

CS 4ZP6A: Software Requirements  
Specification  
Cat-and-Mouse Game

Team #8, Claw-some Games

Yuan Gao (1330064)

Su Gao (1330065)

James Lee (1318125)

James Zhu (1317457)

November 18, 2016

# Contents

<b>1</b>	<b>Project Drivers</b>	<b>1</b>
1.1	The Purpose of the Project . . . . .	1
1.1.1	Background of the Project . . . . .	1
1.1.2	Goal of the Project . . . . .	1
1.2	The Stakeholders . . . . .	1
1.2.1	The Client . . . . .	1
1.2.2	The Customers . . . . .	1
1.2.3	Other Stakeholders . . . . .	2
1.3	Mandated Constraints . . . . .	2
1.3.1	Solution Constraints . . . . .	2
1.3.2	Implementation Environment of the Current System . . . . .	2
1.3.3	Partner or Collaborative Applications . . . . .	2
1.3.4	Off-the-Shelf Software . . . . .	3
1.3.5	Anticipated Workplace Environment . . . . .	3
1.3.6	Schedule Constraints . . . . .	3
1.3.7	Budget Constraints . . . . .	3
1.3.8	Enterprise Constraints . . . . .	3
1.4	Naming Conventions and Terminology . . . . .	4
1.5	Relevant Facts and Assumptions . . . . .	4
<b>2</b>	<b>Functional Requirements</b>	<b>4</b>
2.1	The Scope of the Work and the Product . . . . .	4
2.1.1	The Context of the Work . . . . .	5
2.1.2	Work Partitioning . . . . .	6
2.1.3	Individual Product Use Cases . . . . .	11
2.2	Functional Requirements . . . . .	11
<b>3</b>	<b>Non-Functional Requirements</b>	<b>14</b>
3.1	Look and Feel Requirements . . . . .	14
3.1.1	Appearance Requirements . . . . .	14
3.1.2	Style Requirements . . . . .	14
3.2	Usability and Humanity Requirements . . . . .	15
3.2.1	Ease of Use Requirements . . . . .	15
3.2.2	Personalization and Internationalization Requirements . . . . .	15
3.2.3	Learning Requirements . . . . .	15
3.2.4	Understandability and Politeness Requirements . . . . .	15

3.2.5	Accessibility Requirements . . . . .	16
3.3	Performance Requirements . . . . .	16
3.3.1	Speed and Latency Requirements . . . . .	16
3.3.2	Safety-Critical Requirements . . . . .	17
3.3.3	Precision or Accuracy Requirements . . . . .	17
3.3.4	Reliability and Availability Requirements . . . . .	17
3.3.5	Robustness or Fault-Tolerance Requirements . . . . .	17
3.3.6	Capacity Requirements . . . . .	17
3.3.7	Scalability or Extensibility Requirements . . . . .	17
3.3.8	Longevity Requirements . . . . .	18
3.4	Operational and Environmental Requirements . . . . .	18
3.4.1	Expected Physical Environment . . . . .	18
3.4.2	Requirements for Interfacing with Adjacent Systems . . . . .	18
3.4.3	Productization Requirements . . . . .	18
3.4.4	Release Requirements . . . . .	18
3.5	Maintainability and Support Requirements . . . . .	18
3.5.1	Maintenance Requirements . . . . .	18
3.5.2	Supportability Requirements . . . . .	19
3.5.3	Adaptability Requirements . . . . .	19
3.6	Security Requirements . . . . .	19
3.6.1	Access Requirements . . . . .	19
3.6.2	Integrity Requirements . . . . .	19
3.6.3	Privacy Requirements . . . . .	19
3.6.4	Audit Requirements . . . . .	19
3.6.5	Immunity Requirements . . . . .	19
3.7	Cultural Requirements . . . . .	20
3.8	Legal Requirements . . . . .	20
3.8.1	Compliance Requirements . . . . .	20
3.8.2	Standards Requirements . . . . .	20
3.9	Health and Safety Requirements . . . . .	20
<b>4</b>	<b>Project Issues</b>	<b>20</b>
4.1	Open Issues . . . . .	20
4.2	Off-the-Shelf Solutions . . . . .	21
4.2.1	Ready-Made Products . . . . .	21
4.2.2	Reusable Components . . . . .	21
4.2.3	Products That Can Be Copied . . . . .	21
4.3	New Problems . . . . .	21

