

การพัฒนาโปรแกรมเกมต่อคำศัพท์ภาษาอังกฤษ
บนเครือข่ายระยะใกล้

DEVELOPMENT OF MULTI PLAYERS SCRABBLE
PROGRAM ON LOCAL AREA NETWORK



จิตรา ลีวารินทร์พานิช
ปิ่นทิพย์ นวธรรมพิเชษฐ์
สิริพร จิตภิรมย์ลาภ

เลขหมู่.....
เลขทะเบียน 43024
วัน, เดือน, ปี 26 ส.ย. 2545

b.....
i.....

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านธุรกิจ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1605

**DEVELOPMENT OF MULTI PLAYERS SCARBBLE
PROGRAM ON LOCAL AREA NETWORK**



**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2001**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ การพัฒนาโปรแกรมเกมต่อคำศัพท์ภาษาอังกฤษบนเครือข่ายระยะใกล้
DEVELOPMENT OF MULTI PLAYERS SCRABBLE PROGRAM ON
LOCAL AREA NETWORK




ชื่อนักศึกษา นางสาวจิตรา ลีวารินทร์พาณิชย์ 41056012
นางสาวปิ่นทิพย์ นวธรรมพิเชษฐ 41056060
นางสาวสิริพร จิตภิรมย์ลาภ 41056118

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2544

	คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	ดร.นันทิกา เบญจเทพานันท์	
กรรมการ	อาจารย์วีระศักดิ์ นิมขุนทด	
กรรมการและอาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์	



(ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การพัฒนาโปรแกรมเกมต่อคำศัพท์ภาษาอังกฤษบนเครือข่ายระยะใกล้	
ชื่อนักศึกษา	นางสาวจิตรา ลีวารินทร์พาณิชย์	41056012
	นางสาวปิ่นทิพย์ นวธรรมพิเชษฐ์	41056060
	นางสาวสิริพร จิตภิรมย์ลาภ	41056118
ปริญญา	วิทยาศาสตร์บัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2544	
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรัักษ์พงษ์	

บทคัดย่อ

การพัฒนาโปรแกรมเกมต่อคำศัพท์ภาษาอังกฤษบนเครือข่ายระยะใกล้นี้ เป็นการพัฒนาโปรแกรมเพื่อสร้างเกมต่อคำศัพท์ที่มีผู้เล่นได้พร้อมกัน 2 ถึง 4 คน บนเครือข่ายระยะใกล้ โดยใช้ภาษาจาวาบนระบบปฏิบัติการวินโดวส์ 98 ติดต่อกันโดยใช้โปรโตคอล TCP/IP เนื่องจากได้สังเกตเห็นถึงความสำคัญของการใช้ภาษาอังกฤษในชีวิตประจำวัน และเกมต่อคำศัพท์ภาษาอังกฤษเป็นสื่อที่เหมาะสมอย่างหนึ่ง โปรแกรมที่พัฒนาขึ้นมีข้อดีคือ ช่วยขจัดปัญหาการจับเก็บอุปกรณ์ สามารถเพิ่มและลบคำศัพท์ได้ตามที่ผู้เล่นต้องการ มีการตรวจความหมายของคำศัพท์ด้วยคอมพิวเตอร์ในทันที และมีการเก็บสถิติที่เป็นคะแนนสูงสุด โดยแบ่งออกเป็น 3 ประเภท คือ สถิติที่มีผู้เล่น 2 , 3 และ 4 คนตามลำดับ

Special Project Title	Development Of Multi Players Scrabble Program On Local Area Network	
Students	Miss Jitra Leevarinpanich	41056012
	Miss Pinthip Nawathamphichet	41056060
	Miss Siriporn Chitphiromlarp	41056118
Degree	Bachelor's Degree of Science	
Department	Mathematics and Computer Science, Faculty of Science	
Programme	Computer Science	
Academic Year	2001	
Special Project Advisor	Assistant Professor Praiboon Pantaragphong	

ABSTRACT

This special project is a development of game in the multi players scrabble on Local Area Network. This game is purpose for 2 to 4 players which is developed by Java Programming language on Windows98 operating system connected with TCP/IP. Because , English is important for living days and the scrabble is one of appropriate thing. The advantages of this program are get rid of the problem about collecting equipments , can add and remove vocabularies by him/herself , check the meaning of vocabularies by computer immediately and record top five of score which separate 3 types (2 , 3 and 4 players).

