THE

A

TEAM

Software Requirements Specification

Team Members

Jason Acevedo

Vincent Agriesti

Thomas Campus

Ming Lei

Sally Ng

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements for the Clue-Less system as described by the A-Team. This document will specify the requirements for the target Clue-Less system, as described in the A-Team Clue-Less Project Plan.

1.2 Scope

Clue-Less is a computerized version of the popular board game Clue. Clue-Less is being developed to allow players to enjoy the mechanics of the board game Clue in the digital world. See the A-Team Clue-Less Vision Document for more information about Clue-Less.

1.3 Glossary

- Command Line Interface (CLI) - A CLI is an interface that allows a user (human) of a system to interact with the system using a command line, usually a console. It typically only uses the keyboard.
- Graphical User Interface (GUI) - A GUI is an interface that allows a user (human) of a system to interact with the system using windows, icons, and graphics. It typically uses both the keyboard and mouse.
- Unified Modeling Language (UML) - A standard for descriptively mapping and sequencing entity relationships in a system architecture, primarily for the use of software engineering and surrounding activities.

1.4 References

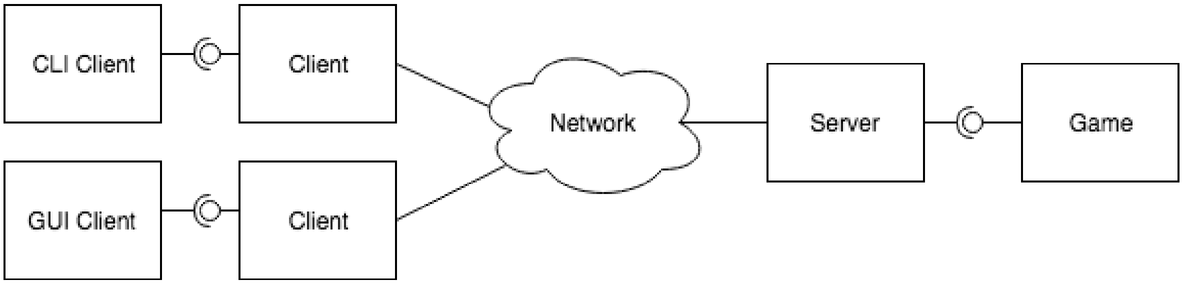
A-Team. (2018). *Clue-Less Project Plan.* Unpublished document.

1.5 Overview

This SRS document contains an overview of the system, use cases, traditional requirements, and supplementary specifications.

2. Overall System

2.1 System Overview



Surrounding the network, the system is composed of the Client and Server subsystems. These subsystems are responsible for the serialization and parsing of well-defined Clue-Less message formats, establishing/maintaining the TCP/IP network connections, and the passing of messages across these network connections in an ordered and consistent manner.

2.2 Use Cases

Use cases are used to document functional requirements in an easy to follow, story-like manner. Use cases list for Clue-Less are described below:

Label	**Use Case List**
a	Host Starts Server Application
b	Player Starts Client Application
c	Player Requests Game Start
d	Server Starts Game
e	Next Player Selection
f	Player Ends Turn
g	Player Enters Room
h	Player Leaves Room
i	Attempt To Disprove Suggestion
j	Player Makes A Suggestion
k	Player Makes An Accusation

a. Host Starts Server Application

Actors: Host
Pre-Conditions: Host has not started Server already.
Post-Conditions: The Clue-Less server is running and ready to accept client connections

Main Scenario / Flow of Events

1. The Use Case starts when the Host starts the Server application.
2. The Server waits for new Clients to connect to the Server.

