國 立 交 通 大 學

電控工程研究所

碩士論文

實體化遊戲互動介面之協同設計

Towards the Design of Coordinated Tangible Devices

研 究 生：吳思穎

Student: Shih-Ying Wu

指導教授：黃育綸　博士

Advisor: Dr. Yu-Lun Huang

中華民國九十九年七月

July, 2010

實體化遊戲互動介面之協同設計

Towards the Design of Coordinated Tangible Devices

研 究 生：吳思穎　　Student: Shih-Ying Wu

指導教授：黃育綸　博士　　Advisor: Dr. Yu-Lun Huang

國 立 交 通 大 學

電控工程研究所

碩士論文

A Thesis

Submitted to Institute of Electrical Control Engineering

College of Electrical Engineering

National Chiao Tung University

in partial Fulfill of the Requirements

for the Degree of

Master

in

Institute of Electrical Control Engineering

July, 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年七月

實體化遊戲互動介面之協同設計

學生：吳思穎　　　　　　　　　　　　　　　　　指導教授：黃育綸　博士

國 立 交 通 大 學電控工程研究所碩士班

摘　　要

圖形化使用者介面（Graphical User Interface, GUI)讓人們可以透過滑鼠來與數位資訊互動，而不需要記憶特殊的控制指令。然而，這樣的操作方式與一般人們的生活方式並不相同。因此使用者介面（Tangible User Interface, TUI）的概念因應而生。透過緊密結合的實體物體與數位資訊，提供一個更為友善的互動方式：使用者能以更簡單及更直覺的方式來完成他們想做的事情。

本論文延伸實體使用者介面的概念，為多實體裝置的協同動作建立一個系統架構。我們將這套架構應用在一般的網路麻將上，開發出一套新玩法的網路麻將。透過系統的協同機制，使用者所有的遊戲操作都與一般現實生活中一樣，無需再學習新的操作方法，也能透過系統與遠端的其他玩家一同進行遊戲。

Towards the Design of Coordinated Tangible Devices

Student: Shih-Ying Wu Advisor: Dr. Yu-Lun Huang

Institute of Electrical Control Engineering

National Chiao Tung University

Abstract

Computer-Human interaction studies the interaction between a user and a computer. The traditional computer-human interaction solutions, including the command line interface (CLI) and graphical user interface (GUI), provide user interaction with a computer through a keyboard or a mouse. However, such interfaces are non-intuitive since the manipulation may be inconsistent with human daily life. In recent years, a new user interaction using tangible user interface (TUI) was proposed to solve the above inconsistence. The goal of TUI is to augment the real physical world by seamlessly coupling digital information to physical objects and environments, and provide a better user-friendly interface. In this paper, we adopt the concept of TUI and design a model to cooperate multiple tangible devices for a networked game. In our implementation, multiple tangible devices are installed at the client sides, which is then divided into two parts: a worker and a messenger. The worker thread deals with the rule of game, while the messenger helps to map the relationships between a virtual identity and a physical tangible devices. We ultimately implement the proposed model with a networked mahjong game (QKMJ). With our implementation, players can play the game easily and intuitively.

誌謝

本論文能順利完成，首先要感謝我的指導老師，黃育綸博士，在這兩年中認真和耐心地教導我，培養我正確的研究方法，尤其是在任何事情的完整性、邏輯性等，讓我受益匪淺。此外，老師更在硬體設備方面鼎力襄助，採購許多實作上所必要的硬體，讓我能安心地完成這篇論文及實作。

再來要感謝蔡欣宜學姊，在碩二與學姊同寢期間，不論是生活、論文研究及寫作各方面都受到學姊的照顧和幫忙。還要感謝陳柏廷學長的提點，讓我能以更流暢的方式完成我的論文。還有宛真、雅萱、虹君、廷芳、佳旻，從大學時期就讓我擁有許多美好回憶。當然還有RTES的大家，創造出輕鬆的研究環境，讓我在作實作與寫論文之外，亦添增了許多歡樂。此外要特別感謝晉澤，協助我一一修正口試報告的不足之處，讓我能順利完成口試，在我面臨低潮時給予我支持及鼓勵，陪我走過人生的重要關卡。最後，感謝家人及朋友們的支持，讓我的求學生涯一路順遂，並期許自己在將來能更加精進。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the design of computer-human interfaces for networked games. We also discuss the existing games using tangible devices that offer a more friendly interface to users interacting with a computer.

1.1 Computer-Human Interfaces

The Computer-Human interface (CHI) is the system that provides users with interaction between humans and computers. CHI is composed of several input devices supplying the users to operate the computer and output peripherals supplying the computer to show the results. The interaction model of CHI is composed of a series of request-response transactions. These transactions can be realized by different types of requests and responses, which contributes different types of CHI.

The following are the most common of several types of public, including Command-Line User Interface (CLI), Graphical User Interface(GUI). Then, we take a brief description of Tangible User Interface(TUI).

1.1.1 CLI

The Command-Line Interface (CLI) is a mechanism supporting interaction between a user and a computer using text-only commands. Starting from 1969, the Command-Line Interfaces

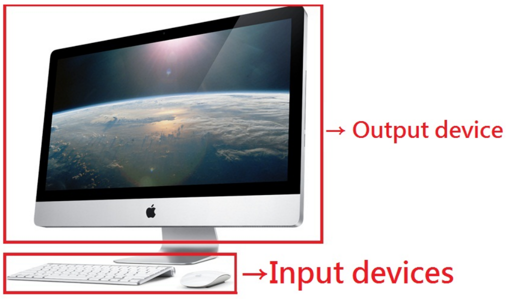


Figure 1.1: traditional peripheral

(CLIs) have been widely deployed to a variety of applications. With such an interface, textual commands with special vocabularies typed from a keyboard are sent to the computing unit. These textual commands are treated as requests from a user. Similarly, textual responses are printed on the screen to tell the users the computational results, in response to the requests.

1.1.2 GUI

By leveraging the advantages of graphic technologies, the Graphical User Interface (GUI) is used to provide a more user-friendly interface when interacting with a computer [1, 2]. The first generation of GUI was set by Xerox Star workstation in 1981 [3]. The Xerox Start workstation introduced the first GUI operating system. Windows, Icons, Menus and Pointing device (also known as WIMP interfaces [4]) are used to support commands to open and delete a file. People interacts with the computers through GUI near 30 years. The expression of a GUI request is more friendly than that of CLI. In addition to a keyboard, a mouse is adopted as one of the input devices when adopting GUI as the computer-human interface. With the comprehensive input devices, users can directly operate their computer without special commands. When applying GUI, the response of an application can be either graphical or textual.

1.1.3 TUI

Figure 1.1 shows the traditional peripherals we used to interact with a computer. The traditional input devices include a keyboard and a mouse while a monitor is used as an output device. Such a design has relatively remained unchanged for the past 40 years. However, such a design is non-intuitive since a user has to learn another input method, inconsistent with human daily life, to interact with his computer.

In 1997, Hiroshi Ishii and Brygg Ullmer proposed an interactive conception named Tangible User Interfaces (TUIs) as a new type of CHI [5]. The goal of TUI is to augment the real physical world by seamlessly coupling digital information to everyday physical objects and environments to narrow the gap between virtual world and reality . That is to say the input devices are no longer limited to keyboard or mouse, and the output device is not just monitor. With such an interface, users can easily and intuitively manipulate what they want to do.

1.2 Existing Tangible Games

In recent years, there have been numerous papers to discuss tangible user interfaces from a variety of perspectives after tangible user interface published in 1997 [5, 6]. In 2006, Nintendo Wii [7] has published with its action-detecting controller, Wii Remote. This is an innovative device to replace traditional input peripherals, but keeping a monitor as its output peripheral. For intuitive operation, in 2007, Chia Han Tsai proposed an Emotion Teaser [8] composed of a LED panel as a output device and a mercury switch as a input device. However, thid device is just designed for a special game. In 2009, David Merrill proposed Siftable[9] with firmware and API for general cases.

1.3 Motivation

In this paper, we propose a model cooperating multiple active tangible devices for a networked game. In our design, multiple tangible devices are installed at the client sides to support tangible user interfaces. Then, we revise the client sides to deal with the changes of extrinsic multiple tangible devices. We ultimately realize the proposed model with the implementation of a network mahjong game, QKMJ.

The remainder of this paper is organized as follows. Section 2 introduces the existing researches supporting tangible user interface. Section 3 describes the background of our implementation. Section 4 presents the proposed model and Section 5 discusses the our implementation in detail. Finally, Section 6 concludes the work.

Chapter 2

Releted Work

There have been numerous papers to discuss tangible user interfaces from a variety of perspective [10, 11, 12] after tangible user interface published in 1997 [5, 6]. Among numerous tangible user interface applications, we introduce some of them in the following: Wii, Emotion Teaser, and Siftables.

2.1 Wii

In 2006, Nintendo [13] introducted a new game concept which involved focusing on a new form of user interaction to community. That is the well-known home video game console, wii [7].