4.3.1	Effects on the Current Environment . . . . .	21
4.3.2	Effects on the Installed Systems . . . . .	21
4.3.3	Potential User Problems . . . . .	21
4.3.4	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product . . . . .	21
4.3.5	Follow-Up Problems . . . . .	22
4.4	Tasks . . . . .	22
4.4.1	Project Planning . . . . .	22
4.4.2	Planning of the Development Phases . . . . .	22
4.5	Migration to the New Product . . . . .	22
4.5.1	Requirements for Migration to the New Product . . . . .	22
4.5.2	Data That Has to Be Modified or Translated for the New System . . . . .	23
4.6	Risks . . . . .	23
4.7	Costs . . . . .	23
4.8	User Documentation and Training . . . . .	23
4.8.1	User Documentation Requirements . . . . .	23
4.8.2	Training Requirements . . . . .	23
4.9	Waiting Room . . . . .	24
4.10	Ideas for Solutions . . . . .	24

## List of Tables

1	<b>Revision History</b> . . . . .	iii
2	<b>Business Event List for the Cat-and-Mouse Game</b> . . . . .	6

## List of Figures

1	Context Diagram for the Cat-and-Mouse Game . . . . .	5
---	--	---

Table 1: **Revision History**

Date	Version	Notes
Oct. 9, 2016	1.0	First Draft
Nov. 17, 2016	1.1	Made document more abstract

This document describes the requirements for the Cat-and-Mouse game. The template for the Software Requirements Specification (SRS) is a subset of the Volere template (?).

# **1 Project Drivers**

## **1.1 The Purpose of the Project**

### **1.1.1 Background of the Project**

This project's purpose is to create an online multiplayer game that is both entertaining and promotes teamwork between players. Our motivation for the project stems from the popularity and commercial success of games which emphasise positive player interaction and collaboration, such as Overwatch and League of Legends.

### **1.1.2 Goal of the Project**

The goal of this project is to create a multiplayer game which allows users to control characters which interact with a virtual environment, in order to complete various objectives. Such objectives should require effective strategic collaboration and teamwork in order to be accomplished. Collaboration should be possible through a network connection to a server, and the product should be usable on desktop computers running on the Microsoft Windows operating system.

## **1.2 The Stakeholders**

### **1.2.1 The Client**

This game will be developed for [Dr. Spencer Smith](#) and Dr. Wenbo He.

### **1.2.2 The Customers**

Our goal for this project would be to target people of all ages interested in playing an online strategic video game where they can develop teamwork, communication, and planning skills. As a video game, it is also meant for individuals who want to take a break with a few minutes of entertainment.

### 1.2.3 Other Stakeholders

Game testers will be necessary, as we will need their feedback to help us improve the game's overall experience.

## 1.3 Mandated Constraints

### 1.3.1 Solution Constraints

**Constraint 1:** Game must be created using a game engine.

**Rationale:** a game engine will allow us to develop the game faster as we do not have to create features such as shaders from scratch.

**Fit criterion:** The developers and customers confirm that the game is using the Unity 5 engine.

**Constraint 2:** The product will have network capabilities.

**Rationale:** A major part of our game is the multiplayer, therefore we need to be able to implement it.

**Fit criterion:** Multiplayer works. If the host leaves the game, it should not kick everyone else out of the game due to the dedicated server.

**Constraint 3:** Game will only be playable with computers that runs on a Windows Operating System from Windows 7 and onwards

**Rationale:** To cater to Windows Operating System users.

**Fit criterion:** Developers must confirm that the game runs on Windows Operating Systems (Windows 7 and onwards).

### 1.3.2 Implementation Environment of the Current System

The game will be written in an object-oriented language and created a game engine. 3D modeling software will be used to create 3D models for use in game.

### 1.3.3 Partner or Collaborative Applications

[Blender](#) for making the 3D models that will be used for characters and other objects in this game.

#### **1.3.4 Off-the-Shelf Software**

Off-the-Shelf Software may include Unity 5 game engine, Photon Unity Networking, and Blender.

#### **1.3.5 Anticipated Workplace Environment**

Users will be using our product at home on their Personal Computers. Other environments include at school, in internet cafes, and public locations that people use their laptops in. We anticipate that most people will play this game at home, as it is not intended to be a game for mobile devices. The environment should not affect the game's visual and audio cues.

