**Requirements Document**

**<** Space Crucible**>**

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision # | Author | Revision Date | Comments |
| 1.0 | Isaac Colon | August 26, 2021 | initiated |
| 2.0 | Isaac Colon,  Parth Patel,  Meshwa Patel,  Yifan Zhang,  Kwadwo Gyasi-Danquah | September 3, 2021 | Added user stories, diagram, new project name, and updated project designs using group ideas |
| 3.0 | Isaac Colon,  Parth Patel,  Meshwa Patel,  Kwadwo Gyasi-Danquah | September 15, 2021 | added use cases and updated block diagram, added feature and general requirements |
| 4.0 | Parth Patel, Isaac Colon | November 19, 2021 | Revised use cases, system overview, block diagram, features and requirements |

**Table of Contents**

[System Overview 4](#_Toc88428406)

[System Block Diagram 6](#_Toc88428407)

[Document Overview 7](#_Toc88428408)

[General Requirements 7](#_Toc88428409)

[Features and Requirements 7](#_Toc88428410)

[Use Cases 8](#_Toc88428411)

[Use Case #1: 8](#_Toc88428412)

[Use Case #2: 9](#_Toc88428413)

[Use Case #3: 9](#_Toc88428414)

[Use Case #4: 9](#_Toc88428415)

[Use Case #5: 10](#_Toc88428416)

[Use Case #6: 10](#_Toc88428417)

[Use Case #7: 10](#_Toc88428418)

[Use Case #8: 10](#_Toc88428419)

## System Overview

Space Crucible is a two-dimensional, top-down perspective action-puzzle game with support for both single and multiple players. Levels will be defined as a series of square tiles on a fixed grid- however, player and monster movement will not be locked to discrete points on this grid. The combat will be in real-time and take place on tile-based levels. Space Crucible will have a science fiction theme and will emphasize run-and-gun combat style. Players take on the role of “Exterminators” and must use a variety of tactics to combat foes, from weaponry to coercing them to fight each other.

The objective is to explore the map while eliminating monsters, dodging traps, and finding the exit to proceed to the next level. Multiplayer is a key feature; levels can be designed that require more than one player to complete (in fact, the multiplayer-oriented level design will be prioritized). A level editor will be included so users can create their scenarios and save them in a simple text-based format. Levels can be compiled into level packs using a simple archive format called a “.WAD”, which contains and organizes any necessary graphics, sounds, music, and monster code (defined in scripts, rather than hard-coded Java).

Players will receive a launcher upon starting the game. The launcher will allow the user to pick a level pack, a specific level from the pack, choose a difficulty, and either launch a single-player session or join a multiplayer session by inputting the lobby code in the join lobby menu. Real-time action will use WASD or arrow keys to move the player, while the player can simultaneously aim using the mouse. Certain map tiles or objects can be interacted with to proceed in the level or trigger traps. An in-game chat will allow players to communicate with each other. A straightforward light system will allow tiles far from light “sources” to darken, obscuring important puzzle components or hiding sneak attacks. MIDI files will be used as background music.

There will be a Master server that will handle all clients connecting to the multiplayer game mode. Once a player has selected create a lobby or join a lobby option, they will be directed to a specific game server. Each lobby will have a game server that will handle all incoming connections and data from the clients. In multiplayer mode, the game server controls the whole game to ensure that all players are in sync and the host does not have an advantage over other players. The clients are only responsible for sending their input data to the game server; the game server handles moving the players and sending the updated rendering data back to the clients.

Players will be able to host their own lobbies by selecting Create Lobby option in the co-op mode menu. Each lobby will have its unique 4-digit code that the host can share with other players to invite them to the lobby. The 4-digit code is linked with the lobby’s IP address and port number. When the client makes a request to create a lobby, the master server sends the lobby info to the client, and then the client joins the lobby. The players who want to join an existing lobby will choose the Join Lobby option and input the unique 4-digit code. The join lobby request is sent to the master server, and the master server finds the lobby associated with the code and sends the lobby details back to the client. The lobby host will have access to the difficulty selection menu and start game option that other players won’t be able to see. Once the host selects the start game option, the level beings for all players in the lobby. If the host leaves the lobby before starting a game, the host privileges are passed on to the player who joined second.

Once the lobby host starts the level, new players can still join the ongoing level as long as they have the lobby code. This feature also allows players to leave and join the ongoing game as they wish. Another feature we implemented is if a player leaves in an ongoing game, the player will be replaced by a bot/AI player who will help the players finish the level. The bot player can follow the closet player and shoot and chase the visible enemies.