Alternate Scenarios

- No network interface
    1. Server application displays a message indicating no suitable network interface available
    2. *Use Case Ends Here*

b. Player Starts Client Application

Actor: Player
Pre-Conditions: Host Starts Server Application

Main Scenario / Flow of Events

1. Player starts Client Application
2. Client application establishes a connection with the server.
3. Server sends a list of unclaimed suspects to Client.
4. Client presents unclaimed suspects list to Player
5. Player selects a suspect from the list

6. Client sends the requested suspect to Server
7. Server registers Player as provided suspect.

Alternate Scenario

- No network interface
  1. Server application displays a message indicating no suitable network interface available
  2. *Use Case Ends Here*
- Failed to connect to Server
  1. Server application displays a message indicating failure to connect to Server
  2. *Use Case* Ends Here
- Suspect No Longer Available
  1. Server sends (updated) unclaimed suspect list to Client
  2. Client presents updated list to Player
  3. *Continue with Main Scenario Step 5*

c. Player Requests Game Start

Actor: Player
Trigger: Player from User Interface Command
Pre-Conditions: Player Starts Client Application
Post-Conditions: The game has started, the envelope is populated, all cards are distributed, and play order is determined

Main Scenario / Flow of Events

1. User sends start message to System
2. System returns player cards
3. System notifies Player that game is started

Alternate Scenario

- Another user has not indicated they want to start.
  1. System does not start the game
  2. *Use Case Ends Here*

d. Server Starts Game

Actor: Server
Trigger: Player Request Game Start
Used Use Cases: Player Requests Game Start
Pre-Condition: Host starts server application

Main Scenario / Flow of Events

1. For each player connected to server, check for:
   - *Player Requests Game Start*
2. System shuffles cards.

3. System populates the confidential envelope.
4. System deals face up cards.
5. System deals remaining cards to the users.
6. System sets the first player as the first player associated with a Suspect in the ordered list:
    1) Scarlet
    2) Mustard
    3) White
    4) Green
    5) Peacock
    6) Plum
7. System alerts all users the game has started.
8. System notifies the first player that they are the active player.

Alternate Scenario

- Another player connected to server has not requested to start game.
- Not enough players connected and configured with Suspects.

e. Next Player Selection

Actor: Server
Trigger: Player Ends Turn, Server Starts Game
Pre-Conditions: Server Starts Game
Post-Conditions: A player has been selected to take a turn

Main Scenario / Flow of Events

1. Server attempts to select the next Suspect with an associated Player, starting from the last Player's Suspect that took a turn in the ordered list:
    1) Scarlet
    2) Mustard
    3) White
    4) Green
    5) Peacock
    6) Plum
2. Server sets the next Player with Suspect associated as the active player.

f. Player Ends Turn

Actor: Player
Trigger: Player User Interface Command
Used Use Cases: Next Player Selection
Pre-Condition: Next Player Selection

Main Scenario / Flow of Events

1. Player indicates they want to end their turn.
2. System performs: *Next Player Selection*

3. System notifies all Players the player has ended their turn and the next active player.
4. The next active Player is notified they are the active player.

g. Player Enters Room

Actor: Player
Trigger: Player User Interface Command
Pre-Conditions: Next Player Selection, Server Starts Game

Main Scenario / Flow of Events

1. Player chooses to move their suspect into an adjacent Room or Hallway via a direction:
    - NORTH
    - SOUTH
    - EAST
    - WEST
    - SECRET
2. System moves the current location of the Suspect
3. System updates the Suspect in the target location

Alternate Scenario

- Movement is invalid
    1. System reports the move is invalid
    2. *Use Case Ends Here*

h. Player Leaves Room

Actor: Player
Trigger: Player User Interface Command
Pre-Conditions: Next Player Selection
Post-Conditions: Player leaves room

Main Scenario / Flow of Events

1. Player chooses to move their suspect into an adjacent Room or Hallway via a direction:
    - NORTH
    - SOUTH
    - EAST
    - WEST
    - SECRET
2. System moves the current location of the Suspect
3. System updates the Suspect in the target location

Alternate Scenario

- Movement is invalid

1. System reports the move is invalid
2. *Use Case Ends Here*

- Hallway is occupied
  1. System reports the hallway is occupied.
  2. *Use Case Ends Here*

i. Attempt to Disprove Suggestion

Actor: Player
Trigger: Player Makes a Suggestion
Pre-Conditions: Player Makes a Suggestion

Main Scenario / Flow of Events

1. System indicates a suggestion was made with a message indicating the cards in the suggestion.
2. Player sends a single card in their hand to use to disprove the suggestion to the System

Alternate Scenario

- The Player has no card to disprove.
  1. Player indicates that they have no cards to disprove suggestion cards.
  2. *Use Case Ends Here*

j. Player Makes A Suggestion

Actor: Player
Trigger: Player User Interface Command
Used Use Cases: Attempt To Disprove Suggestion
Pre-Conditions: Next Player Selection, Player Enters Room
Post-Conditions: Player gains knowledge about what is in the confidential envelope. Suggested suspect and weapon are relocated to current Player's Room.