#### **1.3.6 Schedule Constraints**

- Proof of Concept Plan, October 26th
- Test Plan Revision 0, November 2nd
- Proof of Concept Demonstratio, November 21st to November 25th
- Design Document Revision, January 11th
- Demonstration Revision 0, February 13rd to February 17th
- Users Guide Revision 0, March 1st
- Test Report Revision 0, March 22nd
- Final Demonstration (Revision 1), April 11th to April 27th
- Final Documentation (Revision 1), April 5th

#### **1.3.7 Budget Constraints**

N/A

#### **1.3.8 Enterprise Constraints**

N/A

## **1.4 Naming Conventions and Terminology**

"The Cat" will refer to both the cat character and the user playing as the cat. "The Mice" will refer to both the users playing as the mice and their characters.

## **1.5 Relevant Facts and Assumptions**

Naturally, we assume that people from all age groups and background are interested in playing a strategic online multiplayer video game. Since new video game titles are released every year, we can assume that there is always an interest in video games among a large section of the population.

As the project will be developed targeting the Windows PC, we will assume that most individuals own, use, or have access to obtaining the necessary components to run the game, be it hardware or software. This means that the users have proper internet access and hardware that meet the requirements.

Of course, our final assumption would be to assume that the project that we are developing, a real time, strategic, multiplayer online level up based Cat and Mouse game is original and has not already been created by other individuals.

# **2 Functional Requirements**

## **2.1 The Scope of the Work and the Product**



### 2.1.1 The Context of the Work

A context diagram for the Work is indicated in **Figure 1**. This indicates the adjacent systems which interact or *interface* with the Cat-and-Mouse Game.

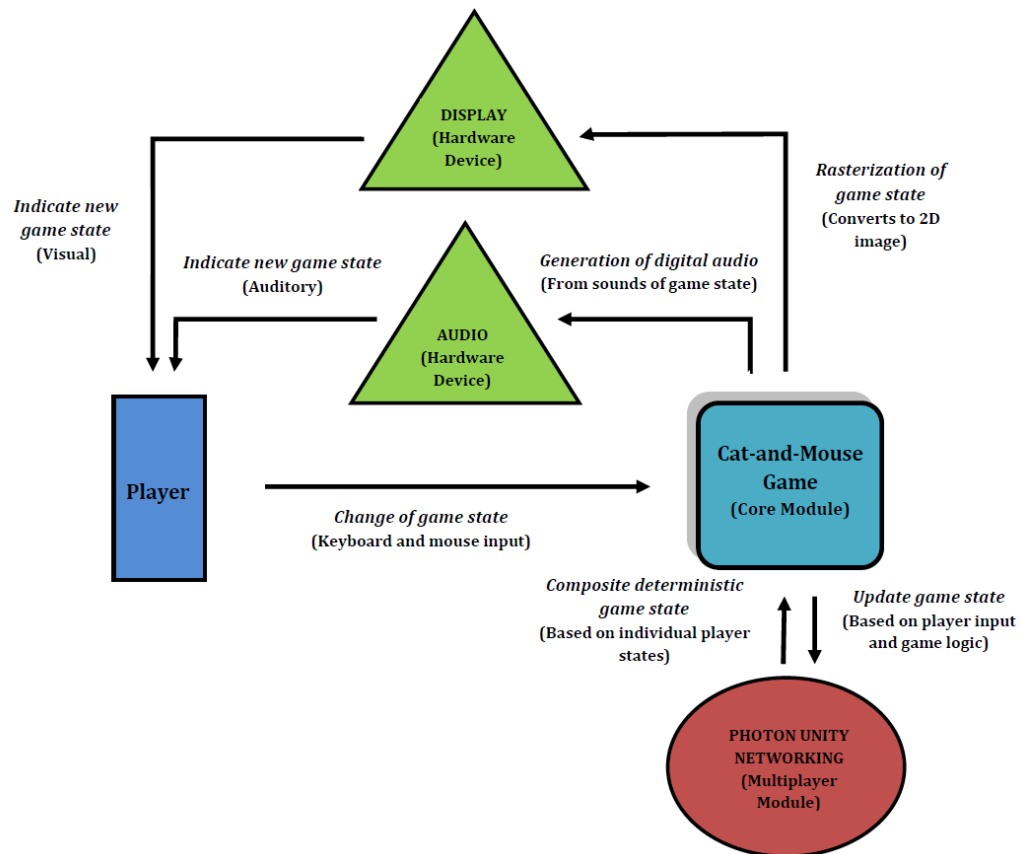


Figure 1: Context Diagram for the Cat-and-Mouse Game

### 2.1.2 Work Partitioning

A list displaying all business events to which the game responds is indicated in **Table 2**. These 'real-world' events represent the general operations allowed in the game and indicates the action(s), if any, which should be taken.