The particular feature of wii is the operational method of its wireless controller, wii remote. This is an innovative device to replace traditional input peripherals [14]. The user can use wii remote as a handheld pointing device and the movement of this user on wii remote can be detected in three dimensions.

The principle of movement-detecting work is a combination of built-in accelerometers and infrared detection. This infrastructure can used to detect the action and sense position of wii remote in 3D space. With this infrastructure, a user can control the game instinctively.

The data transmission between wii and wii remote applies Bluetooth wireless protocol [15], widely deployed to a variety of applications. With Bluetooth, a device can communicate with seven decives, and now, wii can support at most four wii remote at the same time .

2.2 Emotion Teaser

Emotion teaser is proposed by Chia Han Tsai in 2007 [8]. This system adopts the concept of tangible user interface, and it is a wireless control system. The purpose of Emotion Teaser is to transform human emotions into multimedia animation of digital contents. The user expresses he/she emotions through shaking an emotion teaser device intuitionally, and shares with each other's emotions through multimedia animation of digital contents.

The Emotion Teaser device is the characteristic of this system. In addition to a input peripheral with a mercury switch, the Emotion Teaser device is a kind of output peripheral through subjoining a LED panel to it. Therefore, the interaction between the user and digit content is more direct than before.

Unlike wii, Emotion Teaser adopts another wireless protocol, ZigBee[16], for data transmission. The usage of ZigBee enables the system to communicate with more devices than one of Bluetooth, which is bounded with seven devices.

2.3 Siftable

In Feb of 2007, David Merrill, Jeevan Kalanithi, Pattie Maes proposed Siftable in an early stage [17]. Siftables are the sets of the cookie-sized computers with motion and neighber sensing, graphical display and wireless communication. Accelerometers is applied to detect the motion while the infrared detection in all four directions is applied to sense neighber.

Developing several years till 2009, Siftables are not only the tangible devices, but a mature event-driven operating system [18]. David Merrill developed the firmware running on the computering unit on the Siftable, the ASCII language specification for basic remote control, and the Python API for developing application to interact with Siftables.

Several Siftables applications are published in these years: Telestory, Color Pouring, Scraboggle, ...etc. Seth Hunter and Katya Popova created Telestory[19], a language-learing application designed to teach vocabulary to children. The child places the Siftable showing the dog next to the Siftable showing the cat ,then dog and cat display adjacent in the large screen.

Jeavan Kalanithi created Color Pouring that allows pouring a color from one Siftable to another adjacent one by tilting the Siftable to the side of adjacent one. Scaboggle is a word-finding game. Each Siftable in this game shows an alphabetic character, and user spell words by placing Siftables into contiguous rows or columns. Motion detecting is characteristic of Color Pouring while cooperation of multiple devices is characteristic of Scaboggle.

Chapter 3

Coordinating Model

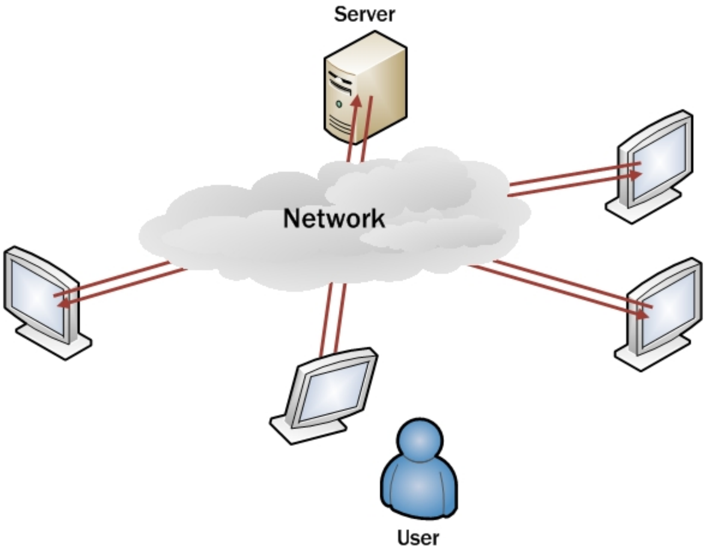
3.1 System Architecture

This section discusses the model we proposed to manipulate multiple active tangible devices for a networked program.

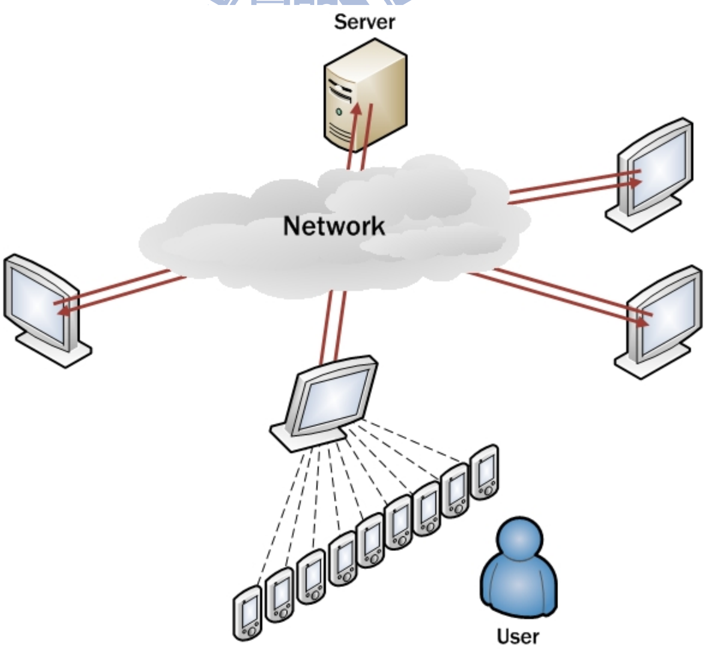
Figure 3.1a shows the traditional model of a networked program. Such a model consists of one connection, one server and several clients. Each computing unit (either a server or a client) communicates with each other through the connection. Figure 3.1b shows the proposed model that can be used to coordinate multiple tangible devices. In such a design, multiple tangible devices are installed at the client sides to support tangible user interfaces.

Comparing the two above models, in addition to an external connection between the game client and the server, multiple network connections are required among the game client and tangible devices. To distinguish the two types of connections, the former connection is named as an external connection while the latter ones are named as internal connections, which help forward messages between the game client and the tangible devices.

In our design, we divide the client into two components: a worker thread and a messenger. The worker thread deals with the rule of program, and the messenger copes with the translation between the worker thread and the tangible devices, as illustrated in Figure 3.2. Such a design maintains the communication mode between the server and the client, then the user of the



(a) Traditional Model



(b) Proposed Model

Figure 3.1: Traditional Model v.s Proposed Model

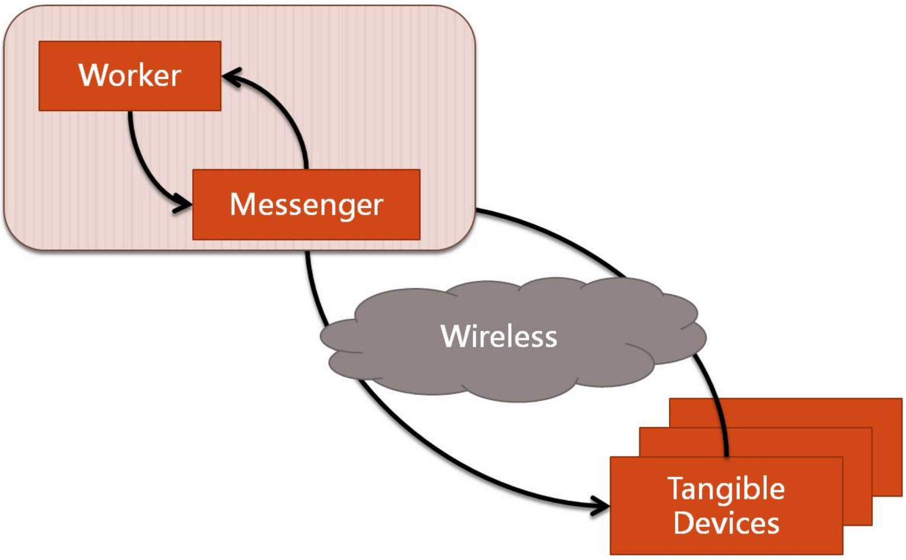


Figure 3.2: The Detail Schematic Diagram of a Client

proposed model can communicate with others using either the traditional model or the proposed model.

3.2 System Components

Consequently, the proposed model consists of the following components: an external connection, a server, a client consisting of multiple tangible devices, a worker thread, a messenger, and multiple internal connections.

- External Connection

  As mentioned above, the external connection between the server and the client helps forward messages between the two parties. Similar to other networked programs, the external connection can be realized by a heterogeneous network, either wired or wireless.