Main Scenario / Flow of Events

1. Player chooses to make a suggestion by specifying a weapon and a suspect
2. System verifies the Player is in a Room
3. System verifies the Player has not made another Suggestion since entering the room without leaving.
4. Suggested Suspect and Weapon are moved into the current Player's Room location.
5. For each opponent to the left of suggesting player in player turn order sequence:
   - *Attempt To Disprove Suggestion*
6. Player enters gained knowledge into Notebook

k. Player Makes Accusation

Actor: Player

Trigger: Player User Interface Command
Pre-Conditions: Next Player Selection, Server Starts Game
Post-Conditions: Player no longer able to move or suggest.

Main Scenario / Flow of Events

1. Player indicates an intention to accuse by selecting a weapon card, suspect card, and room card.
2. System notifies all opponents an accusation is being made.
3. System checks if the cards in the accusation match the cards in the sealed envelope.
4. System returns the result of the accusation to all opponents.
5. System indicates to the accuser the envelope cards
6. Player no longer able to move or suggest.

Alternate Scenario

- Player gets the accusation correct.
  1. System sends notification to all users the Player/Suspect won.
  2. System ends the game.
  3. Use Case Ends Here

2.3 Assumptions/Dependencies

- Low level network input/output (IO) will be handled by a third-party library.
- Terminal control will be handled by a third-party library.

3. Interface Requirements

3.1 User Interfaces

Clue-Less will implement a CLI as part of the minimal system. The CLI is used for both input and output to and from the system. The CLI will allow a user to play game of Clue-Less from start to finish with no other user interface. The CLI will interact with the user through a command line. Clue-Less will also implement a GUI as part of the target system. The GUI is used for both input and output to and from the system. The GUI will allow a user to play a game of Clue-Less from start to finish with no other user interface. The GUI will provide a more modern interface to the user than the CLI.

3.2 Communications Interfaces

Due to the nature of the system and the data it will contain and transmit, no security or encryption will be present. Clue-Less will utilize TCP/IP protocols with custom message formats to meet the requirements of the system.

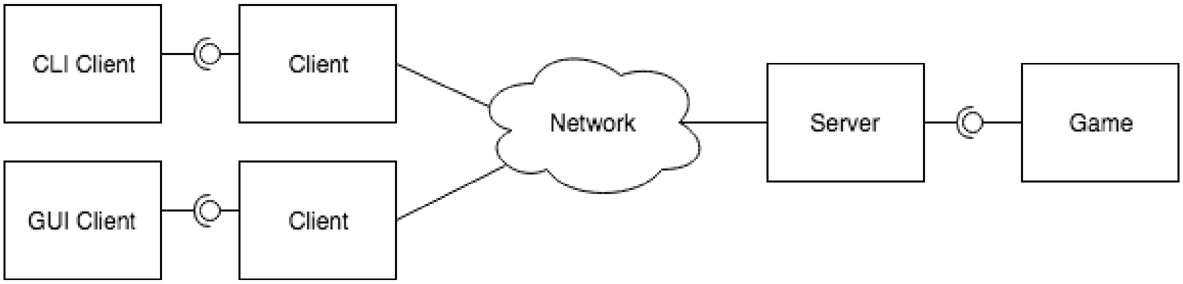
The messaging interface between the client and the server will be handled by a third-party dependency, ZeroMQ. ZeroMQ handles the management of network sockets and socket I/O in a separate thread. The Server subsystem will initialize itself as a ZeroMQ ROUTER, giving it the ability to address messages to a specific ZeroMQ client socket. The Client subsystem will initialize itself as a

ZeroMQ DEALER, giving it the ability to receive messages from the Server subsystem asynchronously.

All messages sent over the ZeroMQ socket are Serializable Java objects. Messages are serialized into a byte buffer and sent to the Client or Server. Messages are received in a byte buffer, parsed, and marshaled into a Java Object from the Client or Server.

