Authors: **George, Linh, Troy, Kerim, Rasa**

Revision History

Date	Version	Description	Author
3/23/2020	Version 1.0	Initial release	Clue-Dunit Team

1. Document's Purpose and Objectives

The main purpose of this Software Requirements Specifications (SRS) is to present the essential elements of the Clue-less system such as:

- Software architecture of the system.
- Describing the flow of information and flow of processes (gameplay functions) by breaking down the software into different domains
- Using use-cases to document functional requirements with details about the user's input. and corresponding system's output
- The general design concept of the initial user interface.
- Listing the condition of the system in terms of non-functional requirements.

2. Glossary

- Server/Client: the basic roles of a network communication in which client sends requests in order to access the shareable resources and services, and server responses its availability on those resources and services.

- IPv4: the fourth version of Internet Protocol (IP) which is a 32-bit unique number assigned for network address.
- TCP/IP (Transmission Control Protocol/Internet Protocol): The Internet protocol that is allowed the communication between multiple computers.
- GUI (graphical user interface): the visual component of the software that users can interact with the system.
- Actor: an entity that acts on the system.

3. Software Architecture

Related “Software Architecture Document” maps Clue-Less game system between two main components Server and Client and defines how software will be structured.

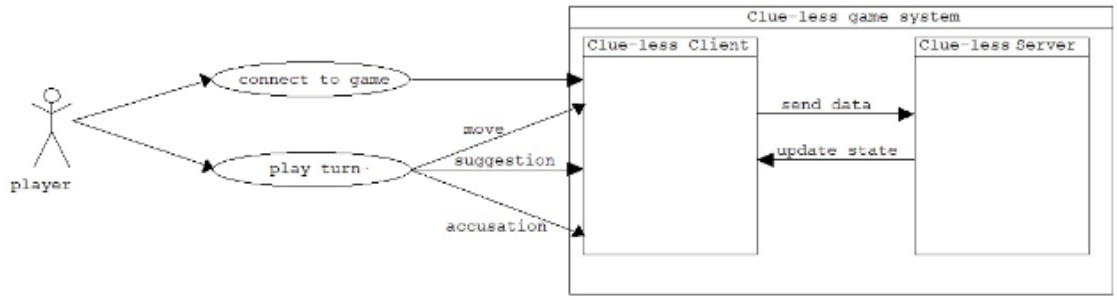


Figure 1: Top level use-case diagram for the Clue-less game system

4. Domains

4.1. Information Domain

The Clue-Less game will store data only at runtime on the game server, in other words the data will exist during the game but will be removed from the server system’s memory when the game is completed and shut down by the server. Similarly, the multiple clients will store runtime data of the game board and current game status until they are shut down. Data

communicated from the game client GUI to the game server will be transferred through a message packet of varying length with a header to identify the message size, command type, and message contents to the server and back to the client for updates to the game status.

A breakdown of the software itself is shown with the Unified Modeling Language (UML) class diagram below which describes the major classes, attributes, and functions in Clue-Dunit's Clue-Less client-server game. The server has functions and attributes that are used for setting up its environment on the local computer and for connecting to the internet. The client has functions that load the GUI, process input from the player, log data for debugging, and take in and process command line arguments.

Figure 2: A UML class diagram of the Clue-Less game.

4.2. Functional Domain

Functional domain section describes capabilities the game shall perform. The Clue-Less game system is functionally divided into 4 subsystems - Game Setup, Game Start, Game Play and Player Moves. The interactions between Player and these sub systems captured using use

cases. Detailed supplementary documentation can be found under (*4.2.2 Supplementary Specification for Use Cases)*.