- Server

  The server of a distributed program is able to communicate with multiple clients through one or more external connections. Since no change is required between the endpoints of the external connection, the server of a distributed program may remain unchanged when supporting tangible interfaces. Only revisions on the game client, consisting multiple tangible devices, a worker thread, a messenger and multiple internal connections, should be made.

- Client

  The client consist of several components, including multiple tangible devices, a worker thread, a messenger, and multiple internal connections:

  – Tangible Devices

    Tangible devices, interacting with users, can be used to replace the input peripherals, such as a keyboard, a mouse, etc. The tangible devices help detect user behaviors and actions, and respond the actions to the worker thread via the internal connections. A tangible device is composed of a sensor and a computing unit. If image display is required, the tangible device may optionally be equipped with a display panel. The sensor detects the motion of the tangible devices while the computing unit copes with several control commands to either control the display panel or respond the user behaviors to the client.

  – Worker Thread

    The worker thread mainly executes the processes running on the client. As mentioned above, to support tangible user interface, the traditional input peripherals of the client are replaced with one or more tangible devices. To communicate with these tangible devices, the client needs to execute a mechanism to coordinate the tangible devices and analyze the messages responded by the tangible devices.

  – Messenger

A component of the client. Literally, the messenger is a forwarder in the proposed model. The messenger is responsible for transmitting commands and receiving responses to and from tangible devices. The messenger also help mapping the relationships between the virtual identities and the physical tangible devices.

- Internal Connection

  Contrast with the external connection, an internal connection exists between a messenger and a tangible device. Unlike the external connection, the internal connections are normally realized by wireless networking technologies, including 802.11, ZigBee, etc.

3.3 System Sequence Flow

Finally, we introduce the basic transmitting flow between worker and tangible devices as shown in Figure 3.3. When the worker receives a message, the demand for user behavior, from server or other clients, it has to send a command to some specific tangible devices to obtain user behavior. However, the command is not sent to some specific tangible devices directly, and it is sent to the messenger first. When the messenger receives the command form worker, it analyzes the command. Then, the messenger obtains the receivers, the actual specific tangible devices, via mapping relationship and delivers the command to the receives.

On the other hand, When the tangible device detects user behaviors and actions, it must to respond the user behaviors to the worker. And similarly, the response is not sent to the worker directly, sent to the messenger. Then, the messenger obtains the corresponding virtual identity, which is well recognized by the worker, through mapping relationship. Finally, the messenger forwards the response to the worker, and the worker analyzes the response(s) to get a conclusion to reply the server or other clients.

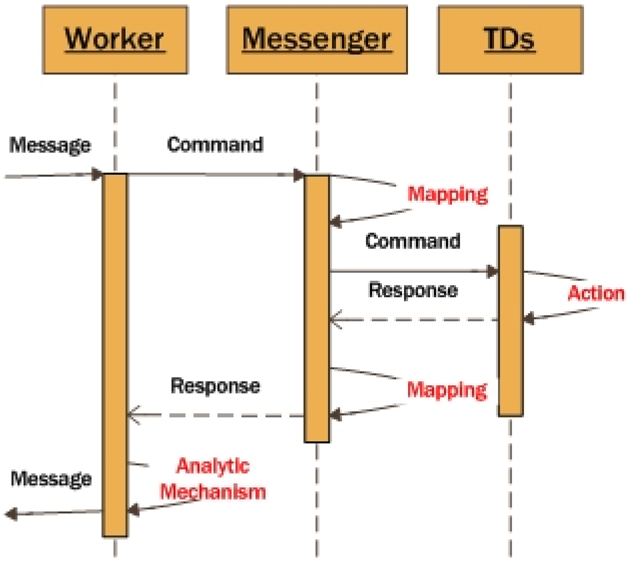


Figure 3.3: The transmitting flow between worker and tangible devices

We summary the system sequence flow in following steps, and step 2, 4, 5, 6 must be customized for different applications.

1. Worker sends the command to messenger.
2. Messenger analyzes the command to obtain the mapped receiver(s).
3. Messenger delivers the command to the mapped receiver(s).
4. Tangible Device detects user behavior and sends the response to messenger.
5. Messenger analyzes the response to obtain the corresponding virtual identity.
6. Worker analyzes the response(s).

Chapter 4

Mahjong Game

4.1 Preliminaries

In this section, we epxplain the main parts of terminology and rule of mahjong game used in our implementation [20, 21].

4.1.1 Game Terminology

First, we introduce the terms used to explain our implementation: realizing a networked mahjong(MJ) to support TUI through our proposed model. It is helpful for basic understanding to begin with a rough classification, so we sort the terms of MJ into 5 categories: process, role, entity, meld, and operation.

MJ Process

MJ is a kind of draw-and-discard game. The basic action, named turn, is to draw a tile from draw pile and discard one to discard pile. A player in turn is called dealer.

MJ Role

Four players form a MJ game board, sitting around a square table (as shown in Figure 4.1). We refer the player itself as Local Player (P_L). For each P_L, the player whose seat is left/

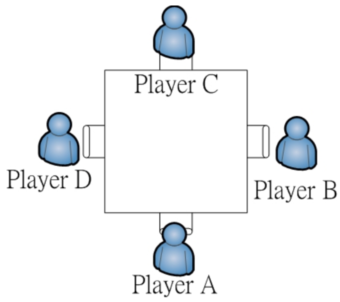


Figure 4.1: MJ game board

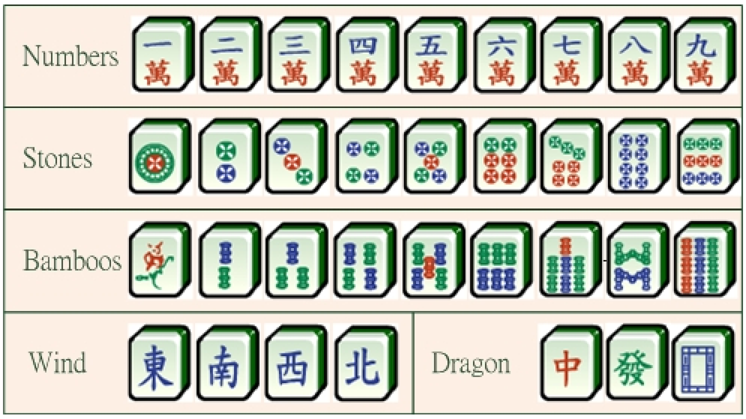
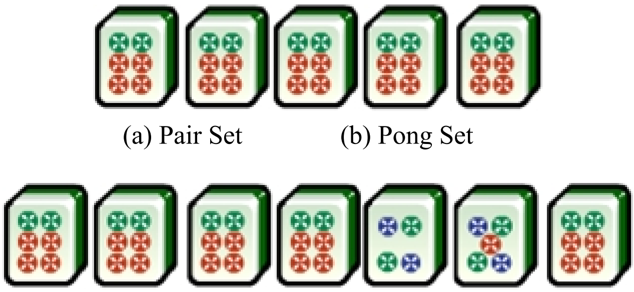


Figure 4.2: MJ tiles

right side is Previous Player (P_P)/Next Player (P_N), and the player is Opposite Player(P_O) when sitting opposite. Eg. in Figure 4.1, player D is P_P of player A, player A is P_N of player D, and player B is P_N of player A.

MJ Entitiy

We bring to the notice of terms about MJ entity, called tile in general. The content of a tile is rather like a standard deck of card, and it is composed of different suit and value. There are five suits in MJ game: Numbers, Stones, Bamboos, Wind, Dragon, and just the first three suits have value, running from 1 to 9. Figure4.2 shows the tiles mentioned above, and the quantity of each kind of tiles is four.



(a) Pair Set (b) Pong Set

(c) Kong Set (d) Sheung Set

Figure 4.3: MJ Melds

MJ Meld

Term meld is used to represent the combination of multiple tiles, and there are four melds used in MJ game: pair set, pong set, kong set and sheung set. Actually, pair set, pong set and kong set are similar melds with the same kind of tiles, but the quantity of these sets is different: two tiles for pair set , three tiles for pong set, and four tiles for kong set. Sheung set is only with the same suit, but value in a row. Example for four melds is revealed in Figure 4.3.

MJ Operation

Operation, the remaining category is discussion of some operations differing from the basic operation, drawing and discarding. There are four operations used in MJ: Pong, Kong, Sheung and Concealed-Kong. The first three is the operation of how to get a tile discarded by other players, while the last one is the self-completed operation. We explain these operations in two aspects: when a player can do and how a player do the operation.

First, a player can do Pong/Kong when other player discards a tile to which he/she has two/three ones identical, i.e these three/four tiles make Pong/Kong Set. Sheung only be declared off the discard of previous player, and he/she must have two tiles to make it a Sheung Set. However, only dealer can declare Concealed-Kong when having four identical tiles without being discarded or exposed.

