The address of this document is: http://www.perceptek.com.au/kteam/docs/srs.html

# Software Requirements Specification

# The MultiMahjong Project

# K-Team

## Team Members:

1. **Joanna Araminta (Project Manager)**
2. **Victor Leung (Technical Researcher)**
3. **Joel Brakey (Client Liaison Officer, Web Site Manager)**
4. **Michael Hart (Secretary, Web Site Manager, Backup Manager)**
5. **Dean Cortinovis (Technical Researcher)**
6. **Long Tang (Technical Researcher and Risk Manager)**

## Abstract:

**MultiMahjong is a single/multi player Mahjong computer game. This document describes the requirements for the MultiMahjong product.**

## Maintainer:

**Michael Hart (mwhart)**

## Version Control:

**$Revision: 2.1 $**
**$Date: 1999/08/16 12:00:44 $**

## Table Of Contents

# Table Of Figures

## 1. Introduction

This document formally states the requirements for the MultiMahjong product.

These requirements have been prioritised into three levels:

1. *Essential*
   The minimum set of requirements for the product to be accepted (see Section 8).
2. *Highly Desirable*
   Requirements that are considered to be likely inclusions to the product, time

permitting.

3. *Desirable*
   Requirements that are not likely to be added in this version, but should be considered for future modifications.

MultiMahjong is a product consisting of two programs - a MultiMahjongServer and a MultiMahjongClient. This Server/Client architecture will allow up to 4 players to play Mahjong against each other over a TCP/IP network. The MultiMahjongClient program will also allow 1 player to play in a stand-alone mode.

As any game of Mahjong requires 4 players to play, and there may not be 4 people available for a network game, the game will allow users to choose enough computer opponents to make up the required 4 players. In a single player game, the user will play against 3 computer opponents.

To play the game, users will use the MultiMahjongClient. The MultiMahjongServer is to reside on a TCP/IP server and will communicate with MultiMahjongClients.

The client requires the product for commercial purposes. The MultiMahjongClient program is to be sold to potential users (see Section 5.1.1) and the MultiMahjongServer is to initially reside on a server owned or operated by the client.

The client for this project is:

Steve Goschnick, Managing Director
Solid Software Pty Ltd
Level 3, Bouverie Street,
Carlton VIC, 3053
Ph: 03 9344 9322, 03 9344 0154
E-Mail: gosh@solidsoftware.com.au, gosh@cs.mu.oz.au

Our team for the project is called K-Team and consists of:

| | |
|---|---|
| Joanna Araminta (jiar) | Ph: 9889 4423 (Project Manager) |
| Victor Leung (vhle) | Ph: 9706 1560 |
| Joel Brakey (jebr) | Ph: 9859 6038 |
| Michael Hart (mwhart) | Ph: 9859 5419 |
| Dean Cortinovis (dcort) | Ph: 9798 2684 |
| Long Tang (lqkt) | Ph: 9540 8992 |

The supervisor for the project is:

Anthony Senyard (anthls) Ph (W): 9344 1940
Ph (H): 9417 2839

This document contains many references to the rules of Mahjong, specifically the Chinese rules of Mahjong. It is assumed that the reader of this document is familiar with these rules as many of the requirements are Mahjong specific. These rules can be found in the following book:

Carkner, K.J. "How to play Mah Jong",
1993, Penguin Books Australia Ltd.

## 2. The Existing Logical System

The client has no existing system for MultiMahjong.

## 3. The Proposed Logical System

As this is only a *suggestion* for the proposed logical system, all requirements mentioned in this section are Level 2 requirements.

### 3.1 Data Flow Diagrams (DFDs)

The following diagrams show the flow of data between the entities of the MultiMahjong system:

**Figure 3.1.1 - Level 0 DFD of the MultiMahjong System**

**Figure 3.1.2 - Level 1 DFD of the MultiMahjong System**

Note that these diagrams only describe the top level processes and the data that flows between them. They do not describe the functions of the MultiMahjongClient and MultiMahjongServer programs as the processes are shared between both programs. The functionality of the MultiMahjongClient and MultiMahjongServer programs is described in more detail in Section 4.