4.2.1. Overview of Functional Domain

Subsystem	Description of the Subsystem	Use Cases	Increment
Game Setup	*Game Setup* Subsystem area covers setup capabilities that shall be present in order for users to install and remove the Clue-Less game.	UC01 Player installs game UC02 Player uninstalls game	Functionality for the Game Setup subsystem is part of **TARGET** increment.
Game Start	*Game Start* covers functionality related to forming a game	UC03 Player connects to game server UC04 Game Manager starts a game	Functionality of Game Start subsystem is part of **MINIMAL** increment
Game Play	*Game Play* subsystem defines the main capabilities needed to play a game	UC05 Player makes an accusation UC06 Player makes correct accusation UC07 Player makes incorrect accusation UC08 Player makes	Functionality of Game Start subsystem is part of **MINIMAL** increment

		correct suggestion UC09 Player makes incorrect suggestion UC10 Player disproves a suggestion UC11 Player leaves the game	
Player Moves	*Player Moves* subsystem defines functionality for player's moves on the board	UC12 Player making first move UC13 Player moves through passage UC14 Player moves through passage UC15 Player moves to hallway	Text only functionality of Player Moves subsystem is part of **MINIMAL** increment. GUI functionality of Player Moves subsystem is part of **TARGET** increment.
	Overview of Actors	**Player**: an active player in the game. **Game Manager**: a game server, responsible for managing the game and information being exchanged between players (clients) and the game (server). ● Managing suggestions/accusations/clues functionality ● Displaying status of the game (notifying when game starts/ends)	

	• Displaying status of the players (managing inputs and outputs, providing necessary feedback on the progress of game to all Players)

4.2.2. Supplementary Specification for Use Cases

The purpose of this document is to identify use cases for Clue-Less game and use this model to capture complete set of requirements on Clue-Less Game System.

4.2.3. Supplementary Specification for Functional Requirements

The purpose of this document is to document functional requirements for Clue-Less game that are not amenable to use cases specifications.

4.2.4. State chart Diagram

State chart diagram below (Figure 2) models lifetime of a player's Turn from beginning to end. Player's Turn will get terminated if Player loses, wins or forfeits the Turn. During each Turn, Player has three options for actions. He can Move and make a Suggestion (d), he can Forfeit a Turn (c) (if he can't move due to room being occupied), and at any time during his Turn he can make an Accusation (b).