Table 4.1: MJ terms

category	term	explanation
Process	turn	to draw a tile from wall and discard one
	dealer	the person in turn
Role	Local Player(P_L)	the player itself.
	Previous Player(P_P)	the person sitting left side of the player
	Next Player(P_N)	the person sitting right side of the player
	Opposite Player(P_O)	the person sitting opposite of the player
Entity	tile	mahjong set, composed of suit and value
	suit	including Numbers, Stones, Bamboos, Wind, Dragon
	value	range from 1 to 9 for Numbers, Stones, Bamboos
Meld	Eye set(M_E)	a set of 2 identical tiles
	Pong set(M_P)	a set of 3 identical tiles
	Kong set(M_K)	a set of 4 identical tiles
	Sheung set(M_S)	a set of 3 suited tiles in a row.
Operation	Pong(O_P)	When : A player has 2 tiles identical with the discarded one How : expose the two tiles in hand
	KongO_K	When : A player has 3 tiles identical with the discarded one How : expose the three tiles in hand
	SheungO_S	When : only be declared off the discard of P_P How : expose the two tiles in hand
	Concealed-KongO_CK	When : dealer has 4 identical tiles in hand How : conceal four tiles in hand

Second, if a player want to do Pong, Kang or Sheung, he/she has to expose two tiles for Pong or Sheung and three tiles for Kong, and these tiles cannot be used anymore in this game. A player conceals four tiles to indicate these tiles is non-use ones for Concealed-Kong.

MJ terms mentioned above are summarized in Table 4.1.

4.1.2 Game Rules

Each player has sixteen tiles in hand at beginning. A player on turn draws a tile and then discards one in hand. The Goal of MJ game is to make six melds within one M_E.

Manipulating mode

There are two modes: normal mode & bid mode for each player in a running game.

Normal mode When it turns to the local player, the manipulating mode is automatic translated to normal mode. There are two operational steps in normal mode. The first step in normal mode is that the local player must draw a tile from wall. However, there are at least one option and at most four options the local player can select to execute in second step.

On the whole, *discarding a tile in hand* is the option that always can be selected without any constraints and it be known as the basic operation in MJ game. The remaining three options can be selected by the local player under some constraints. *Exposing a tile in hand as Kong* also named *Plus-Kong*(O_{PK}) is the option that if the local player has done O_P in this game with a kind of tile and he/she has the fourth of that particular tile in hand. *Concealed-Kong* mentioned in 4.1.1 is the third option. Then, if the local player selected Plus-kong or concealed-kong, he/she has to draw another tile as the first step of new cycle in normal mode. Finally, *winning this game*, O_W the last option can be selected when the local player has six melds within one M_E regardless of exposed or in-hand.

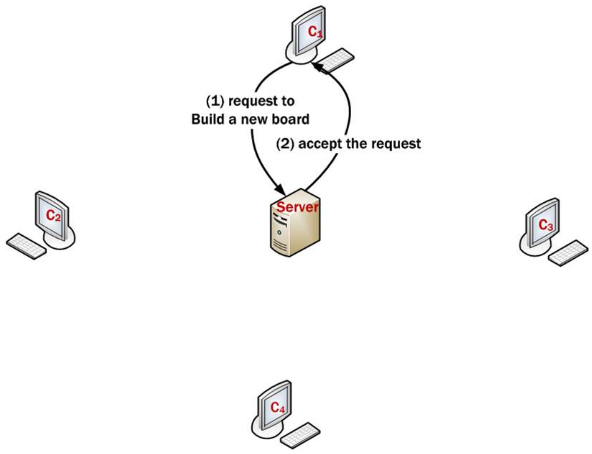
Bid mode If other player(except the local player) discarded a tile which the local player could use to complete a meld and want to do it, then the bid mode is triggered. There are at least one up to four options the local player can select : *Sheung, Pong, Kong, winning this game*, all of these options also called bid operation. If the local player selects Sheung / Pong / Kong, it means the player should do O_S / O_P / O_K , then he/she should discard another tile in hand to accomplish the steps of bid mode. Therefore, the next default dealer is P_N of the local player. On the other hand, if the local player selects winning this game, this game is over. However, if a discarded tile was bid by more than one player, the priority ordering from high to low is winning this game, Kong, Pong and Sheung.

4.2 QKMJ

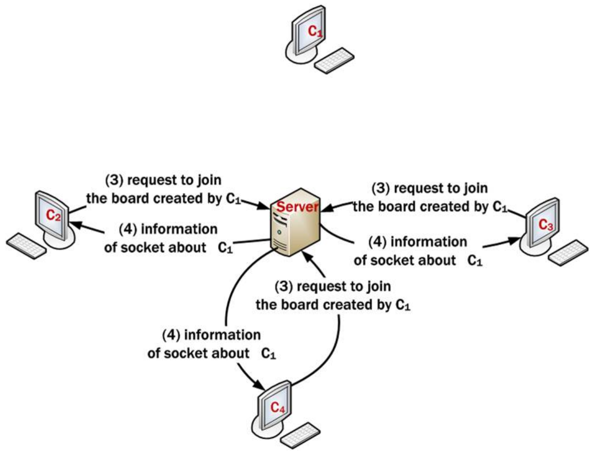
In this section, we introduce the mahjong game on which our implementation is based. QKMJ, a kind of Taiwan Mahjong game, is proposed by Shian Yow Wu [22] in 1994. It is a networked game based on client-server model running on unix platform. The user interface of QKMJ is command-line interface.

In our implementation, we focus on how to support multiple tangible devices on game procedure, so we describe previously how to start a MJ game board in the following. There are three phases before user starts a MJ game: Initial phase, Join phase, Game phase, as shown in Figure 4.4.

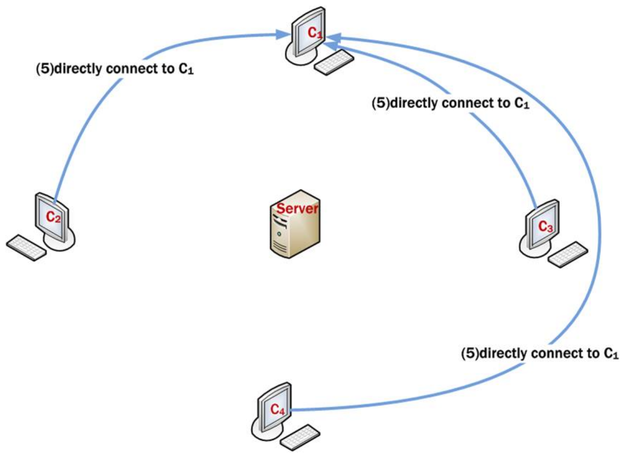
In initial phase, A player (C_1) sends a request to game server in order to build a new game board. The game server accepts the request and records the socket information of C_1 as illustrated in Figure 4.4a. Then, other players can get message about the above game board from game server, and they can send request to server to join the game board built by C_1. When the server receives the joining request by other player, it sends the socket information of C_1 to the demander as shown in Figure 4.4b. Therefore, the other player can directly communicate with C_1. After gathering four players including C_1 shown in Figure 4.4c, the MJ game starts , and our implementation begins from here.



(a) Initial Phase



(b) Join Phase



(c) Game Phase

Figure 4.4: Preliminary of QKMJ

Chapter 5

Prototype

In this chapter, we use a networked mahjong game, QKMJ, to implement our model, and we name it with Tangible Mahjong (TMJ). The following discusses TMJ with two aspects: hardware and software.

5.1 Hardware

We have mentioned in section 3.1 what kind of hardware requirement we need. Then, we introduce our development platform in this section.

Game Server

The architecture of game server remain unchanged as described in section 3.1. The operating environment of our TMJ is like one of QKMJ, running on unix platform. Hence, the game server of TMJ applies a person computer, whose operating system is Ubuntu9.10.

Game Client

Computer Unit

A person computer is applied to process the worker thread in our implementation and a wireless aggregator is attached it as the messenger. Such an implementation, the Universal Serial Bus (USB) is the communicating interface between the worker and the messenger.