### 3.2 Data Dictionary

*name:* **player action**
*aliases:* user input
*where/how used:* **process player action** (input)
*description:*
**player action = [mouse state | keystroke]**
**mouse state = co-ordinate + button state**
**co-ordinate = xval + yval**
**xval = *horizontal distance (in pixels)***
**yval = *vertical distance (in pixels)***
**keystroke = *character from keyboard***

*name:* **pref. data**
*aliases:* preferences data, prefs
*where/how used:* **process player action** (input and output)
**preferences** (input and output)
*description:*
**preference data = [current player's name | icon | limit for winning hand | sound on/off]**
**current player's name = *any 30 character unicode-based string***
**icon = *colour image in a JPEG or GIF formatted file***
**limit for winning hand = *6 digit number***
**sound on/off = *boolean***

*name:* **screen changes**
*aliases:*
*where/how used:* **update screen** (input)
*description:*
**screen changes = ["display preferences window" command |**
**"update icon" command |**
**"update name" command |**
**"update game area" command |**
**"display action button" command |**
**"updated score" command]**
**"display preferences window" command = *content to be decided in SDD***
**"updated icon" command = *content to be decided in SDD***
**"updated name" command = *content to be decided in SDD***
**"updated game area" command = *content to be decided in SDD***
**"display action button" command = *content to be decided in SDD***
**"updated score" command = *content to be decided in SDD***

*name:* **game data**
*aliases:*
*where/how used:* **process player action** (input and output)
**handle game** (input and output)
**determine computer opponent action** (input and output)
**game info** (input and output)
*description:*
**game data = [tile position | current round |**
**first Wind of the Round | current Wind of the Round |**
**current player]**
**tile position = [The Wall | Discard | Dead Tile | Exposed Set |**
**Revealed Kong]**
**Exposed Set = [Exposed Kong | Exposed Pung | Exposed Chow]**
**current round = *1 digit number***
**first Wind of the Round = *content to be decided in SDD, will be**
**player ID***
**current Wind of the Round = *content to be decided in SDD,**
**will be player ID***
**current player = *content to be decided in SDD, will be player ID***

*name:* **game action**
*aliases:*
*where/how used:* **process player action** (output)
**handle game** (input and output)
**determine computer opponent action** (input and output)
*description:*
**game action = [discarded tile | picked up tile | exposed tile]**
**discarded tile = *tile class (tile class to be decided in SDD)***
**picked up tile = *tile class (tile class to be decided in SDD)***
**revealed tile = *tile class (tile class to be decided in SDD)***

*name:* **graphic data**
*aliases:*
*where/how used:* **update screen** (input)
**graphics** (output)
*description:*
**graphic data = *colour image file***

*name:* **graphic request**
*aliases:*
*where/how used:* **graphics** (input)
**update screen** (output)
*description:*
**graphic request = *command to be determined in SDD***

*name:* **screen image**
*aliases:*
*where/how used:* **player** (input)
**update screen** (output)

*description:*
**screen image = *what appears on screen***

## 4. Functional Requirements

This section states the requirements that relate to the functionality of the MultiMahjong system. Each requirement has been prioritised according to the levels set out in the Introduction (see Section 1). Requirements that relate to what is displayed on the user's screen are defined in Section 6.

Note that for single player games, the MultiMahjongServer will not be required and the player need not be connected to a TCP/IP network. Although many requirements mention that the MultiMahjongClient will send certain information to the MultiMahjongServer, in the single player game, this is not the case. In a single player game, the MultiMahjongClient will intercept this information and process it internally. The reference to the MultiMahjongServer is made to reduce duplicate requirements.

### 4.1 The MultiMahjongServer

*Level 1 Requirements:*

**4.1.1** The MultiMahjongServer will be designed to reside on a central computer (server) that is connected to a TCP/IP network and has a unique IP address.

**4.1.2** The MultiMahjongServer will allow connections from MultiMahjongClients and communicate with them using IP.

**4.1.3** The MultiMahjongServer will serve as a node for MultiMahjongClients. That is, it will be able to relay data received from one MultiMahjongClient to other MultiMahjongClients that request it.

*Level 2 Requirements:*

**4.1.4** The MultiMahjongServer will automate the game initialisation functions (as opposed to the MultiMahjongClient doing them), such as: randomising tile order, randomising seating position and storing the "Game Info" database (see Figure 3.1.2).

**4.1.5** The MultiMahjongServer will contain a High Scores list with the names and scores of the top MultiMahjong players.

*Level 3 Requirements:*

**4.1.6** The MultiMahjongServer will allow players on MultiMahjongClient programs to chat with each other in real time.

**4.1.7** The MultiMahjongServer will allow players on MultiMahjongClient programs to modify their own names and icons during gameplay (as opposed to only at the beginning).

**4.1.8** The MultiMahjongServer will support an architecture whereby all Computer Opponents (CO) required will reside on it instead of on the MultiMahjongClient programs.

**4.1.9** The MultiMahjongServer will have a graphical user interface with which the administrator of the server can get log information and change game settings.