Server administrators can use the remote control (RCON) application to monitor the servers and send instructions to perform certain tasks. The RCON client can log into both the master server and the game servers using a lobby code (or MASTER for the master server) and a password (the master server’s host chooses the master password, the game server password is generated randomly upon lobby creation). When RCON is connected to the master server, it can monitor the number of game servers the master is hosting and check their lobby codes, RCON passwords, and add-on files. When connected to a game server, player positions and health and packet information can be monitored in real-time, chat can be sent to the players from the server, and settings such as game skill and level can be changed.

The program will be written in Java and will support all desktop operating systems (i.e., Windows, macOS, and Linux). There will be separate executable JAR files for the master and game servers, client programs.

A Java IDE will be necessary to develop this project, preferably a common one shared by all developers. The Java LibGDX library is a game development library that includes graphic, sound, music, and networking functionality and will be the framework we use to develop the game. An open-source library, DoomStruct, exists to manipulate .WAD files, which will be used to access game data neatly in an archive format. The KryoNet library is used for network operations, exchanging packets between the game client and the master and game servers over the TCP protocol.

## System Block Diagram

Diagram

Description automatically generated

**Figure 1**: System Block Diagram

The Space Crucible project will be split into three executables: client application, master server, and game server. Figure 1 shows how the three main components interact with each other.

The master server handles all clients connecting to the multiplayer mode. The clients send two types of requests to the master server, create a lobby, and join a lobby which the master server, in turn, redirects the clients to the game servers. The game servers also connect to the master server as clients to send a ping when all players have left that server and tell the master server that they are free and ready to be used again.

The client program is one of an arbitrary amount of player clients simultaneously connected to the server program. The client program should only have two main responsibilities. The first is to take user input in order to control that user’s player (and *only* that user’s player) and send that data to the server. The second client-side responsibility is to render the game scene from the user’s character’s perspective. Therefore, the client receives information about the current game state from the server and renders the graphics and sounds representing everything that the player should be able to see and hear, respectively. It also plays the level’s music back on the user’s side.

In cooperative mode, the server is the “brain” of the whole game and controls every aspect of the game except for the players (controlled by their respective clients). This means any non-player characters (i.e., monsters), any moving parts in the level such as doors or traps, any ambient effects, any potential level scripting effects, etc. The server also starts the lobby whenever a new game is initiated and starts the game loop properly once every user has joined. In single-player mode, this game logic is run in a thread on the game client itself.

The level editor is used to create custom scenarios for Space Crucible. One level is represented by one file in a text-based format which describes all of the walls, monsters, player start points, and other map objects in that level. Multiple levels are then stored combined in a .WAD archive format. Groups of levels designed to be played in sequence can be defined as “episodes”. WAD files can also store graphics, sounds, and music, which are packaged alongside new levels as needed by the level designer. For instance, if a new user-level took place on an alien planet which did not resemble any location in the base game, the .WAD would include new level background graphics that better represent the appearance of this new setting.

## Document Overview

The document will provide a high-level overview of the game as well as define the features and requirements. It includes the following sections:

* System Overview
* General Requirements
* Features and Requirements
* Use Cases

## General Requirements

* Desktop or Laptop running Windows, Mac, or Linux operating systems
* Master and Game Servers will run on a Linux machine (headless, desktop or laptop)
* Java runtime environment
* Keyboard and mouse to control in-game movement
* Optional - Internet connection to access multiplayer mode

## Features and Requirements

* Space Crucibles will have a main menu that will be presented when players load into the game
  + The main menu will have five options: Single player, Co-op, Level Editor, Settings, Exit
* The player will be able to enter their in-game username in the settings menu
* Each predefined level in the game will have five difficulties, Very Easy, Easy, Medium, Hard, and Nightmare
  + Each difficulty level will get progressively harder by spawning more monsters, monsters will respawn in harder levels, monsters will have increased health, and be able to run faster.
* The Single player option will show a level difficulty window that will allow the user to select the difficulty of the level
  + Once a difficulty is selected the player will spawn in-game and will be ready to play the game
* The Co-op menu will have three options: Create Lobby, Join Lobby, and Back
  + In the Create Lobby option the player will be able to share a four-digit letter/numeric lobby code that routes to the IP address and port number of the lobby
  + In the Join Lobby option, the player will be able to input the lobby code to join their friend’s lobby
  + The Back option will bring them back to the main lobby
* In the lobby, the lobby host will be able to change the difficulty level of the game
* The level editor will allow a user to create a new level from scratch using game assets
* If the player is playing a game in co-op mode leaves the game a bot/AI player will replace the player and help the teammate finish the level.
* The ESC key menu will have two options: Restart Level, and Return To Title Screen
* In the settings menu the player will be able to change the volume of in-game sounds and music, change username, and add addons
* The lobbies have no limit to how many players can join so the user can invite unlimited number of players to the lobby
* The game will allow new players to join in middle of co-op game
* In both single-player and multi-player modes the players will have a minimap that has the layout for the whole map.
* The game loop will run at 55 ticks per second to improve input precision
* The players will be able to create their own levels and add custom entities to the game
* The players will be able to play their custom levels with each other in co-op mode.
* Different types of monsters with different abilities will be introduced to the game
* There will be an in-game chat that players will be able to communicate through by typing.