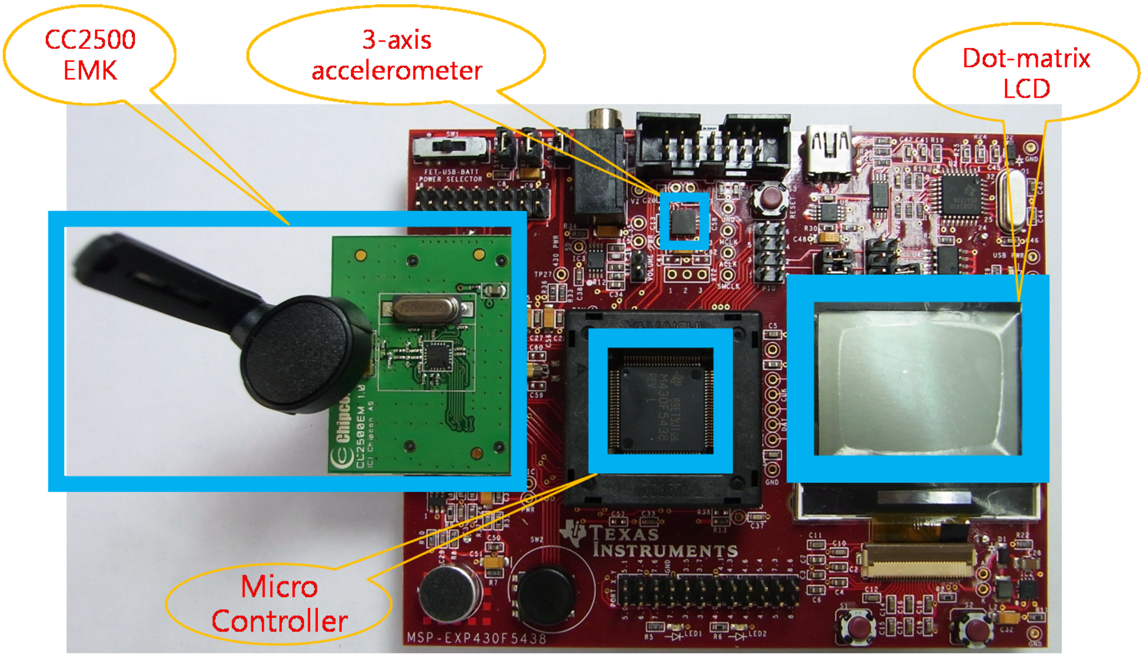


Figure 5.1: Tangible Device (Mahjong Tile) - Combination of MSP-EXP430F5438 and CC2500.

Tangible Devices

According to the requirement of tangible device mentioned in section 3.1:sensor, panel and wireless transmission, an experiment board, MSP-EXP430F5438 [23] is applied to address these requirement. Although this experiment board itself is no wireless transmitting capability, it has a wireless solution : its wireless kit - CC2500 [24]. Finally, a MJ tile is the combination of MSP-EXP430F5438 and CC2500 (as shown in Figure 5.1).

The develop platform for MSP-EXP430F5438 is Code Composer Studio(CCS) v4, an Integrated Development Environment(IDE) running on Microsoft Windows 7/Vista/XP. We use the free version of CCS v4, has 16kB code size limited for MSP-EXP430F5438.

- The sensor we used within MSP-EXP430F5438 is 3-Axis Accelerometer, and its model is ADXL335. The Data Sheet of ADXL335[25] provides some information and schematic diagram: it uses a single structure for sensing the X-axis, Y-axis and Z-axis [1]

[1] The three axes' sense directions are highly orthogonal and have little cross-axis sensitivity.

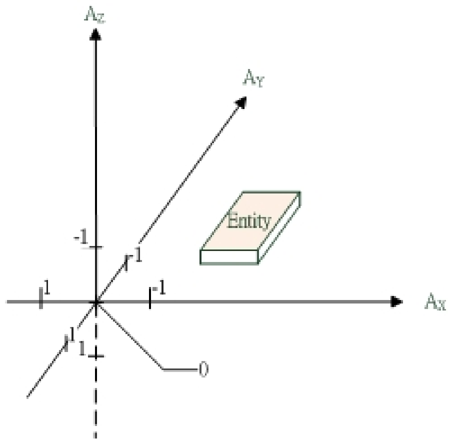


Figure 5.2: Axes of Acceleration Sensitivity

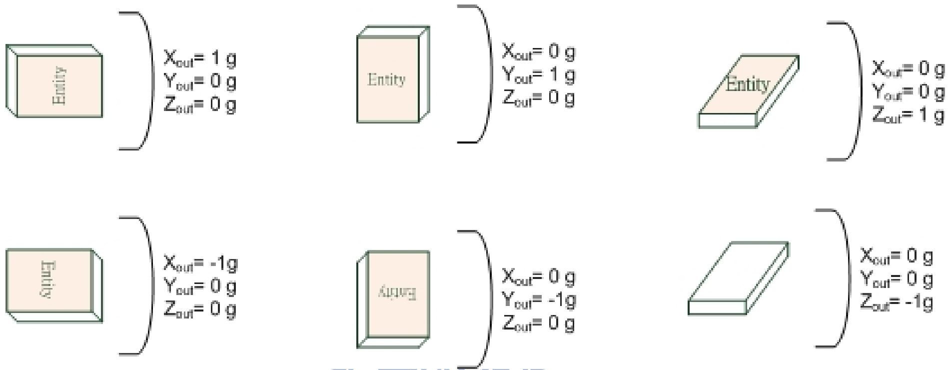


Figure 5.3: Output Response vs. Orientation to Gravity

as shown in Figure 5.2.

The values of axis scale are meaningless but the relative of these values is meaning. The output response is proportional to acceleration along the sensitivity axis, even if the tangible device is static, the gravity still makes no-zero output response (as shown in Figure 5.3). For convenience, we define the output response of these three axes as a three-dimension(3D) vector(X, Y, Z).

- The model of built-in panel is HD66753. It is a Hitachi dot-matrix LCD with a resolution of 138*110, 4-level grayscale pixels.

- The wireless model CC2500 [24] is low-cost, low-power 2.4GHz RF transceiver.

5.2 Sofeware

We introduce our sofeware prototype for TMJ in this section.

Game Server

The game server of our TMJ is the same with that of QKMJ.

Worker of Game Client

The worker in our TMJ is based on the game clinet in QKMJ. However, we modify it, so it can communicate with tiles through messenger and has ability to analyze messages from tiles. Due to two manipulating mode in mahjong game, we divide the analysis into two portions : Judge Mechanism in normal mode and in bid mode.

Judge Mechanism - Normal Mode

As mentioned in 4.1.2, there are 2 operational steps in normal mode. The first step is fixed operation, drawing a tile from wall, so if the local player do action draw (defined in 5.2) on a tile, the worker can directly determine correctly the operation the local player do. However, the player has at most 4 options in second step, so we must to design a operation-judging mechanism for the worker (Figure 5.4) to assist us to determine the operation correctly. Notice: the assumption of Figure 5.4 is that the local player had completed the step 1 of normal mode, ie. drawing a tile from wall.

Before the local player acts on tile(s), to reset the counters is necessary and to check[2] whether the local player can do O_W or O_{CK} to set the corresponding flag(s)[3]. These flags will

[2] The corresponding constraints are described in 4.1.2.

[3] It's meaningless to check the flag of O_{PK} in advance, because this operation is decided based on the content of the active tile.

assist us in operation-judging period. On the other hand, by observing actions of each operation, we discover the action of O_{CK} is conceal (defined in 5.2) and all the one(s) of O_W, O_{PK} and discard are filp (defined in 5.2). As a result, the worker can preliminary distinguish whether the local player want to do O_{CK}.

When the local player do action conceal on a tile with state in-hand (defined in 5.2), the worker will receive a corresponding response message form the tile. Then, it increases counter if the flag of O_{CK} is set and the content of the concealed tile is valid for O_{CK} (shown in Figure 5.4 as the conditional expression : " Correspond with O_{CK}? "). When the number of counter equals 4, O_{CK} is the operation the local player do, and the operation-judging work finish.

However, when the local player do action flip on a tile with state in-hand, the worker cannot distinguish what the local player want to do only by a flip-reporting message. Therefore, some conditionals are applied to assist us to determine which operation is correct. We discover the number of manipulated tiles of the possible operation O_W is different than O_{PK} or Discard. The number of manipulated tiles of operation O_W must larger than 2, but the one of the last two is 1. It means that the worker not only receives the flip-reporting message one time if the local player want to do O_W. As a result, the worker prior considers whether the flag of O_W is set (shown in Figure 5.4 as the conditional expression :" Correspond with O_W? ").

If the flag of O_W is not set, then the operation the local player want to do only remain 2 possibilities: O_{PK} or Discard. Therefore, the worker considers the content of this flip tile, if the tile meets the constraints of O_{PK}: the content of the tile is the same as the one of the exposed pong meld (shown in Figure 5.4 as the conditional expression : " Correspond with O_{PK}? "), the worker can get a conclusion that the O_{PK} is the operation the local player do, and the operation-judging work finish.