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องการพัฒนาโปรแกรมเกมต่อคำศัพท์ภาษาอังกฤษบนเครือข่ายระยะใกล้สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ไพโรบลย์ พันธรักษ์พงษ์ อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้ และอาจารย์ชาญชัยแห่งบริษัท Andamansoft ที่ให้ความกรุณาในการตอบสารพันปัญหา จนการทำปัญหาพิเศษนี้สามารถลุล่วงตามวัตถุประสงค์ได้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนการทำปัญหาพิเศษครั้งนี้สำเร็จด้วยดี รวมทั้งพี่ และ เพื่อนๆ ทุกคน ที่ให้ความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ
มีนาคม 2545

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	1
1.4 ขอบเขตของการศึกษา.....	2
1.5 ขั้นตอนในการศึกษา.....	2
1.6 ข้อตกลงเบื้องต้น.....	2
1.7 ข้อจำกัดของการศึกษา.....	3
1.8 คำจำกัดความที่ใช้ในการศึกษา.....	3
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	4
2.1 การติดต่อสื่อสารบนเครือข่ายระยะใกล้.....	4
2.1.1 โพรโตคอล TCP/IP.....	4
2.1.1.1 ประวัติความเป็นมา.....	4
2.1.1.2 โพรโตคอล TCP.....	5
2.1.1.3 โพรโตคอล IP.....	7
2.1.2 ช่องทางของการสื่อสารข้อมูล (Socket).....	10
2.1.2.1 Socket แบบสตรีม.....	10
2.1.2.2 Socket แบบดาต้าแกรม.....	10
2.1.2.3 ขั้นตอนการเปิด Socket โดยภาษาจาวา.....	11

สารบัญ (ต่อ)