**4.1.10** The MultiMahjongServer will be able to save preferences to a file and read from that file at start-up.

Note that all requirements hereafter are concerned with the MultiMahjongClient program.

### 4.2 Beginning the Game

*Level 1 Requirements:*

**4.2.1** When the user begins the MultiMahjongClient program, they will be able to choose to create a new multi player game, to create a new single player game, to join an existing multi player game, or to quit the program. These options will be available at any stage while the program is running as long as there is no game currently being played by the user (see Section 4.4 for more details about quitting).

**4.2.2** When a user creates a new multi player game, they must:

**4.2.2.1** Enter their name.

**4.2.2.2** Choose an icon from a predetermined list.

**4.2.2.3** Decide the number of human and computer opponents they want. As any game of Mahjong requires 4 players, the MultiMahjongClient will limit the user to choosing 4 players in total.

**4.2.2.4** Set the score limit for winning a hand of Mahjong.

**4.2.2.5** When all the necessary human players have joined the game, the user may choose to begin the game.

**4.2.3** The MultiMahjongClient must send this game initialisation information to the MultiMahjongServer so that the MultiMahjongServer can create a new game.

**4.2.4** The processing for any Computer Opponents (CO) (see Section 4.5 for more detail) will be done by the MultiMahjongClient program. In a single player game, the MultiMahjongClient will need to process for 3 COs. In a multi player game with 2 human players, each MultiMahjongClient will support 1 CO. In a multi player game with 3 human players, only 1 of the 3 MultiMahjongClients will support a CO. Obviously, in a multi player game with 4 human players, no CO is required.

**4.2.5** When a user creates a new single player game, they must:

**4.2.5.1** Enter their name.

**4.2.5.2** Choose an icon from a predetermined list.

**4.2.5.3** Set the score limit for winning a hand of Mahjong.

**4.2.6** When a user selects the option to join an existing game, the MultiMahjongClient will retrieve a list of any games that still require players from the MultiMahjongServer.

**4.2.7** When a user wishes to join a multi player game, they must:

**4.2.7.1** Enter their name.

**4.2.7.2** Choose an icon from a predetermined list.

**4.2.7.3** Choose which of the available games they wish to join.

**4.2.8** The MultiMahjongClient must send this join information to the MultiMahjongServer.

**4.2.9** In a multi player game, each MultiMahjongClient must retrieve the following data from the MultiMahjongServer:

**4.2.9.1** Whether it must support a CO.

**4.2.9.2** The score limit for a hand.

**4.2.9.3** All players' names, icons and seating positions.

**4.2.9.4** The tiles of the human and CO (optional) player supported on the MultiMahjongClient.

*Level 2 Requirements:*

**4.2.10** The list of un-started games that the MultiMahjongClient fetches from the MultiMahjongServer will be displayed in such a way that users can see the following information about each game:

**4.2.10.1** The name and icon of the player who created the game.

**4.2.10.2** The score limit for winning a hand.

**4.2.10.3** The number of places that still need to be filled before the game can commence.

**4.2.11** The MultiMahjongClient will save the user preferences in a file and read them in when the program begins.

*Level 3 Requirements:*

**4.2.12** Users who create a new game will be able to change the ability level of any CO involved in the game. There will be 3 levels available: Beginner, Intermediate and Advanced.

**4.2.13** Users who create a new game will be able to choose to play different variations of Mahjong rules other than those described in Section 4.3.1. The variations will include the Cleared Hand variation and the Japanese variation (see the rulebook for more detail).

**4.2.14** Any user who creates or joins a game will be able to upload their own JPEG or GIF image that they wish to have displayed above their name, if they do not wish to use a preset icon.

### 4.3 Playing the Game

*Level 1 Requirements:*

**4.3.1** The MultiMahjongClient must only allow players to make moves according to the Chinese rules of Mahjong as described in the rulebook, "How to Play Mah Jong".

**4.3.2** The MultiMahjongClient will inform the user whose turn it is to pick up a tile.

**4.3.3** The MultiMahjongClient will retrieve other players' changes in the game situation from the MultiMahjongServer. The user's screen is then updated to display these changes. The following game changes are displayed:

**4.3.3.1** Another player picks up a tile from the wall.

**4.3.3.2** Another player exposes/reveals a set of tiles.

**4.3.3.3** Another player discards a tile.

**4.3.4** If the configurations of: the current discard, the tiles in a user's hand, and the player whose turn it is, would allow the user to Chow/Pung/Kong/Mahjong under the Chinese rules of Mahjong, the MultiMahjongClient will inform the user that they can do so. If they choose to take this action, then the MultiMahjongClient must send this information to the MultiMahjongServer.