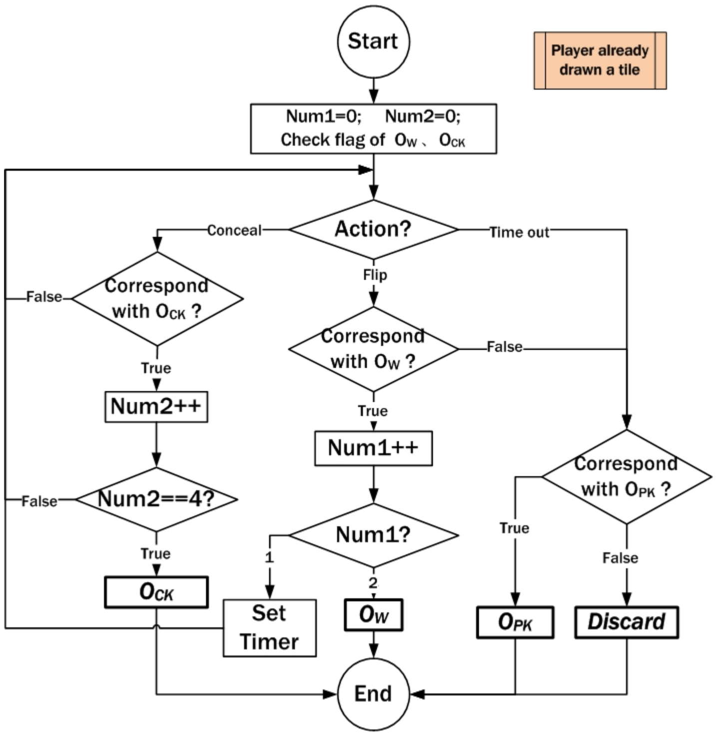


Figure 5.4: Judge Mechanism - Normal Mode

On the other hand, if the flag of O_W is set, there are still 3 possibilities of the operation. Nevertheless, another counter be applied to record the number of flip-reporting message(s), if the worker just receive a message now, timer will be applied to assist judging operation: when the worker receives another flip-reporting message in the limited period, it can determine the O_W is the operation the local player do, and the operation-judging work finish. However, if the worker receive the Timeout-reporting message, it means the O_W is not the one, so the approach for not setting flag of O_W will be applied to cope this situation.

Judge Mechanism - Bid Mode

We analyzed the bid mode in 4.1.2, so we know that the bid mode not always be triggered every time. Therefore, the worker have to judge whether the local player wants to bid the discarded tile, and a timer is applied to assist the worker to understand the intention of the local player. If worker receives any flip-reporting message in the limited period, it will be accounted

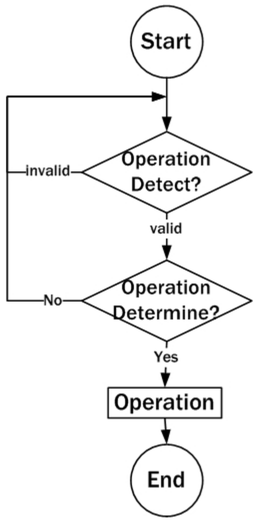


Figure 5.5: Judge Mechanism - Bid Mode : Basic Processing Logic Schema

that the local player intents to bid the discarded tile, and vice versa. Then, after the worker accounts the local player's intention to bid the discarded tile, the operation-judging mechanism in bid mode is triggered.

Figure 5.5 shows the processing logic of the operation-judging mechanism after it receives a flip-reporting message. First, it checkes whether the message is valid for one of the bidding operations the local player allowed to do. (We will discuss what is valid later) The message will be ignored if it's invalid. But, if it's valid, the worker subsequently checkes whether the valid messages collected so far (in this judging-operation period) meet the constraints of a specific bidding operation. If these messages meet the constraints, the worker can get a conclusion that this specific bidding operation is that the local player want to execute, then the operation-judging mechanism terminated. However, if these messages don't meet the constraints, this operation-judging mechanism still works until it receives enough constraint-meeting messages.

Before we discuss which message is valid for operation-judging process, we list TMJ notations used in Table 5.1.

Contents of a active tile, number of active tiles, and timer are the bases applied to judge

Table 5.1: TMJ Notation

Notation	Definition
T_i	Single tile
ST_i	Suit of T_i
VT_i	Value of T_i
Γ_{act}	Set of act tiles in this period
ΔT	A limited time
Δt	The time between detecting active tiles

Table 5.2: Valid Messages for Judge Mechanism in Bid Mode

ID	Schema	Conditional	Handling
$VM1$		F_{O_S} [1] $= 1 \cap\ ST_i = ST_{bid} \cap$ $0 < \lvert VT_i - VT_{bid} \rvert \leq 2$	$F_{O_P} = 0$, $F_{O_K} = 0$, $\Gamma_{act} + +$
$VM2$		$F_{O_S} = 1 \cap\ \Gamma_{act} = 1 \cap$ $(T_i, T_{act}, T_{bid}) \in M_S$	$F_{O_P} = 0$, $F_{O_K} = 0$, $\Gamma_{act} + +$
$VM3$		$F_{O_P} = 1 \cup\ F_{O_K} = 1$	$F_{O_S} = 0\ \Gamma_{act} + +$
$VM4$		$\Delta t \geq \Delta T$	end of judging
$VM5$		$F_{O_W} = 1 \cap$ $\notin (VM1, VM2, VM3, VM4)$	end of judging

[1] $F_{O_j} = 1$ represents that the flag of O_j is set, and vice versa.

whether the message is valid. A total of the valid messages is five, and each of them has corresponding conditional listed in Table 5.2.

Among these valid messages, the judging basis of $VM4$ is timeout, while the ones of others are contents or number of active tiles. There are 2 situations that we apply timer to assist judging operation. First, the flag of O_W is set and the flag of O_P or O_K or O_S is set. Second, Both the flag of O_P and O_K are set.

Like judge mechanism in normal mode, we check whether the local player can do operation, O_S, O_P, O_K and O_W, then setting the corresponding flag(s). We use a circle symbol including some of operations (as illustrated in Figure 5.6) to represent a kind of situation that flags of these operations are set.

Figure 5.6 shows all the situations of judging procedure of the operation-judging mechanism in bid mode. At beginning, there must exist a kind of situations, all possible situations at beginning are listed on the left of Figure 5.6a. If a valid message appears, the worker executes the corresponding handling also listed in Table 5.2. With these handling, the quantity of possible operations decreases gradually. However, even if one possible operation is remained, the operation-judging mechanism keeps working until receiving enough valid messages to meet the corresponding constraints.

The scenario of first valid message adopted by the worker is illustrated in Figure 5.6a. Obviously, for most of situations on the left, the valid message reduces the possible operations. However, the quantity of valid message is insufficient, so operation-judging mechanism continues to work.

Similarly, the scenario of second, third, and forth valid message adopted by the worker are illustrated in Figure 5.6b, Figure 5.6c, and Figure 5.6d. We paint the circle symbol with colorful background to represent the worker ends judging operation, and combine these scenarios into Figure 5.6e.

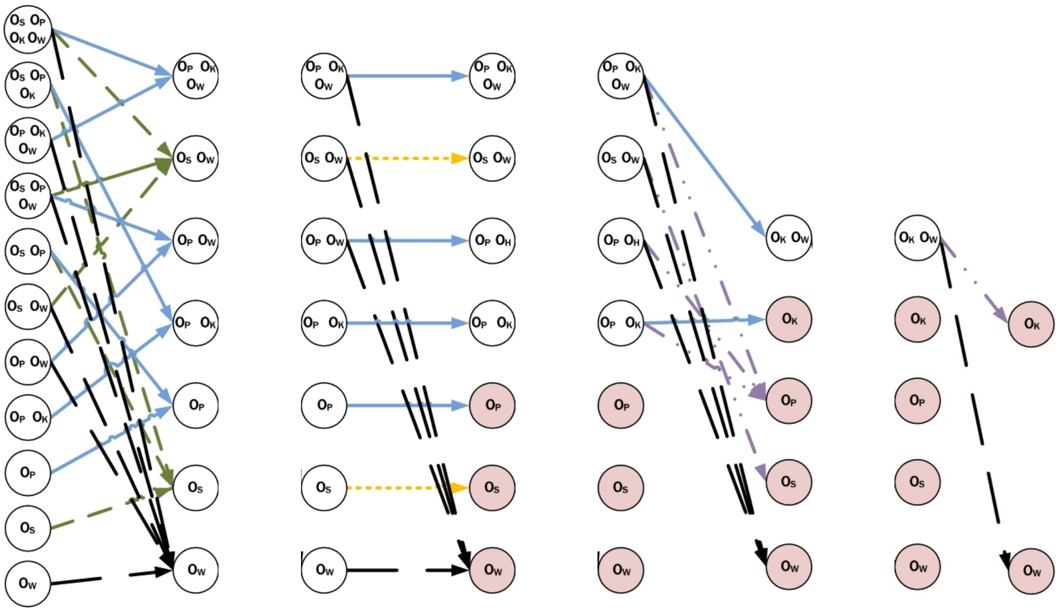
Tangible Entity of Game Client

We define some terms to explain our design in tangible entity(tile). And these terms can be divided into two categories:State and Action .