Table 2: **Business Event List for the Cat-and-Mouse Game**

Event Name	Input and Output	Summary of BUC
1. Player begins a new multiplayer session		A new multiplayer session is started on the server.
2. Player enters an existing multiplayer session	Change of game state (in); Update game state (out);	A new player is added to 'Lobby Screen' of the specified multiplayer session.
3. Player requests to start the Tutorial	Composite deterministic game state (in); Rasterization of game state (out);	A single player session is started on the server. The tutorial maps are loaded. The Player is assigned to a Cat or Mouse character (of their choice).
4. Players vote on the mode of gameplay	Generation of digital audio (out); Indicate new game state - Visual (out);	Each player is randomly assigned to a Team and a map is loaded. If the vote is tied, mode of gameplay is randomly selected.
5. A single Player in a multiplayer session requests to start the match	Indicate new game state - Auditory (out)	Displays 'Player Ready' notification for that Player.

6. All Players in a session request to start the match	All Players are assigned to a character on a Team randomly. The selected map and game mode are loaded.
7. Player leaves the match	The current match (if active) should continue. Players may be switched between Teams to rebalance.
8. Player Character is moved	The Player's character is moved within the map.
9. Player Character makes contact with the environment	Depending on the type of environment, will affect the character's state in varying ways.
10. Player Character is near an Item	The character is able to pick up the Item.
11. Player Character picks up Item	The Item is transferred into character's inventory.
12. Player Character drops Item	The Item is removed from character's inventory. May be picked up by other characters.
13. Player Character uses Item in inventory	Affects the Attributes, Skills and/or Items of the character.
14. Player Character makes contact with another Player Character from the same Team	Both players' movement is stopped.
15. Player Character makes contact with Monster	Will drain the Health Points (HP) of the Player Character. May also inflict other effects.

16. Player Character makes contact with another Player from the opposing Team	Both Players' movement is stopped. Other effects may take place dependent on each character's respective Skills and Attributes.
17. Player Character is near a Non-Playable Character (NPC)	Character can interact with the NPC to receive Tasks.
18. Player Character starts Task	Players provided with specific conditions which must be fulfilled.
19. Player Character finishes Task	Players granted enhanced Attribute(s), Skill(s) and/or Item(s).
20. Player Character attacks another Player Character from the same Team	Nothing occurs.
21. Player Character attacks another Player Character from the opposing Team	Will drain the Health Points (HP) of the opposing Player Character (dependent on Attributes and Skills). May also inflict other effects.
22. Player Character attacks Monster	Drains the Health Points (HP) of the Monster. May also inflict other effects.
23. Player Character beats Monster	Grants Player Character with enhanced Attribute(s), Skills(s) and/or Item(s).

24. Player Character uses Skill		Affects the attributes of the Player Character and/or the environment around them (including other characters). May use up Mana Points (MP).
25. Player Quits the game		Player is disconnected from the server and the game returns to the "Main Menu".
26. Monster attacks Player Character	Update game state (out); Composite deterministic	Drains the Health Points (HP) of the Player Character. May also inflict other effects.
27. Player Character's Health Points (HP) reaches zero	game state (in); Rasterization	The Player Character will die.
28. Player Character's Mana Points (MP) reaches zero	of game state (out); Generation of digital audio	The Player Character will be unable to use Skills which require MP.
29. Player Character dies	(out); Indicate new game state - Visual (out);	The Player Character will be returned to the "Lobby Screen".
30. All Player Characters on a Team are beaten.	Indicate new game state - Auditory (out)	The opposing Team is granted one Point. The current match will end. The Players will be returned to the "Lobby Screen" for the session.
31. Player disconnects from the server		The player will be returned to the "Main Menu". An error message is displayed.