## Use Cases

### Use Case #1:

Harold wants to spend more time with his grandson David, who is very much into the shooter genre. Harold wants to play with his grandson, so he creates a lobby and invites him:

1. From the main menu Harold clicks “Co-op” option and is presented with two options “Create lobby” and “Join lobby”.
2. Harold clicks Create lobby option to host his own lobby. When he loads into the lobby, he sees a lobby code in the bottom left corner of the screen. Harold shares his lobby code to David.
3. To join his grandfather’s lobby, David clicks on the “Join lobby” option. He is prompted with a text box where he can input the lobby code, David inputs the lobby code and clicks “Ok”.
4. David loads into the lobby and both Harold and David are able to see each other’s usernames in the lobby. Since Harold is the host of the lobby, he has access to the start game option. Harold clicks start game and two players load into the game.

### Use Case #2:

Alex just downloaded the game and wants to play the game in single player mode.

1. Alex boots up the game and loads into the main menu
2. Alex selects single player on the main menu
3. Alex selects “very easy” option in the difficulty window and clicks “Ok” to confirm
4. Alex’s player entity loads into the game
5. Alex presses W, A, S, and D on his keyboard to control the up, left, right, and down movement of the player
6. Alex moves the mouse to aim his weapon and clicks left click to shoot his weapon
7. Alex explores the map and kills the monsters along the way.
8. Alex finds the exit and runs to it when he reached the exit, he spawns into the next level.

### Use Case #3:

David recently downloaded Space Crucible out of curiosity and blasted through the game with his friends. They enjoyed the game, but now they have completed all of the default levels and are quite bored. However, thanks to the game’s built-in level editor, David can resolve this dilemma by creating his own levels.

1. David opens the grid-based editor and first creates the shapes of the rooms in his new scenario.
2. Then, he adds the player start points, and any monsters, traps, or scenery in the level after that.
3. When he saves his custom level, the editor takes the new level info and stores it in a simple text-based format.
4. He can compile many of these level files in a .WAD archive, along with any new graphics or sounds he may need.
5. He can then distribute the .WAD to his friends and play his new scenario- or even post it on the internet for strangers to download and play.

### Use Case #4:

Bob wants to change his in-game username.

1. Bob loads into the main menu.
2. Bob selects the Settings option on the main menu.
3. Bob clicks the “Change player name” option.
4. Bob enters “Bob” in the text field and clicks “Confirm”.
5. Now when Bob plays the game, he has a username on top of his player that says “Bob”.

### Use Case #5:

John gets a phone call and has to leave the game. But after he is done, he wants to join back into the lobby.

1. John is playing a game in co-op mode with his friends.
2. John gets a phone call and has to attend the call.
3. John exits the game, and a bot player replaces him.
4. The bot player plays on John’s behalf.
5. John finishes the call and rejoins the game using the same lobby code.

### Use Case #6:

James wants to mute the music in-game because he wants to listen to his own music.

1. James clicks on the “Settings” option on the main menu screen.
2. In the settings menu James selects “Volume” option.
3. Inside the volume menu James turns down the SFX slider to zero and BGM slider to zero.
4. Both game volume and music get muted so now James can listen to his own music.

### Use Case #7:

Pedro’s friend created a new, challenging level and Pedro wants to help him test it.

1. Pedro downloads the .WAD file that his friend created
2. Pedro opens the Space Crucible client.
3. Pedro uses the built-in file browser and chooses the downloaded .WAD file.
4. He can now play the game with the new levels, and any other additions.

### Use Case #8:

Sam has beat all the levels by himself in the easy difficulty mode. He wants more of a challenge, so he wants to turn up the difficulty to the highest mode.

1. Sam selects the single-player option on the main menu.
2. In the choose difficulty level menu he chooses the nightmare option which is the hardest difficulty in the game.
3. When Sam clicks go, he spawns into the game and is immediately ambushed by a monster.
4. Sam finds this mode to be very challenging and has fun trying to clear it.