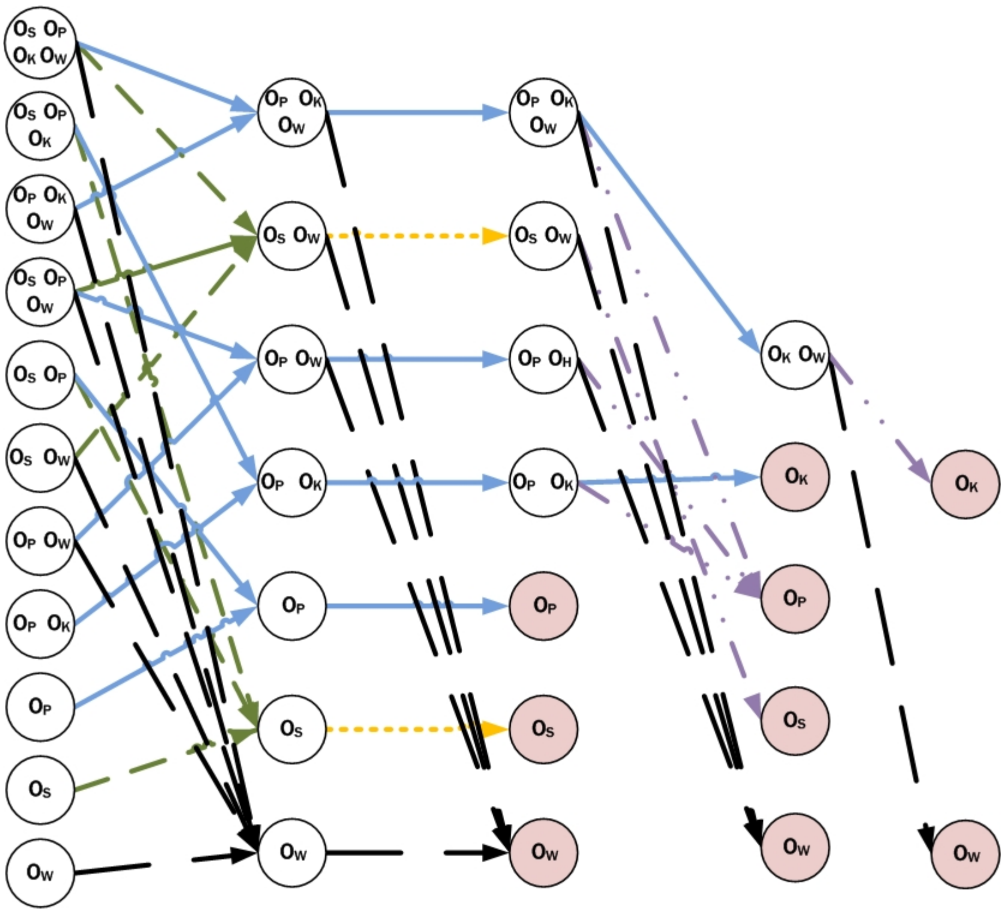
- State

  Despite many kinds of tile state in real mahjong:in player's hand, exposed, concealed, discarded and in draw pile, some states have the same characteristics. Therefore, because of integrating some of tile states, we don't need to define 5 kinds of state, only needing 3 kinds of game state for tangible tile: None, In-hand, Fixed.

  - S_1:None



(a) 1st valid action (b) 2nd valid action (c) 3rd valid action (d) 4th valid action



(e) A combination of (a)˜(d)

Figure 5.6: Judge Mechanism - Bid Mode : Convergence by Valid Messages
When this

If a tile is discarded or in draw pile, we will say the state of the tile is None. When a tile is in draw pile, it is no need to display the tile's image on panel of tangible entity, but it needs to sense the action on the tile and needs to receive the control message through the wireless transmission. On the other hand, after a tangible tile is discarded by local player, we can reuse it as a waiting-drawn tile, i.e. tile in draw pile. That is why we integrate these two states in real mahjong as one state in our TMJ.

– S_2: In-hand

If a tile is in player's hand, we will say the state of the tile is In-hand. When a tangible tile is in local player's hand, it is fundamental to display the tile's image, so the player can manipulate specific tile directly. Since action on tile may occur, it needs sensor to sense the action and needs to return message of the action through the wireless transmission. Therefore, all functionality the tangible tile has is turned on.

– S_3: Fixed

If a tile is exposed or concealed , we say the state of the tile is Fixed. After a tangible tile is exposed by local player, the action on the tile is no longer meaningful, so we can turn off sensor until this game over. Despite of no active message having to return, it must keep wireless transmission to receive the control message. In order to remind local player what he/she exposed, it is necessary to display the tile's image. The same scenario can be applied to a concealed tile, hence, we coordinate these two states in real mahjong as one state in our TMJ.

The functionality of a tile is altered base on which state the tile is, summarized in Table 5.3

Table 5.3: State Definition of Tangible Tile

	Panel	Sensor	Wireless
None	X	O	O
In-Hand	O	O	O
Fixed	O	X	O

- Action

According to Operation of MJ mentioned in Section 4.1.1, we discover just some action types make sense for mahjong game, and we distinguish these actions into 3 types: draw, flip, conceal. Then, we obtain the different 3D vector made by 3-Axis Accelerometer between before and after action.

  - A_1: Draw

    In real world, when a player needs to draw a tile, the action he/she does is turning a face-down tile to face him/her. At this moment, if the drawn tile is our tangible tile, we can discover a change of 3D vector output by sensor from $(0, 0, -1)$ (face-down) to $(1, 0, 0)$ or $(-1, 0, 0)$ or $(0, 1, 0)$ or $(0, -1, 0)$(player-faced). The reason of producing 4 results is that the tile has 4 sides that can be used as shaft. By comparing with the before-action vector and 4 after-action vectors, we can generalize a conclusion: the key to this action is a change of Z, but changes of X and Y are useless, then we use vector $(\times, \times, 0)$ to represent. As a result, if a 3D vector is changed from $(0, 0, -1)$ to $(\times, \times, 0)$, we define this action as Draw.

  - A_2: Flip

    In real world, when a player wants to discard or expose a tile, the action he/she does is turning a tile faced him/her to face up. At this moment, if the drawn tile is our tangible tile, we can discover a change of 3D vector output by sensor from $(1, 0, 0)$ or $(-1, 0, 0)$ or $(0, 1, 0)$ or $(0, -1, 0)$ (player-faced) to $(0, 0, 1)$ (face-up).

Table 5.4: Action Definition of Tangible Tile

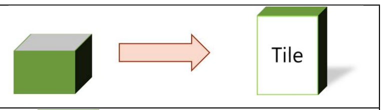
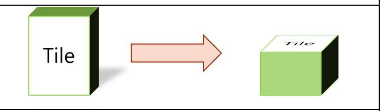
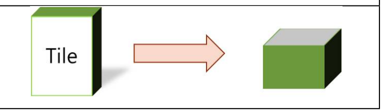
Action	Change of 3D vector	Schematic diagram
Draw	$(0, 0, -1) \longrightarrow (\times, \times, 0)$	
Flip	$(\times, \times, 0) \longrightarrow (0, 0, 1)$	
Conceal	$(\times, \times, 0) \longrightarrow (0, 0, -1)$	

Table 5.5: Action on Tile vs. Tile State

	Draw	Flip	Conceal
None	O	X	X
In-Hand	X	O	O
Fixed	X	X	X

O : meaningful
X : pointless

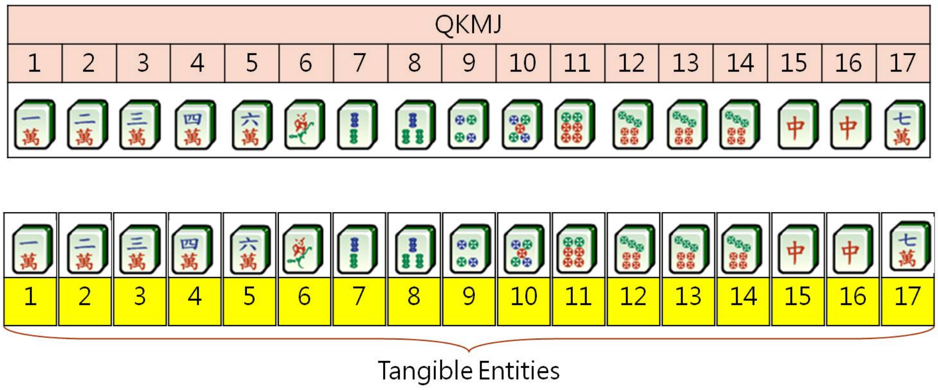
We already recognized that why a tile has 4 kinds of before-action vector, and by comparing these vectors, we generalize the same conclusion with Draw: the key to this action is a change of Z, but changes of X and Y are useless. As a result, if a 3D vector is changed from $(\times, \times, 0)$ to $(0, 0, 1)$, we define this action as Flip.

– A_3: Conceal

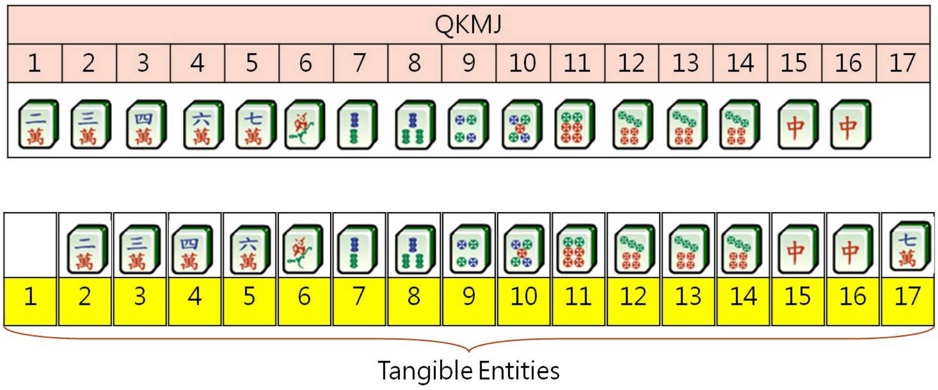
If local player wants to do O_{CK}, the action he/she does is turning a tile faced him/ her to face down. We discover this action is action draw in reverse order, so a change of 3D vector output by sensor is also in reverse order. As a result, if a D vector is changed from $(\times, \times, 0)$ to $(0, 0, -1)$, we can define this action as Conceal.