32. All Players have left the multiplayer session		The session (and all connections) will be closed.
33. Player starts the Cat-and-Mouse Game	Change of game state (in); Rasterization of game state (out);	Displays the "Main Menu". Options to "Start New Multiplayer Session", "Start Tutorial", access "Game Options", and "Quit Game".
34. Player selects the "Options" Menu from the Main Menu	Generation of digital audio (out); Indicate new game state -	Allows the Player to adjust various settings for the game (eg. graphics, audio, gameplay, etc).
35. Player selects the "Quit Game" option from the Main Menu	Visual (out); Indicate new game state - Auditory (out)	Allows the Player to terminate the game.

### 2.1.3 Individual Product Use Cases

Ideally, the user would receive the game from an online source such as a game distribution platform such as Steam, where they would download the game and install it. Individually, the game cannot be played but when they join the game's online rooms they will be able to find other players also looking for games to join. The user will then be able to play the game with other people online.

## 2.2 Functional Requirements

**Requirement 1:** The Cat-and-Mouse Game should allow players to adjust in-game options (such as graphics, audio, the UI and gameplay settings) to suit their preferred experience and hardware.

**Rationale:** This will allow for the product to be accessible to a broader audience as well as a wider variety of hardware.

**Fit criterion:** Players have the ability to set the graphics fidelity in-game to match their computer's capabilities, toggle audio (background music, sound effects), customise input controls, as well as personalise the placement of various elements of the User Interface to their liking.

**Requirement 2:** The Cat-and-Mouse Game should allow players to join a Multiplayer Hub located on a server wherein they can interact and communicate with other players.

**Rationale:** This will allow Players to meet and greet their teammates for the upcoming Match and foster a sense of camaraderie. Players will also have the chance to decide together on a game mode and discuss strategy during the match.

**Fit criterion:** Players are able to connect successfully to the game server and communicate via text or voice with their teammates both in the Multiplayer Hub and while a match is in progress.

**Requirement 3:** The Cat-and-Mouse Game should allow players to collaboratively customise various aspects of the Match to suit their preferences.

**Rationale:** To appeal to a wide audience, players should have the freedom to adapt their gameplay experience according to their interests, skill level

and play style.

**Fit criterion:** Players are able to chose from a variety of in-game modes, maps and difficulty settings. Players should be able to agree on these settings through a vote-based system.

**Requirement 4:** The Cat-and-Mouse Game should randomly assign Players to play on either the 'Cat' or 'Mouse' team for every Match.

**Rationale:** This introduces a competitive element to the game, as well as fosters a sense of camaraderie, to encourage effective teamwork and collaboration between members fighting for the same cause.

**Fit criterion:** Players are able to chose from a variety of in-game modes, maps and difficulty settings. Players should be able to agree on these settings through a vote-based system.

**Requirement 5:** The Cat-and-Mouse Game should allow players to see and interact with each other through avatars in real-time to complete tasks and obectives within a virtual maze.

**Rationale:** This aspect enables key collaborative elements around which the core mechanics and gameplay have been designed.

**Fit criterion:** Individuals are able to see the position and status of, as well as interact with the virtual environment simultaneously with the other players participating in the match.

**Requirement 6:** The Cat-and-Mouse Game should allow players to move their characters around the map as the environment allows.

**Rationale:** Ensuring freedom of movement encourages players to be adventurous and creative in their interactions with not only the game objectives, but other fellow participants as well.

**Fit criterion:** Individuals are able to move to any point in the map without going through walls or solid objects.

**Requirement 7:** The Cat-and-Mouse Game should allow players to attack and deal damage to the corresponding players on the opposing team.

**Rationale:** This allows for game progression, and encourages team cooper-



ation by introducing a survival element to the gameplay.

**Fit criterion:** Players are able to wield various attack and defensive skills to cause damage to a player on the opposing team. The ability to utilise skills is regulated by a Mana Indicator, which must be kept filled. Remaining Health is also indicated and upon reaching a certain level, the player "dies" and is (temporarily) unable to participate further in the match.

**Requirement 8:** The Cat-and-Mouse Game should allow teams to gain points through accomplishing certain in-game tasks or objectives, depending on the selected game mode. Game modes include: capture the flag, escape the maze, and team deathmatch. killing opponents, capturing the flag, and killing cpu controlled monsters all give points.