	หน้า
2.1.2.4 ขั้นตอนการสร้างสตรีมข้อมูลเข้า (Input Stream)	
โดยภาษาจาวา.....	12
2.1.2.5 ขั้นตอนการสร้างสตรีมข้อมูลออก (Output Stream)	
โดยภาษาจาวา	12
2.1.2.6 ขั้นตอนการปิด Sockets โดยภาษาจาวา.....	14
2.2 โครงสร้างข้อมูล (Data Structure)	15
2.2.1 อาร์เรย์ (Array).....	15
2.3 ส่วนควบคุมการประมวลผลโปรแกรม.....	16
2.3.1 Exceptions.....	16
2.3.1.1 Exception Messages.....	16
2.3.1.2 ตรวจสอบโดยใช้คำสั่ง try, catch และ finally	16
2.3.1.3 Exception Propagation	17
2.3.2 Threads	18
2.3.2.1 การสร้าง Thread.....	18
2.3.2.2 ปัญหาของ Thread	18
2.3.2.3 Synchronization	19
2.3.2.4 การควบคุมการทำงานของ Thread	19
2.4 คลาส Properties	19
2.5 การติดต่อกับส่วนผู้ใช้ (Graphics User Interface : GUI)	20
2.5.1 การเขียนโปรแกรมที่มีการตรวจสอบการทำงานของผู้ใช้ (Events-Driven Programming)	21
2.5.2 เหตุการณ์ (Event)	21
2.5.3 การกำหนดเหตุการณ์ตอบสนอง (Event Handling)	22
2.5.4 Listener.....	23
2.5.5 การตรวจสอบเหตุการณ์ (Handle Event)	24
2.5.6 เหตุการณ์ของเมาส์ (Mouse Events)	25
2.6 กฎกติกาสากลของการเล่นเกมต่อคำศัพท์ภาษาอังกฤษ	26
2.6.1 อุปกรณ์และความหมาย	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.6.2 การเริ่มเกม.....	28
2.6.3 การคิดและคำนวณคะแนน.....	29
2.6.4 การสิ้นสุดเกม และการได้ผู้ชนะ.....	30
2.6.5 ตัวอย่างการคิดคะแนน.....	30
บทที่ 3 การออกแบบโปรแกรม.....	34
3.1 รูปแบบการเชื่อมต่อของคอมพิวเตอร์.....	34
3.2 อัลกอริทึมและแผนผังการทำงาน.....	35
3.2.1 อัลกอริทึมในการหาผู้ชนะ.....	35
3.2.2 แผนผังแสดงการทำงานของระบบ.....	36
3.2.2.1 แผนผังแสดงการทำงานของ Server.....	37
3.2.2.2 แผนผังแสดงการทำงานของ Client.....	38
3.2.2.3 แผนผังแสดงการทำงานของ Play.....	39
3.2.2.4 แผนผังแสดงการทำงานของการเล่น.....	40
3.2.2.5 แผนผังแสดงการทำงานของการเล่น.....	41
ของผู้เล่นแต่ละคน.....	41
บทที่ 4 ประเมินและอภิปรายผล.....	42
4.1 ลักษณะโปรแกรม.....	42
4.2 ข้อกำหนดของการประเมิน.....	43
4.3 การประเมิน.....	43
4.4 ผลการทำงานของโปรแกรม.....	44
4.5 การอภิปรายผล.....	59
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ.....	60
ภาคผนวก ก การติดตั้งโปรแกรมเกมต่อคำศัพท์ภาษาอังกฤษ.....	61
ภาคผนวก ข คำอธิบายส่วนเมนูบาร์ของโปรแกรม.....	64
บรรณานุกรม.....	78