**4.3.5** User's will be able to pick up a tile from the wall if it is their turn. If they choose to pick up, the MultiMahjongClient must send this information to the MultiMahjongServer.

**4.3.6** User's will be able to discard a tile if they have picked one up. When they choose to discard, the MultiMahjongClient must send this information to the MultiMahjongServer.

**4.3.7** The MultiMahjongClient must inform the user if another player is fishing.

**4.3.8** The MultiMahjongClient must inform the user if another player goes Mahjong. The wind of the round is also updated (if required).

**4.3.9** The MultiMahjongClient must inform the user if the hand has ended in a draw.

*Level 2 Requirements:*

**4.3.10** The MultiMahjongClient will keep the user's tiles in order.

**4.3.11** The MultiMahjongClient will play sound effects associated with certain game actions.

**4.3.12** Users will be able to turn the sound on or off at any stage during the game.

**4.3.13** Users will have access to the User's Documentation online (see Section 7.2) at any stage during the game.

**4.3.14** Users will be able to view the High Scores list that is located on the MultiMahjongServer.

**4.3.15** The MultiMahjongClient will automate the picking up of a tile from the wall if the user has no other option.

*Level 3 Requirements:*

**4.3.16** User's will be able to Undo their last move if playing in single player mode.

**4.3.17** User's will be able to Save their game if playing in single player mode.

**4.3.18** User's will be able to chat with each other in real time.

**4.3.19** User's will be able to change their icons/pictures and names during the game.

### 4.4 Ending the Game

*Level 1 Requirements:*

**4.4.1** The game will end when all 4 rounds have been played or if the user chooses to end the game.

**4.4.2** The user will be able to end the game at any stage of the game. The user will then be able to quit the program or begin/join a new game.

**4.4.3** The MultiMahjongServer will inform all MultiMahjongClients connected to the game that the game has ended. Users of any MultiMahjongClient previously connected to the game will have the choice to quit the program or to begin/join a new game.

*Level 2 Requirements:*

**4.4.4** If a user leaves the game prematurely, a new CO is created on another user's machine to fill their place. This CO will take over the user's current position and circumstance within the game. This CO will play at an ability level as defined in the Level 2 Requirements for the CO (see Section 4.5.6).

*Level 3 Requirements:*

**4.4.5** If a user leaves the game prematurely, a new CO is created on another user's machine to fill their place. The user who created the game will decide at what level (see Section 4.5.8) the CO will play at.

### 4.5 Computer Opponent (CO)

*Level 1 Requirements:*

**4.5.1** The CO must play moves according to the Chinese rules of Mahjong.

**4.5.2** The CO will not have access to the data containing the faces of any tiles in the wall or any concealed tiles in any other players' hands.

**4.5.3** The CO will have access to the number of tiles in the wall or in a player's hand.

**4.5.4** The CO will be able to read any player's exposed or revealed hands.

**4.5.5** The CO will be able to pick up tiles off the wall, discard tiles, reveal Kongs, and Chow/Pung/Kong/Mahjong.

*Level 2 Requirements:*

**4.5.6** The CO will be able to determine what tiles have not been played according to the tiles that have already been discarded, exposed or revealed.

**4.5.7** The CO will be able to recognise "almost finished" sets of tiles (i.e. 2 tiles in a sequence is an almost finished Chow) and will not discard tiles in such sets. This is to give the CO a basic level of strategy.

*Level 3 Requirements:*

**4.5.8** The CO will be able to play at 3 different ability levels: Beginner, Intermediate and Advanced.

**4.5.9** The CO will be able to perform look ahead algorithms to determine its best move in the Intermediate and Advanced levels.

**4.5.10** The CO will be able to perform calculations at any stage of the game.

**4.5.11** The CO will be able to play any version of the rules that are specified by the user who creates the game.

**4.5.12** There will be CO's of different playing styles that the user can choose from.

## 5. Non-Functional Requirements

This section states all the requirements of the MultiMahjong system that are not related to the functionality of the MultiMahjong system.

### 5.1 Nature of the Users

### 5.1.1 MultiMahjongClient

*Level 1 Requirements:*

**5.1.1.1** The potential users of the MultiMahjongClient product will be anyone who enjoys playing or is interested in the game of Mahjong.

**5.1.1.2** The product is aimed at users who have a basic knowledge how to operate a PC, Macintosh or Unix box (see Section 5.3.1).

**5.1.1.3** Users who wish to play multi player Mahjong will not need to have advanced knowledge of TCP/IP networking. It is assumed however, that the computer has been configured correctly to connect to a TCP/IP network.

**5.1.1.4** Prior knowledge of Mahjong rules is not necessary, as the product will be designed for both players with some or no experience.