**Rationale:** This allows for game progression, and a the competitive element encourages greater team cohesion in order to get ahead.

**Fit criterion:** Each team is granted points for accomplishing certain objectives over the course of a match. Dependent on the mode and parameters of the match, the game is considered "won" either when a Team attains the highest number of points after a certain period of time, a special condition has been fulfilled (e.g. capture the flag), or all members of the opposing Team have been eliminated.

**Requirement 9:** The Cat-and-Mouse Game should ensure that all players start on an even level playing field, so that there is a lower barrier of entry for new and casual players.

**Rationale:** The social nature of the game and strong focus of the game around teamwork and cooperation is such that the need for such skills to succeed should not be mitigated by the sheer amount of time spent in-game.

**Fit criterion:** Limited or no game state data is retained upon the termination of a multiplayer session. When starting a new game, previous player performance and accomplishments do not impact the skills or abilities which are granted to a player. Although both the 'Cat' and 'Mouse' Teams host varying Skills and Attributes, the game mechanic will be balanced so that neither team has a predetermined advantage over the other.

**Requirement 10:** The Cat-and-Mouse Game should allow for opportu-

nities for Players to enhance their Skills and Attributes through various in game interactions, such as by completing quests, defeating Monsters and collecting Items and Consumerables.

**Rationale:** The ability (and indeed, necessity) for a Player to work their way up from baseline in every session motivates individuals to fully explore all the possibilities which are present in every situation. Thus, all participants will be fully engaged in the game experience, and newcomers to the game will not be at a disadvantage in term of enjoyment (e.g. in terms of more 'experienced' players using the same skills over and over again in every match).

**Fit criterion:** All Players are able and encouraged to interact with the game environment (and other players) in new and innovative ways in every match, and a variety of different paths to the same goal will provide for high replay value for even the most seasoned players.

## 3 Non-Functional Requirements

### 3.1 Look and Feel Requirements

#### 3.1.1 Appearance Requirements

The game's interface shall be attractive to audiences of all ages and customers should be able to easily and quickly use the interface as it will be as intuitive as possible. We will use darker colours for the most part, avoiding flashy or warm colours in favour of colder colours to match the labyrinth theme and to help create a sense of tension.

#### 3.1.2 Style Requirements

We want this game to appear simple and intuitive, so that it does not intimidate certain age groups. However, it will have a mature look to it and a western art style, and together they will create a intense, dungeon-like feel. The players should feel a light sense of tension when playing, the Cat because he wants to find the Mice as soon as possible and the Mice because they do not want to be found.

## **3.2 Usability and Humanity Requirements**

### **3.2.1 Ease of Use Requirements**

The product will be easy to use for users of all ages, requiring basic English skills and use of a computer mouse and keyboard. The user should be able to learn the basics of the game very easily, namely the controls, core mechanics, and the objective of the game with merely going through a quick tutorial. All players who successfully complete the tutorial should be able to understand the basic components of the game.

### **3.2.2 Personalization and Internationalization Requirements**

The basic controls will be customizable, allowing the user to change certain hotkeys to whatever they please. The interface will be semi-customizable, allowing the user to move or resize items to their preference.

The first release of the product will only support English, but it will be very basic English that even individuals not fluent in English will be able to understand. Upon further development, we may include language packs such as French or Chinese.

### **3.2.3 Learning Requirements**

Even though the game will be designed to be easy to learn, and easy to get a basic feel of the game, it will be very hard to master. Playing this game at the highest level will require the user to memorize the most efficient paths or abilities, and make very few errors in terms of gameplay. The game should be difficult enough to inspire motivation to improve, but also easy enough to avoid discouraging users from playing. A benchmark will be such that a new user will have a general feel of the game after going through the tutorial, but becoming a master at the game will take hundreds of hours of gameplay. The game should have a lasting impression on the users in terms of fun and desire to improve, and make them want to come back and play again.

### **3.2.4 Understandability and Politeness Requirements**

Since the product is a video game, there will be concepts that are foreign to non video game players. However, the basic concept of the game is designed

around the game "cat and mouse" which most people should be familiar with; a game where the cats try to catch the mice. A tutorial will be included in the game, and will be written in a way that is easy to understand for anyone, avoiding difficult concepts or jargon when possible, and explaining them in detail when required.