สารบัญตาราง

ตารางที่	หน้า
2-1 Mouse Event Adapter	26



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2-1 รูปแบบของ TCP packet	5
2-2 รูปแบบของ IP ดาต้าแกรมประกอบด้วยส่วน header และ payload	8
2-3 ชนิดของเหตุการณ์.....	21
2-4 Listener Interface	23
2-5 กระดาน SCRABBLE	32
2-6 ตัวอย่างการคิดคะแนนของคำที่ลง	33
3-1 รูปแบบการเชื่อมต่อของคอมพิวเตอร์	34
4-1 จอภาพเพื่อให้ผู้เล่นเลือกสถานะของตน	44
4-2 จอภาพเพื่อให้ผู้เล่นที่เป็น Host ใส่ชื่อของตนและเลือกสัญลักษณ์แทนตนเอง	45
4-3 จอภาพเพื่อให้ผู้เล่นที่เป็น Join ใส่หมายเลข IP ของ Host , ชื่อของตน และเลือกสัญลักษณ์ตนเอง	45
4-4 จอภาพของผู้เล่นฝ่าย Host เพื่อรอรับการติดต่อจาก Join	46
4-5 จอภาพของผู้เล่นฝ่าย Join เพื่อรอให้ผู้เล่นอื่นฝ่าย Join เชื่อมต่อกับ Host และรอให้ Host เป็นผู้กดปุ่มเริ่มเล่นเกม (Start).....	47
4-6 จอภาพเมื่อผู้เล่นฝ่าย Host ทำการเพิ่มคำศัพท์.....	48
4-7 จอภาพเมื่อผู้เล่นฝ่าย Host ทำการลบคำศัพท์.....	48
4-8 จอภาพเพื่อให้ผู้เล่นฝ่าย Host ทำการเลือกประเภทของพจนานุกรม.....	49
4-9 จอภาพเมื่อผู้เล่นฝ่าย Host เลือกใช้พจนานุกรมฉบับเพิ่มเติมคำศัพท์.....	49
4-10 จอภาพของผู้เล่นในขณะที่เล่นเกม โดยในที่นี้ Host เป็นผู้เริ่มเล่นก่อน	50
4-11 จอภาพที่ใช้ในการเปลี่ยนตัวอักษรของผู้เล่น	51
4-12 จอภาพเมื่อมีผู้เล่นคนใดคนหนึ่งทำการเปลี่ยนตัวอักษร	51
4-13 ผลการสุ่มตัวอักษรตัวใหม่ให้กับผู้เล่น	52
4-14 จอภาพเพื่อให้ผู้เล่นยืนยันการผ่านตาเล่น	53
4-15 จอภาพเมื่อผู้เล่นคนใดคนหนึ่งผ่านตาเล่น.....	53
4-16 จอภาพกรณีที่ผู้เล่นนำตัวอักษรไปวางบนกระดาน	54
4-17 จอภาพให้ผู้เล่นกรอกตัวอักษรที่ต้องการแทนตัว Blank	54
4-18 จอภาพเมื่อคำศัพท์ที่ผู้เล่นลงถูกต้อง	55
4-19 จอภาพเพื่อบอกความหมายของคำศัพท์ทุกคำที่เกิดขึ้น	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4-20 จอภาพเพื่อบอกว่าผู้เล่นลงคำศัพท์ไม่ถูกต้อง	56
4-21 จอภาพเมื่อมีผู้เล่นคนใดคนหนึ่งลงคำศัพท์ไม่ถูกต้อง	57
4-22 จอภาพหลังจากเล่นเกมไปได้สักกระยะหนึ่ง	57
4-23 จอภาพสรุปคะแนนเมื่อเกมจบ	58
4-24 จอภาพเพื่อบอกถึงลำดับคะแนนสูงสุด 5 อันดับ	58
ก-1 จอภาพแรกเมื่อเริ่มทำการติดตั้งโปรแกรม	61
ก-2 จอภาพ Introduction	61
ก-3 จอภาพให้เลือกไฟล์เดสก์ท็อปที่ต้องการติดตั้งโปรแกรม Scrabble.exe	62
ก-4 จอภาพให้เลือก Shortcut Folder	62
ก-5 จอภาพที่มีรายละเอียดต่างๆ ของการติดตั้ง	63
ก-6 จอภาพเมื่อทำการติดตั้งโปรแกรม Scrabble.exe สมบูรณ์	63
ข-1 จอภาพเมื่อผู้เล่นเลือกที่เมนู Game	64
ข-2 จอภาพเมื่อผู้เล่นเลือกเมนู Tile	65
ข-3 จอภาพเมื่อผู้เล่นเลือกเมนู View	66
ข-4 จอภาพเมื่อผู้เล่นเลือกเมนู Dictionary	67
ข-5 จอภาพเมื่อผู้เล่นเลือกเมนู Help	68
ข-6 จอภาพเมื่อผู้เล่นเลือกแท็บ Description	69
ข-7 จอภาพเมื่อผู้เล่นเลือกแท็บ Starting the game	70
ข-8 จอภาพเมื่อผู้เล่นเลือกแท็บ Ending the game	71
ข-9 จอภาพเมื่อผู้เล่นเลือกแท็บ Changing the tile	72
ข-10 จอภาพเมื่อผู้เล่นเลือกแท็บ Passing	73
ข-11 จอภาพเมื่อผู้เล่นเลือกแท็บ Playing the word	74
ข-12 จอภาพเมื่อผู้เล่นเลือกแท็บ Forming word	75
ข-13 จอภาพเมื่อผู้เล่นเลือกเมนู Reference	76
ข-14 จอภาพเมื่อผู้เล่นเลือกเมนู About Scrabble	76