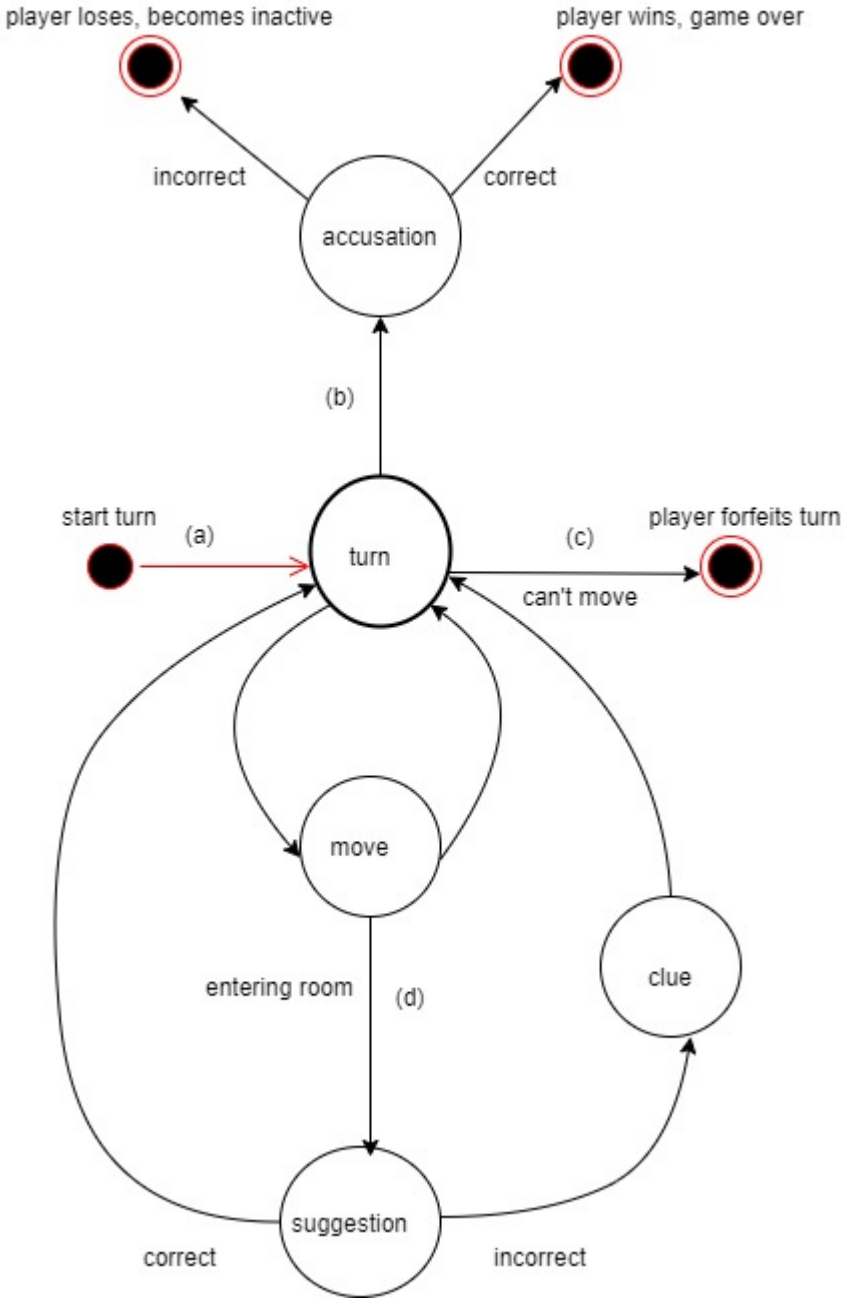


Figure 2: State Chart for Player's Turn

5. Interfaces

5.1. User Interfaces

The Clue-Less client user interface will consist of a GUI presented to each player that shows a 2D representation of the Clue-Less game board, character pieces, weapon pieces, checkboxes for checking off clues, buttons for moving the character and game pieces, one button to provide the game rules in a textbox, and a persistent textbox at the bottom of the GUI for relaying additional information such as indicating which player is currently taking their turn or the status of the game (guessing, accusation, player lost, etc.). The client will also feature a command line interface that allows a user to send commands to the program in order to modify some advanced input settings, if they desire, including:

“d”: setting debug level

“n”: disable login for testing

“l”: setting custom log location

“h”: displaying help and exiting

“v”: displaying version and information

“s”: define server IP address on command line

The Clue-Less server user interface will consist of a simple GUI that shows the status of the game and how many players are connected. Additionally, it will have a command line interface that allows a user to input settings for connecting or debugging the software and any messages displayed to the GUI will also be shown on the command line terminal interface. The CLI allows a user to start the server and broadcast their address as the local PC’s IP address, and will return messages from the server to the terminal in order to provide feedback to a user.

5.2. Hardware Interfaces

The Clue-Less game will support the following personal computer (PC) hardware based on the minimum specs of a Windows 10 computer for the client and server machines:

- A 1-gigahertz (GHz) processor
- 2 gigabytes (GB) of RAM
- 32 GB of hard drive space
- An integrated graphics card or a separate graphics card
- A display with a resolution of at least 800×600 pixels
- An ethernet or Wi-Fi connection to connect to the internet using an IPv4 internet address

Players will interact with the windowed Clue-Less GUI on their computers using a mouse or built-in trackpad in the case of laptops.

5.3. Software Interfaces

The Clue-Less client GUI will run on a Windows 10 operating system (Home edition or the lowest tier of Windows 10). The software itself is compiled by Microsoft Visual Studio Community 2017 and uses the Qt application framework, Qt version 5.12.3, for generating the GUI. The game server runs on a Linux Ubuntu version 16 operating system. The client communicates over the IP protocol with the server in order to play and update the game status for all clients connected to the game.