### **3.2.5 Accessibility Requirements**

As a video game, the physical aspect cannot be easily substituted as the user will always be required to control their character using a keyboard and mouse, or a modified controller. The visual aspect also cannot be replaced as the user is required to be able to see everything that is going on in game, so fully blind individuals will unfortunately be unable to use this product. However, we can provide options for the colorblind by changing the in game colors to suit their needs. To suit the needs of the deaf, we can have subtitles of annotations for them to read.

## **3.3 Performance Requirements**

We hope to target players of all demographics, and we hope that they will be able to enjoy our game even if they do not have the latest computer systems. We will thus design our game to run smoothly on older computers as well as newer models.

### **3.3.1 Speed and Latency Requirements**

As the game operates at a real time level and is also multiplayer, we want each player to constantly have updates on the current game state. This would require information packets to be sent to each player multiple times every second, and each group of packets will contain information such as other the movements or actions of other players. The users will therefore be required to have a minimum internet connection speed of 1 Mbps.

Latency is also an important issue to consider in a real time game such as the one we are developing. Players need to be able to react immediately to what is going on in the game, so their latency should be as low as possible. We will aim to limit each player's ping to 100 ms or less, as any more than

this causes an obvious delay between the user's physical actions and their characters actions.

### **3.3.2 Safety-Critical Requirements**

The game will not have any extremely flashy effects that can cause seizures for certain individuals.

### **3.3.3 Precision or Accuracy Requirements**

N/A.

### **3.3.4 Reliability and Availability Requirements**

The server that a user hosts should not fail until said user closes it themselves. If the user leaves the game but does not close the server, then it should persist. The game itself should also not crash under any circumstance.

### **3.3.5 Robustness or Fault-Tolerance Requirements**

If a server is shut down unexpectedly, the game should kick players and display an error message to let them know what has happened. Any other unexpected errors should also display a message to the clients. If any player leaves the game, the game should continue as usual and switch a player to the other team if it is unbalanced.

### **3.3.6 Capacity Requirements**

Each game instance will host up to four players at a time, and information for each player will need to be stored on the server at any given time. For each user, the game will need to receive and hold data on each of the other three players as well as the current game state.

### **3.3.7 Scalability or Extensibility Requirements**

If the game increases in popularity then our servers will need to be upgraded to handle the increased traffic. The game itself will also be able to handle frequent patches to fix bugs or add new features.

### **3.3.8 Longevity Requirements**

As a computer video game, the expected lifetime of the product is as long as the user is interested in playing. A typical user can be expected to play for 1-2 hours, however if frequent updates are made to the game and the user constantly finds it fun, then the game's lifetime can be much longer.

## **3.4 Operational and Environmental Requirements**

### **3.4.1 Expected Physical Environment**

N/A.

### **3.4.2 Requirements for Interfacing with Adjacent Systems**

This game should work with Windows 7, 8, 8.1 and 10. It should also work with DirectX 11 and DirectX 9 as those are the graphics API that we will be using with Unity.

### **3.4.3 Productization Requirements**

This product will require users to have either DirectX 9 or DirectX 11 installed. The product will be distributed as a ZIP file, all users need to do is unzip it.

### **3.4.4 Release Requirements**

Ideally the game will be released on a game platform such as Steam, which allows users to receive updates with ease. However the first versions of the game will be released as a downloadable bundle where the user will have to download and install manually, then open the game by running the executable files.

## **3.5 Maintainability and Support Requirements**

### **3.5.1 Maintenance Requirements**

Patches and updates will be released as the creators see fit, in order to address bugs in the game, issues with balancing, or to release new game content.

### **3.5.2 Supportability Requirements**

The game will require minimal support, however if the user encounters issues with the game then they can contact the creators directly to look for a fix.

### **3.5.3 Adaptability Requirements**

Currently, we do not plan on porting the product to other platforms or environments.

## **3.6 Security Requirements**

### **3.6.1 Access Requirements**

The users will only be able to access the main game and the options that we see fit. The source code and server code will be hidden and restricted from the user.

### **3.6.2 Integrity Requirements**

Serious damage can be caused if an outsider gains access to the source code, so steps will be taken to prevent unauthorized access. Hacking into the servers can also cause issues with cheating and players gaining unfair advantages, so steps will also be taken to stop outside attacks.