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากภาษาอังกฤษเป็นสิ่งจำเป็นสำหรับชีวิตในปัจจุบันเป็นอย่างยิ่ง ดังนั้นจึงควรมีสื่อช่วยในการพัฒนาภาษาอังกฤษสำหรับเยาวชน สื่ออย่างหนึ่งที่ช่วยในการพัฒนาทางภาษาได้คือ เกมต่อคำศัพท์ภาษาอังกฤษ แต่เนื่องจากเกมต่อคำศัพท์ภาษาอังกฤษประกอบด้วยอุปกรณ์ต่างๆ มากมาย ทั้งกระดานเล่น , แท่นวางตัวอักษรที่สุมจับได้ (Rack) รวมทั้งตัวอักษรในภาษาอังกฤษทั้ง 26 ตัวที่แต่ละตัวมีจำนวนไม่เท่ากัน การเก็บรักษาจึงทำได้ยาก เพราะหากอุปกรณ์ใดหายไปแม้เพียงตัวอักษร 1 ตัวจะไม่สามารถเล่นเกมได้ อีกทั้งการที่ผู้เล่นจะตรวจสอบคำศัพท์ของตนและฝ่ายตรงข้ามว่าถูกต้องหรือไม่นั้นเป็นเรื่องที่ค่อนข้างจะไม่สะดวก จึงเกิดแนวความคิดว่าหากนำเกมต่อคำศัพท์ภาษาอังกฤษมาใส่ลงในคอมพิวเตอร์ได้ จะทำให้ไม่ต้องเก็บรักษาอุปกรณ์การเล่น รวมถึงการตรวจสอบคำศัพท์ยังสามารถทำได้ง่ายโดยให้คอมพิวเตอร์เป็นผู้ตรวจ รวมทั้งจะได้มีการช่วยเพิ่มพูนความรู้ทางด้านคำศัพท์ภาษาอังกฤษอีกด้วย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

เพื่อพัฒนาโปรแกรมคอมพิวเตอร์สำหรับเล่นเกมต่อคำศัพท์ภาษาอังกฤษที่รองรับผู้เล่นตั้งแต่ 2 ถึง 4 คนบนเครือข่ายระยะใกล้ (Local Area Network : LAN) ที่เชื่อมต่อกันโดยโปรโตคอล TCP/IP

1.3 สมมติฐานของการศึกษา

สร้างโปรแกรมเกมต่อคำศัพท์ภาษาอังกฤษที่มีผู้เล่นตั้งแต่ 2 ถึง 4 คน ผ่านเครือข่ายระยะใกล้ด้วยภาษาจาวาบนระบบปฏิบัติการวินโดวส์ 98 ที่ติดต่อกันโดยใช้โปรโตคอล TCP/IP โดยโปรแกรมสามารถทำการตรวจสอบความถูกต้องของคำศัพท์และแสดงความหมายของคำศัพท์เหล่านั้นที่หน้าจอของผู้เล่นทุกคน

1.4 ขอบเขตของการศึกษา

เป็นเกมที่มีผู้เล่นตั้งแต่ 2 ถึง 4 คน โดยแบ่งผู้เล่นเป็น 2 ประเภท คือ ผู้เล่นที่ทำหน้าที่เปรียบเหมือน Server เรียกว่า "Host" จำนวน 1 คน และ ผู้เล่นที่ทำหน้าที่เปรียบเหมือน Client เรียกว่า "Join" จำนวน 1 ถึง 3 คน โดย Join จะต้องขอสร้างการติดต่อไปยัง Host แล้วรอจนกว่า Host จะรับการติดต่อจึงจะสามารถเล่นได้ การติดต่อทำโดย Join ใส่หมายเลข IP ของ Host และ Host สามารถเพิ่มคำศัพท์ได้ตามที่ต้องการ ผู้เล่นสามารถเปลี่ยนตัวอักษรได้คราวละ 1 ถึง 7 ตัว , กำหนดให้ตัว Blank เป็นตัวอักษรใดก็ได้ตามแต่ผู้เล่นต้องการ และสามารถผ่านตาเล่นนั้น หากผู้เล่นไม่สามารถลงคำศัพท์ใดๆ ได้ คอมพิวเตอร์จะทำการตรวจสอบคำศัพท์ที่ผู้เล่นลงในทันที หากถูกต้องจะทำการคิดคะแนนทั้งหมดที่เกิดจากการลงคำศัพท์คำนั้น แต่หากมีคำใดคำหนึ่งไม่ถูกต้อง คอมพิวเตอร์จะคืนตัวอักษรที่ผู้เล่นลง และผู้เล่นคนนั้นจะเสียตาเล่นครั้งนั้นไป