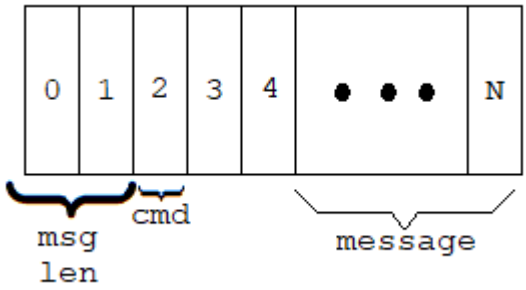
5.4. Communications Interfaces

The Clue-Less client and server will communicate over the internet using an Internet Protocol version 4 (IPv4) address in order to establish a connection from multiple clients to one server with a pre-specified IPv4 address coming from the machine it is running on. The initial release of the software will not have encryption or a way to reconnect games after a connection is lost, but the “dream” version of Clue-Dunit’s system is envisioned to have functionality that will enable basic encryption and the ability to reconnect to games if the connection is dropped.

5.5. Message Interfaces

5.5.1. Message format

The Clue-Less game message format will be as following:



Where bytes 0,1 represents the length of the message. Byte 3 represents command. Bytes 4 to n, command pay-load (exact format depends on the command).

5.5.2. Overview of messages

There will be messages from a client to the server, from the server to a specific client and broadcast messages to all clients from the server. The messages will be dependent on the command and documented as triggers in the use cases specifications.

Command	Message Trigger	Message Type
move	Player moves to the new location	Client to Server
suggest	Player makes suggestion	Client to Server
accuse	Player makes an accusation	Client to Server

connect	Player launches client	Client to Server
send_clue	Player sends the clue	Client to Server
leave_game	Player leaves game	Client to Server
receive_clue	Game Manager forwards clue	Server to Client
state_change	Game Manager notifies all Players Notification shall happen when one of the following state changes: - Game starts - Player inactive - Player moved - Suggestion is made - Player disapproves suggestion - Player is unable to disprove a suggestion - Game ends	Broadcast to all clients from the server

6. Non-Functional Requirements

- The system shall only delete a game and its associated data on a passed majority vote from the players involved in the game or after a duration of inactivity of two weeks.
- The system shall make a copy of each active game's state at the end of every day and store it until a new copy is made.
- For each active game, the system shall store a log of all major actions taken by the players and by the system itself.
- The system shall be able to host four games of up to six players each simultaneously and shall be able to handle updating all games without significant delay.
- The system shall be able to store data for the game states of up to eight games, deleting a completed game's data if a new game is started.
- The system shall be able to load and display the GUI, current games state, and available player actions within a minute of starting the application.
- The system shall be able to process all player actions and display the new game state or necessary errors to all players within a minute of a player submitting an action.
- The system shall be available with internet access in all regions where the source code repository is accessible.
- The system shall remain operational for at least 90% of each week, excluding unforeseen emergencies.
- The system shall communicate to all players of all current games when to expect system downtime during the current week at the start of each week.
- In the case where an error causes the game state to become invalid or corrupted, the system shall attempt to restore the game to the state held in the saved version of that game. This capability shall also be provided to the players of a game via a unanimous vote.
- The system shall modularize individual games so that a breaking error in one game does not impact the other games in-progress.
- The system shall provide a concise tutorial page for how to use the client and play the game available at every state of the GUI.

- The system documentation shall include a readme text file that shall detail the basic instructions for installing the game client and using it to play the Clue-Less game.

7. Implementation constraints

- The project will be built to run on only the Windows 10 operating system.
- The project will keep monetary cost at zero if at all possible; decisions that require payment will require unanimous agreement before implementation.
- The project will deliver functionality in stages at set deadlines, all explained within the team's Project Plan. Critically, the project will fully implement the functionality of the target version by 04/20/20.
- The project will be implemented in C++ using the Microsoft Visual Studio 2017 IDE to build the game client and associated server.
- The project will be written and documented in the English language.

8. Related Documents

These documents have been attached from previous and latest submissions.

- Clue-Less Project
- Clue-Dunit Project Plan
- Clue-Dunit Charter
- Clue-Dunit Vision Document
- Grading Rubric for Requirements Document
- Clue-Less Project Description
- Supplementary Specification for Use Cases
- Clue-Less Architecture Document
- Supplementary Specification for Functional Requirements