### **3.6.3 Privacy Requirements**

This game will not have private information stored. The only information a user will have is their username, which will be stored on the client's machine.

### **3.6.4 Audit Requirements**

N/A.

### **3.6.5 Immunity Requirements**

Users having anti-virus software should be sufficient for protection against unauthorized programs like viruses.

### **3.7 Cultural Requirements**

The product will not have anything offensive or sensitive, culturally or religiously. The game will be in english and is not intended for a market outside of the US and Canada.

### **3.8 Legal Requirements**

#### **3.8.1 Compliance Requirements**

All of our assets and code will be created by ourselves or royalty free.

#### **3.8.2 Standards Requirements**

N/A.

### **3.9 Health and Safety Requirements**

This section is not in the original Volere template.

We do not want our game to use flashing imagery and effects that can potentially cause health issues such as seizures. This includes effects such as the game frequently flashing bright colors at the user, and use of certain patterns which can trigger an seizure in the affected demographics.

## **4 Project Issues**

### **4.1 Open Issues**

- Finding a good balance to the game is an issue. Because teams are asymmetrical (there is only 1 cat, while there is 3 mice), and each team has different abilities, it's hard to gauge how fair it is for one team which can affect how much fun the players have playing this game.
- Still need to think of an algorithm for map generation that makes the maps enjoyable to play on.



## **4.2 Off-the-Shelf Solutions**

### **4.2.1 Ready-Made Products**

N/A

### **4.2.2 Reusable Components**

N/A

### **4.2.3 Products That Can Be Copied**

There are asymmetrical player versus player games currently on the market, which if we could legally copy them, it could help with our game's potential balancing issue. A game that comes to mind is Evolve, which has one team being composed of a single player playing as a monster, and four hunters as the other team. Similar to ours, the teams have their own unique abilities and skillset.

## **4.3 New Problems**

### **4.3.1 Effects on the Current Environment**

N/A

### **4.3.2 Effects on the Installed Systems**

N/A

### **4.3.3 Potential User Problems**

N/A

### **4.3.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product**

The game might not run optimally depending on the hardware a user uses to run it.

### **4.3.5 Follow-Up Problems**

N/A

## **4.4 Tasks**

### **4.4.1 Project Planning**

- Produce sufficient and necessary documents such as requirement document, user guide, etc.
- Acquire necessary knowledge about tools we are using, and to familiarize ourselves with them.
- Develop a playable prototype of the game without all the features yet.
- Add on to the the prototype.

### **4.4.2 Planning of the Development Phases**

- Basic movement/controls of the characters.
- Map generation algorithm.
- Networking so multiplayer works.
- Creating the game's interfaces such as menus.
- Adding abilities and objectives to the game.
- Re-balancing the game after acquiring appropriate feedback from testers.
- Adding graphics and sounds that are not placeholders to polish the game's aesthetics.

## **4.5 Migration to the New Product**

### **4.5.1 Requirements for Migration to the New Product**

N/A

#### **4.5.2 Data That Has to Be Modified or Translated for the New System**

N/A

### **4.6 Risks**

- Certain abilities might be badly balanced, and thus make it unfair for one team.
- Procedurally generated maps using algorithms might affect balance of the game.
- Might not be able to find enough play testers to give satisfying feedback.
- Could run into trouble with networking and making it playable online.
- Time could be badly managed, and will need to cut down on features.

### **4.7 Costs**

There will be a time cost of 7 months for development of the game, but will cost no money as all tools used are free.

### **4.8 User Documentation and Training**

#### **4.8.1 User Documentation Requirements**

We will provide a User Documentation, as it is required as part of our capstone. There will also be a tutorial in game that informs players about the rules and how to play the game. This tutorial will cover topics such as joining a game, joining a team, movements, and objectives of the game.

#### **4.8.2 Training Requirements**

The game will require users to know basic English in order to go through the tutorial and play the game.

## **4.9 Waiting Room**

A requirement that might be done after the initial release of the product is an installation wizard. For now we think it is best to stick with an executable file for the game. Using an installation wizard also gives us the option to use more external dependencies, and thus more features if needed. any further requirements that might be added will be listed here as the document is revised.

## **4.10 Ideas for Solutions**

Currently, we do not have any ideas for solutions. This might change as we update this document.