### 5.1.2 MultiMahjongServer

*Level 1 Requirements:*

**5.1.2.1** The user of the MultiMahjongServer will most likely have general knowledge of a Web Server environment.

**5.1.2.2** The user will need to know how to set-up and operate a TCP/IP server.

**5.1.2.3** The user will understand the basic principles of networking and know any networking information associated with the server the MultiMahjongServer program is run on (hostname and IP address).

## 5.2 Error Handling

Every error that occurs during program execution can be classified into two types - fatal and nonfatal errors. The distinction between the two is that the program is unable to continue to execute upon encountering a fatal error.

### 5.2.1 Nonfatal errors:

*Level 1 Requirements:*

**5.2.1.1** If a nonfatal error occurs, a dialogue box reporting the nature of error will be displayed.

**5.2.1.2** As the error is classified 'nonfatal', the user will be able to continue the current game.

*Level 2 Requirements:*

**5.2.1.3** An option to access the online troubleshooting guide (Section 7.2.4) is included in the error dialogue box.

### 5.2.2 Fatal errors:

*Level 1 Requirements:*

**5.2.2.1** If a fatal error occurs, a dialogue box reporting the nature of error will be displayed.

**5.2.1.2** The only option to the user is to quit the game. All processes will be cleaned up (if possible) and the program execution will terminate.

## 5.3 Implementation Constraints

*Level 1 Requirements:*

**5.3.1** The MultiMahjong system is to be written using JDK 1.2 (the Java development kit from Sun Microsystems).

**5.3.2** The coding standard will be the same as that suggested by Sun Microsystems. This document is available in PDF in our repository at:
MultiMahjong/doc/SQAP/CodeConventions.pdf
or online at:
ftp://ftp.javasoft.com/docs/codeconv/CodeConventions.pdf

*Level 3 Requirements:*

**5.3.3** Multiple languages will be supported using Java's Unicode standard.

## 5.4 Hardware Constraints

*Level 1 Requirements:*

**5.4.1** The software will run on any machine that is capable of running a Java Virtual Machine that supports JDK 1.2 (see Section 5.3). These machines include:

**5.4.1.1** A PC operated under Windows 95/98/NT

**5.4.1.2** A Macintosh operated under OS 8

**5.4.1.3** A Unix box operated under Solaris 2.6 or Solaris 7

*Level 2 Requirements:*

**5.4.2** The minimum system requirements for the product to operate under are:

**5.4.2.1** 100 MHz processor

**5.4.2.2** 32 Mb of RAM

**5.4.2.3** 10 MB of uncompressed hard disk

**5.4.2.4** 16-bit video card with minimum resolution of 800 x 600

**5.4.2.5** 28.8 Kbps modem and a TCP/IP connection to the Internet, or a TCP/IP LAN connection. For multi player game only.

### 5.5 Performance Constraints

*Level 1 Requirements:*

**5.5.1** The CO must respond within 1 minute in both single and multi player modes.

**5.5.2** The MultiMahjongServer must be able to handle requests in multi player mode of up to 10 simultaneous games (max 40 players).

**5.5.3** The MultiMahjongClient must be able to calculate possible Chow/Pung/Kong/Mahjong moves (see Section 4.3.4) within 5 seconds of being informed of the last discard.

### 5.6 Security Requirements

As the data being exchanged between machines does not contain any private information and is not capable of undermining the system integrity of the computers connected to the MultiMahjongServer, encryption and protection of data are not needed.

## 6. User Interface Requirements

This section states all the requirements of the MultiMahjong system that are related to what the user sees and how the user interacts with the MultiMahjong system.

### 6.1 User Interface - MultiMahjongServer

*Level 1 Requirements:*

**6.1.1** The MultiMahjongServer will have no graphical user interface and will be run using a command prompt.

*Level 2 Requirements:*

**6.1.2** The MultiMahjongServer will have a graphical user interface with which the administrator of the server can get log information and change game settings. No requirements for this graphical user interface are set in this document.

### 6.2 User Interface - MultiMahjongClient

*Level 1 Requirements:*

**6.2.1** The MultiMahjongClient will have a graphical user interface that the user will be able to interact with using a mouse and a keyboard. The details of this graphical user interface are described in Section 6.3. Where it is appropriate, keyboard shortcuts will be provided for mouse based operations.

