. We make critical decisions together . Every step of the project will be chosen by the group, and carried out by the person to whom it is delegated. Also, we will meet regularly with our manager, Frat Akba, to discuss the situation, get his or her opinion, and choose the next steps. We spoke often with one another in addition to our weekly meetings to check on the status of our individual work and make sure the project deadlines were met. We assigned duties to each team member, which made it easier to make sure that everyone made a significant contribution to the project.

4.2. Estimation

Until the end of this semester, we planned to at least reach the halfway point of the development. Game will be accessible for beta testing. In the second semester, we will focus on mainly creating stages and start to create functional enemies that can act differently in different situations.

4.3. Process Model

We have divided the development process into various stages to make it more efficient, and we are using the Spiral model, which is flexible and allows us to adapt to any changes during development. Each phase will have the necessary documentation completed, and we will prioritize risk management and testing to ensure that the final product meets our quality standards.