4. Subsystems

4.1 Overview



4.2 Game Subsystem

4.2.1 Game Subsystem Description

- The Game subsystem maintains the state of the cards, game board, suspects, weapons, and player game play.
- The Game subsystem does not maintain the Notebook.
- The Game subsystem checks and verifies all client actions, except for Notebook actions.

4.2.2 Game Subsystem Functions

The Game subsystem provides the following functionality:

- Performs setup of the initial Clue-Less game state:
  - Setup of all Suspect locations
  - Setup of all Weapons
  - Setup of the (Confidential) envelope
  - Setup of the private Player Cards
  - Setup of the face up Cards
- Maintains authority over:
  - All card possesion
  - All board peice locations
  - Envelope contents
  - Player turn order
- Update all connected clients with the game status
- Tracking the state of a Suggestion sequence

- Player action-reduction due to bad accusation
- Tracking the winner due to a successful accusation

4.2.3 Game Subsystem Interfaces

When the Game Subsystem is created, the Server will expose an interface for broadcasting messages to all clients. The Game Subsystem will expose a single command interface and return a response message.

4.3 Server Subsystem

4.3.1 Server Subsystem Description

The Server subsystem manages low level network communications with the clients for passing well defined Clue-Less Message objects.

4.3.2 Server Subsystem Functions

The server provides the following functionality:

- Allow a Client subsystem object to connect to a shared TCP/IP address and port
- Parsing well defined request messages from clients
- Send well defined response messages to a single client
- Send well defined notification messages to all clients

4.3.3 Server Subsystem Interfaces

The Server subsystem encapsulates the low-level network communications with the Client subsystem. The Server is responsible for generating the Game instance and passing and interface for allowing the Game to broadcast well defined messages to all clients.

4.4 Client Subsystem

4.4.1 Client Subsystem Description

The Client subsystem is a shared package to be used by all user interfaces. It is responsible for establishing and maintaining connections to the Clue-Less server subsystem. The Client is responsible for sending and receiving well defined messages from the Game subsystem via the Server Subsystem.

4.4.2 Client Subsystem Functions

The CLI provides all input and output functions to and from the user. It provides the following functionality:

- Configure the client (suspect, username, server IP, server port)
- Serialization of request into well-defined request messages.
- Parsing of well-defined response messages
- Passing messages to the user interface's Event Handler
- Sending messages to the Server
- Disconnecting from the Server

4.4.3 Client Subsystem Interfaces

The Client is initialized by the user interface subsystem and is given an Event Handler interface to pass messages to on reception.

4.5 Command Line Interface (CLI) Subsystem

4.5.1 CLI Subsystem Description

The CLI is being used as part of the minimal system to consume input and provide output to and from the user.

4.5.2 CLI Subsystem Functions

The CLI provides all input and output functions to and from the user. It provides the following functionality:

- Configure the client (suspect, username, server IP, server port)
- Chat with all other clients
- Move the suspect
- Make a suggestion
- Make an accusation
- Disprove a suggestion
- Show the board status
- Show the user's cards
- Exiting the client

4.5.3 CLI Subsystem Interfaces

User requests are sent to the Clue-Less Game subsystem via the Client subsystem's communications mechanisms. Game responses are received by the client and sent to the CLI subsystem by its specialization of the Client subsystems event handler interface.

4.6 Graphical User Interface (GUI) Client Subsystem

4.6.1 GUI Subsystem Description

The GUI is being used as part of the target system to consume user input and provide user output via standard form widgets and mouse and keyboard interactions.

4.6.2 GUI Subsystem Functions

The GUI provides all input and output functions to and from the user. It provides the following functionality:

- Configure the client (suspect, username, server IP, server port)
- Chat with all other clients
- Move the suspect
- Make a suggestion
- Make an accusation
- Disprove a suggestion
- Show the board status

- Show the user's cards
- Exiting the client

4.6.3 GUI Subsystem Interfaces

User requests are sent to the Clue-Less Game subsystem via the Client subsystem's communications mechanisms.
Game responses are received by the client and sent to the GUI subsystem by its specialization of the Client subsystems event handler interface.

5. Document History

The Software Requirements Specification document may be updated during the project design process.

Version	Date	Notes
1	4/1/2018	Initial Release