*Level 2 Requirements:*

**6.2.2** The MultiMahjongClient will have sound effects associated with the game actions.

*Level 3 Requirements:*

**6.2.3** The MultiMahjongClient will have animation also associated with game actions.

### 6.3 Graphical User Interface - MultiMahjongClient

The graphical user interface (GUI) described below only applies to the MultiMahjongClient program. As the validity of these requirements will not be determined until the design phase, most of them are Level 2 or Level 3 requirements. As described in Section 6.2.1, the *existence* of the GUI is a Level 1 requirement, however, the *detail* of the GUI is not necessarily so.

*Level 1 Requirements:*

**6.3.1** The GUI is to be designed so that the user can perform all of the Level 1 requirements set out in Section 4.

**6.3.2** The GUI is to be contained within a screen resolution of 800 x 600 pixels.

**6.3.3** All graphics will be in 16 bit colour.

**6.3.4** There will be a Main Window that will contain the following game elements:

- **6.3.4.1** The faces of the current user's tiles
- **6.3.4.2** The wall with the remaining tiles
- **6.3.4.3** The backs of the other player's tiles
- **6.3.4.4** Any revealed or exposed tiles from any player
- **6.3.4.5** The last discarded tile
- **6.3.4.6** Buttons to allow the user to pick up a tile, to discard a tile, to Chow/Pung/Kong/Mahjong and to reveal a Kong.
- **6.3.4.7** A button will be made inactive (typically greyed out) if the user is not allowed to perform the associated activity.

*Level 2 Requirements:*

**6.3.5** The Main Window will also contain a tabbed frame (with 3 tabs) to display the following items:

**6.3.5.1** Tab1: Player's name

**6.3.5.2** Tab1: Player's icon

**6.3.5.3** Tab1: Wind of the round

**6.3.5.4** Tab2: Score

**6.3.5.5** Tab2: Limit of the hand

**6.3.5.6** Tab3: Preferences

**6.3.6** The options at the beginning of the program, as described in Sections 4.2.1, 4.2.2, 4.2.5 & 4.2.7, will be presented to the user using a dialogue box with buttons associated with each option.

**6.3.7** The list of available games, as described in Section 4.2.7, will be presented to a user in a dialogue box with a scroll bar and selectable list items so that the user can select the game they wish to join.

**6.3.8** The Main Window will show the dead tiles.

*Level 3 Requirements:*

**6.3.9** When a button becomes active, it will use animation to get the user's attention.

**6.3.10** When a player performs a Chow/Pung/Kong/Mahjong, the tiles involved will be animated.

**6.3.11** The user may click in the frame that contains the remaining wall and a frame showing the discarded tiles will be shown.

A draft version of the proposed Main Window appears below:

Main Window Image

**Figure 6.3.1 - Draft of proposed Main Window**

## 7. Document/Training Requirements

This section states all the documentation which is required to be included with the MultiMahjong product. There will be no formal training sessions arranged. The documentation included with the MultiMahjongClient and MultiMahjongServer applications will be of sufficient quality for users (see Section 5.1) to learn how to use the programs without any further explanation.

### 7.1 MultiMahjongServer Documentation

*Level 1 Requirements:*

**7.1.1** An administrator's manual, explaining how to install and operate the MultiMahjongServer, will be provided.

**7.1.2** This manual will be provided as an electronic file.

**7.1.3** The manual will contain the following sections:

- **7.1.3.1** How to install the MultiMahjongServer program.
- **7.1.3.2** The background knowledge the administrator must have in order to set-up and operate the MultiMahjongServer program.
- **7.1.3.3** How to configure the TCP/IP server the administrator is using to run the MultiMahjongServer program.

**7.1.3.4** Troubleshooting guide for any problems the administrator may encounter during setup

*Level 2 Requirements:*

**7.1.4** A section added to the manual which details some technical information on the MultiMahjong product, including:

**7.1.4.1** The languages we used to write the MultiMahjong product and a history of how it developed.

**7.1.4.2** A brief description of how the MultiMahjongServer and MultiMahjongClients interact.

*Level 3 Requirements:*

**7.1.5** The administrator's manual will also be viewable from within the MultiMahjongServer program.

**7.1.6** The manual will contain the following sections (see Section 4.1.9):

**7.1.6.1** How to use the GUI to get log info.

**7.1.6.2** How to use the GUI to change game settings.

### 7.2 MultiMahjongClient Documentation

*Level 1 Requirements:*

**7.2.1** A user's manual, explaining how to install and operate the MultiMahjongClient, will be provided.

**7.2.2** This manual will be provided in both a hard copy book and an electronic file.

**7.2.3** The manual will contain the following sections:

**7.2.3.1** How to install the MultiMahjongClient program.

**7.2.3.2** How to begin a single/multi player game or join a multi player game (see Section 4.2)

**7.2.3.3** How to play a game of MultiMahjong (which describes the rules and scoring in a game of Chinese Mahjong).

**7.2.3.4** An explanation of what all parts of the GUI do (see Section 6.3) and how to use them.

**7.2.3.5** Troubleshooting guide

*Level 2 Requirements:*

**7.2.4** The user's manual will also be viewable from within the MultiMahjongClient program.

### 7.3 Coding and Design Documentation

*Level 1 Requirements:*

**7.3.1** All design documentation and coding documentation will be maintained thoroughly during the project and delivered to the client.