The action definition mentioned above are summarized in Table 5.4.

Acturally, not all of the action makes sense for each tile's state. For example, it's pointless to do action draw on a tile with state In-hand, or to do action Flip on a tile with state None, etc. Therefore, we summarize these situations and showing in table 5.5. However, even if this action is meaningful, if it happens in wrong time, our TMJ will ignore this action at all.



(a) Before Reordering



(b) After Reordering

Figure 5.7: Inconsistency of Ordering

Messenger of Game Clinet

The messenger in our TMJ is a component we installed to supporting Tangible User Interface. As mentioned in 3.2, the assignment of the messenger is transmitting commands and receiving reports to and from tangible devices. However, there exists inconsistency between the serial number in the worker and multiple tangible devices, because the serial number may be reordered for game logic.

We illustrate the inconsistent situation in Figure 5.7. Figure 5.7a shows the tiles before reordering. For game logic, it is necessary to reorder these tiles since the tile numbered 17 disrupts the tiles ordering. The effect of reordering is shown in Figure 5.7b. As a result, the inconsistency between the worker and tangible devices occurs.

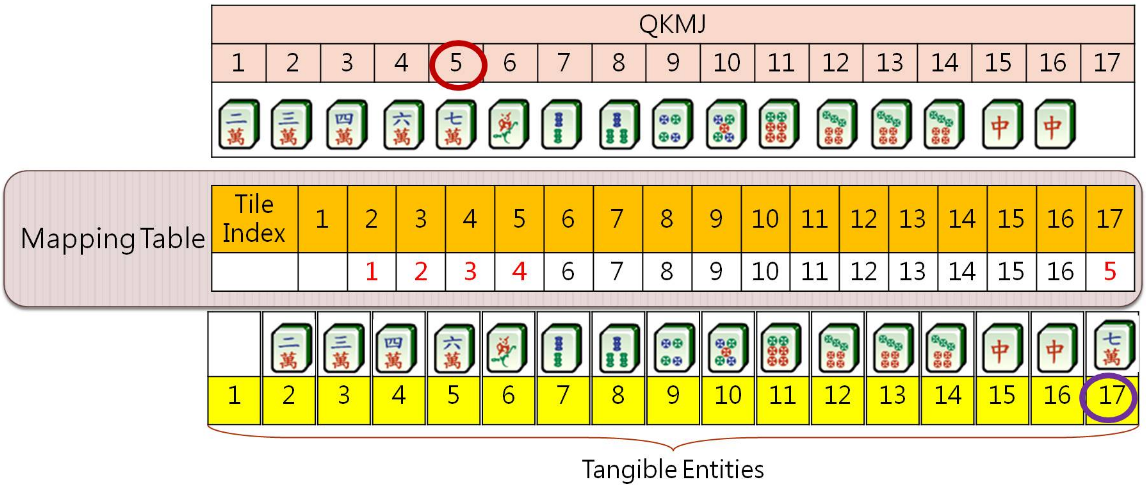


Figure 5.8: Mapping Table

For correct transmitting, we design a mapping table as shown in Figure 5.8 for the messenger to deal with the above inconsistency.

Chapter 6

Conclusion

This thesis adopts the conception of Tangible User Interface. We propose a model cooperating multiple active tangible devices for a networked game. In such a design, multiple tangible devices are installed at the client sides. We revise the client sides, including worker and messenger, to deal with the above changes. The worker deals with the rule of game, while the messenger help convert the relationships between the the virtual identities and physical tangible devices identities. With the proposed model, we ultimately realize networked tangible mahjong game (TMJ) based on QKMJ, and an user of TMJ can play the networked mahjong game with others using either QKMJ or TMJ. With the cooperating system, users can easily and intuitively manipulate what they want to do as daily life.

References

[1] D. R. Olsen, Jr. and E. P. Dempsey, "Syngraph: A graphical user interface generator," *SIGGRAPH Comput. Graph.*, vol. 17, no. 3, pp. 43--50, 1983.

[2] N. Watanabe, Y. Igarashi, and M. Hanaki, "Applying a 3-D-GUI to a distributed network management system," *Selected Areas in Communications, IEEE Journal on DOI - 10.1109/49.842987*, vol. 18, no. 5, pp. 715--722, 2000. [Online]. Available: 10.1109/49.842987

[3] D. C. Smith, C. Irby, R. Kimball, and E. Harslem, "The star user interface: an overview," in *Proceedings of the June 7-10, 1982, national computer conference*. Houston, Texas: ACM, 1982, pp. 515--528. [Online]. Available: http://www.guidebookgallery.org/articles/thestaruserinterfaceanoverview

[4] A. Taylor, "WIMP interfaces," *CS6751 Topic Report: Winter*, vol. 97, 1997. [Online]. Available: http://www.cc.gatech.edu/classes/cs6751_97_winter/Topics/dialog-wimp/

[5] H. Ishii and B. Ullmer, "Tangible bits: towards seamless interfaces between people, bits and atoms," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. Atlanta, Georgia, United States: ACM, 1997, pp. 234--241.

[6] H. Ishii, "Tangible bits: beyond pixels," in *Proceedings of the 2nd international conference on Tangible and embedded interaction*. Bonn, Germany: ACM, 2008, pp. xv--xxv.

[7] "Wii official website." [Online]. Available: http://www.nintendo.co.jp/wii/index.html

[8] C. H. TSAI, "Implementation of emotion teaser using zigbee wireless system," Master's thesis, Department of Electrical Engineering National Cheng Kung University, Tainan, Taiwan R,O,C, 2007. [Online]. Available: http: //etdncku.lib.ncku.edu.tw/ETD-db/ETD-search-c/view_etd?URN=etd-0828107-170424

[9] D. J. Merrill, "Interaction with embodied media," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[10] H. Ishii, "Tangible Bits: Designing the Seamless Interface between People, Bits, and Atoms," in *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 2002, p. 199.

[11] T. Hosokawa, Y. Takeda, N. Shioiri, M. Hirano, and K. Tanaka, "Tangible design support system using rfid technology," in *Proceedings of the 2nd international conference on Tangible and embedded interaction*. Bonn, Germany: ACM, 2008, pp. 75--78.

[12] K. P. Fishkin, "A taxonomy for and analysis of tangible interfaces," *Personal Ubiquitous Comput.*, vol. 8, no. 5, pp. 347--358, 2004.

[13] "Nintendo. official website." [Online]. Available: http://www.nintendo.co.jp/

[14] J. Bernardes, R. Nakamura, D. Calife, D. Tokunaga, and R. Tori, "Integrating the Wii controller with enJine: 3D interfaces extending the frontiers of a didactic game engine," *Comput. Entertain.*, vol. 7, no. 1, pp. 1--19, 2009.

[15] "Bluetooth official website." [Online]. Available: http://www.bluetooth.com/

[16] "Zigbee alliance." [Online]. Available: http://www.zigbee.org/

[17] D. Merrill, J. Kalanithi, and P. Maes, "Siftables: towards sensor network user interfaces,"

in *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*. New York, NY, USA: ACM, 2007, pp. 75--78.

[18] A. Ananthaswamy, "Computing becomes child's play," *The New Scientist*, vol. 201, no. 2696, pp. 18 -- 19, 2009. [Online]. Available: http: //search.ebscohost.com/login.aspx?direct=true&db=ulh&AN=36646853&site=ehost-live

[19] S. Hunter, J. Kalanithi, and D. Merrill, "Make a Riddle and TeleStory: designing children's applications for the siftables platform," in *Proceedings of the 9th International Conference on Interaction Design and Children*. Barcelona, Spain: ACM, 2010, pp. 206--209.

[20] "Mahjong introduction." [Online]. Available: http://www.mahjonged.com/chinese_rules.html

[21] "Mahjong game." [Online]. Available: http://en.wikipedia.org/wiki/Mahjong

[22] "QKMJ." [Online]. Available: http://www.csie.nctu.edu.tw/~sywu/qkmj.html

[23] "MSP-EXP430F5438." [Online]. Available: http://focus.ti.com/docs/toolsw/folders/print/msp-exp430f5438.html

[24] "Data sheet of CC2500." [Online]. Available: http://www.ti.com/lit/SWRS040

[25] "ADXL335." [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/ADXL335.pdf