**7.3.2** This documentation will be available in electronic files, and, with the exception of source code, in hard copy format.

**7.3.3** The documentation includes:

- **7.2.3.1** SQAP (Software Quality Assurance Plan)
- **7.2.3.2** SRS (Software Requirements Specification)
- **7.2.3.3** SADD (Software Architectural Design Document)
- **7.2.3.4** SDD (Software Design Document)
- **7.2.3.5** SVVP (Software Verification & Validation Plan) and TP (Test Plan)
- **7.2.3.6** Source Code

*Level 2 Requirements:*

**7.2.4** The source code will be provided in hard copy format.

## 8. Acceptance Criteria

The acceptance criteria for the product are defined by all Level 1 requirements. That is, Level 1 contains the minimum requirements that have to be met for the project to be accepted. Levels 2 and 3 represent likely and future inclusions respectively, the feasibility of which will be determined during the project.

Note that Level 1 requirements may not necessarily be implemented if they are replaced by corresponding Level 2 or 3 requirements.

## 9. Examples of Behaviour

This section describes a possible scenario that incorporates some of the requirements mentioned in this document. As some of the requirements used are Level 2 and Level 3 requirements, the actual implementation of the product may not reflect this scenario exactly.

### 9.1 Beginning the Game

The administrator (5.1.2) runs the MultiMahjongServer program on a server (4.1.1). This program begins running and waits for MultiMahjongClients to connect (4.1.2).

A user (5.1.1), Bob, runs the MultiMahjongClient program on another computer. A dialogue box (6.3.6) appears on the Bob's computer displaying the options to begin the game (4.2.1, 4.2.2). Bob enters his name (4.2.2.1), chooses an icon (4.2.2.2), sets the score limit (4.2.2.4) and decides to select one human opponent and two CO opponents (4.2.2.3). Bob then waits for a human opponent to join.

Another user, Sally, also runs the MultiMahjongClient program, enters her details and selects to join an existing game (4.2.7) from the dialogue box. The MultiMahjongClient retrieves a list of the available games (including Bob's) from the MultiMahjongServer (4.2.6). The MultiMahjongClient displays this list on the screen (6.3.7). Sally selects Bob's game and joins the game.

Bob sees that Sally has joined and selects to begin the game (4.2.2.5). The MultiMahjongServer receives the begin game command from Bob's MultiMahjongClient and it initialises all information necessary to begin the game (4.1.4) and sends it to both Sally's and Bob's MultiMahjongClients (4.1.3). The processing for one CO player will be done by the MultiMahjongClient running on Bob's machine, the other by the MultiMahjongClient running on Sally's machine (4.2.4).

### 9.2 Playing the Game

A main window containing all the game information (6.3.4) is displayed on both Sally's and Bob's screens.

Bob, Sally and the CO's take turns in picking up (4.3.5) and discarding (4.3.6) tiles . After Sally discards a certain tile, Bob can Chow (4.3.4), and a button named 'Chow' in his main window becomes active (6.3.4.6). Bob chooses to Chow by clicking on this button. Bob's MultiMahjongClient sends this move to the MultiMahjongServer (4.3.4), which subsequently sends this move to the other MultiMahjongClient (4.1.3). The tiles in both Bob's and Sally's main window are updated to show that Bob has Chow'ed (4.3.3.2).

The hand continues in this fashion, with all players making moves within the Chinese rules of Mahjong (4.3.1). After a few moves, the combination of tiles in Sally's hand allows her to Mahjong and a button named 'Mahjong' becomes active in her main window (6.3.4.6). Sally clicks this button and a new hand begins. When all hands have been played, the Wind of the Round indicator is updated (4.3.8) and a new round begins. The game continues in this fashion

### 9.3 Ending the Game

At the end of the 4th round, Bob goes Mahjong. Bob's MultiMahjongClient sends this information to the MultiMahjongServer (4.3.8) which sends it to the other MultiMahjongClient (4.3.1). The main window on both MultiMahjongClients is closed and the original dialogue box (6.3.6) is displayed (4.4.3).

Both Bob and Sally choose to quit the program and the MultiMahjongClients on both machines exit (4.4.1).

## 10. Glossary

| | |
|---|---|
| **Chow** | A set of 3 sequentially numbered tiles from a given suit. |
| **Concealed Set** | A Kong, Pung or Chow that has been formed entirely by picking up from The Wall. |
| **CO** | Computer Opponent. The entity that calculates and executes the computer's moves. |
| **Dead Tiles** | All the Tiles which have already been discarded. |
| **Dead Wall** | The Tiles at the end of The Wall which are used to replenish the extra tile used in getting a Kong. |
| **Dialogue Box** | A small window that expects input data from the user. Usually contains text and buttons. |
| **Discard** | The tile a player chooses to discard at the end of their turn. |
| **Dragon Tiles** | Dragon Tiles include Red, Green and White Dragon Tiles. There are four of each of these kinds of Dragon Tiles. |
| **Draw** | A draw occurs when 3 players go Mahjong at the same time or all the tiles up to the dead wall have been drawn without any player going Mahjong. |
| **Exposed Set** | When a discard is picked up in order to form a Kong, Pung or Chow. This set of tiles must be shown to all players for the rest of the hand. |
| **Fishing** | A player is "fishing for Mahjong" when they need only one tile to go Mahjong. All other players in the game must be alerted that the player is fishing. |
| **Game** | A game ends when 4 rounds have been played. |
| **Hand** | The tiles each player has. A winning hand is when a player goes Mahjong by forming an ordinary or special hand. If a winning hand is achieved then all winds rotate anti-clockwise. Once the player who was the Wind of the Round at the start of the round is again that wind then the round is completed. A hand has to be replayed if there is a draw. |
| **Honour Tiles** | These are special tiles in Mahjong. They can be Dragon Tiles or Wind Tiles. |
| **HTML** | HyperText Mark-up Language. The language used to compose and format most of the content found on the World Wide Web. |
| **Kong** | A set of 4 identical tiles. |
| **Limit** | The highest amount that can be scored for a winning hand. |
| **Mahjong** | A 4 player game which is thought to have originated in China around 500 BC. Also the term given to a winning hand. |
| **Ordinary Hand** | A hand containing any four sets of Kongs, Pungs and Chows, and a pair. |
| **OS** | Operating System, the software that controls the execution of all applications and system software programs. |
| **Pair** | 2 of the same tile. |
| **Pung** | A set of 3 identical tiles. |
| **Revealed Kong** | When a player reveals a concealed Kong. They have to do this otherwise they would be a tile short of being able to go Mahjong. |
| **Round** | When the player who was the Wind of the Round originally is that wind again then a round is over and the Wind of the Round changes. |

| | |
|---|---|
| **SDD** | Software Design Description. A design document to be completed at a later stage. |
| **Server** | A central computer in a LAN/WAN that allows connections. |
| **Special Hand** | When a player forms an exact arrangement of designated tiles. To see what constitutes special hands are possible refer to a Mahjong rule book. |
| **Suit Tiles** | Three kinds of suits exist in Mahjong. These suits are Bamboos, Characters and Circles. Each suit has tiles consecutively numbered from 1 to 9. There are 4 of each of these numbered tiles in a suit. |
| **TCP/IP** | Transmission Control Protocol/Internet Protocol. A set of communications protocols developed by the Department of Defence to link dissimilar computers across many kinds of networks. |
| **The Wall** | A 4 sided arrangement of tiles, 17 tiles long and 2 tiles high in the Chinese game of Mahjong, from which each hand is played. |
| **Tiles** | What is used to play a game of Mahjong, just like cards in Gin. Tiles are either Suit Tiles or Honour Tiles. |
| **Wind** | The position which each player is sitting in. Winds include: East, South, West and North. |
| **Wind of the Round** | The wind whose round it currently is. This is important for scoring and determines when a game finishes. The first round is East Wind's round, the second round is South Wind's round, the third round is West Wind's round and the last round is North Wind's round. |
| **Wind Tiles** | Wind Tiles include East, South, West and North Wind Tiles. There are four of each of these kinds of Wind Tiles. |

## 11. Customer Sign-Off

I (the client), hereby agree that the requirements specified within this document agree with my own requirements and expectations of the product, and abide by the Acceptance Criteria (Section 8).

Any alterations to any of the Acceptance Criteria specified above, are to be negotiated between myself and the team.

______________________________
Steve Goschnick (the client)

Team Members:

______________________________
Joanna Araminta

______________________________
Victor Leung

______________________________
Joel Brakey

______________________________

Michael Hart

___________________________
Dean Cortinovis

___________________________
Long Tang

Date: ___/___/___

---

**Other Documentation**

---