1.5 ขั้นตอนในการศึกษา

- 1) ศึกษารูปแบบและวิธีการเล่นเกม
- 2) ออกแบบและวางโครงสร้างเกม
- 3) ศึกษาโปรแกรมภาษาจาวา
- 4) ศึกษาและทดลองวิธีการเชื่อมต่อคอมพิวเตอร์ผ่านเครือข่าย
- 5) ออกแบบกระดาน และตัวอักษร รวมทั้งส่วนติดต่อกับผู้ใช้
- 6) เขียนโปรแกรม
- 7) ทดสอบและแก้ไขข้อผิดพลาด
- 8) สรุปโครงงานและจัดทำเอกสารประกอบปัญหาพิเศษ

1.6 ข้อตกลงเบื้องต้น

- 1) ทำงานภายใต้ระบบปฏิบัติการวินโดวส์ 98 หรือสูงกว่า
- 2) หน่วยประมวลผลกลาง (CPU) ความเร็ว Pentium II 350 เมกะเฮิร์ต หรือสูงกว่า
- 3) คอมพิวเตอร์เชื่อมต่อบนเครือข่ายระยะใกล้ผ่านโปรโตคอล TCP/IP จำนวน 2 ถึง 4 เครื่อง
- 4) ใช้คอมไพเลอร์ภาษาจาวา (JDK 1.2 ขึ้นไป)
- 5) หน่วยความจำหลักอย่างน้อย 128 เมกะไบต์
- 6) มีพื้นที่ในหน่วยความจำสำรองอย่างน้อย 12 เมกะไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 ข้อจำกัดของการศึกษา

เมื่อผู้เล่นกดปุ่ม “Done” เพื่อยืนยันการลงในแต่ละตาเล่น คอมพิวเตอร์ใช้ระยะเวลาในการประมวลผลนาน

1.8 คำจำกัดความที่ใช้ในการศึกษา

Blank หมายถึง ตัวว่างที่สามารถกำหนดให้เป็นตัวอักษรใดๆ ในภาษาอังกฤษก็ได้ แต่จะมีค่าคะแนนเป็นศูนย์

Bingo หมายถึง ผู้เล่นสามารถลงตัวอักษรทั้ง 7 ตัวในแป้นพร้อมกัน ทำให้เกิดคำใหม่ในตาเล่นนั้นๆ ไม่ว่าจะคำที่เกิดขึ้นใหม่จะประกอบด้วยตัวอักษรที่มีอยู่แล้วในกระดานก็ตัวก็ตาม



บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 การติดต่อสื่อสารบนเครือข่ายระยะไกล

2.1.1 โพรโตคอล TCP/IP

โพรโตคอล (Protocol) คือ ระเบียบวิธีที่กำหนดขึ้นสำหรับสื่อสารข้อมูลให้สามารถส่งผ่านข้อมูลไปยังปลายทางได้อย่างถูกต้อง ในปัจจุบันโพรโตคอลในการสื่อสารข้อมูลมีอยู่หลายโพรโตคอล เช่น โพรโตคอล TCP/IP , โพรโตคอล IPX/SPX , โพรโตคอล NetBIOS และ โพรโตคอล AppleTalk เป็นต้น

2.1.1.1 ประวัติความเป็นมา

TCP/IP เป็นมาตรฐานของการรับส่งข้อมูลระหว่างคอมพิวเตอร์ที่มีจุดเริ่มต้นมาจากการที่กระทรวงกลาโหมของประเทศสหรัฐอเมริกาต้องการเชื่อมโยงคอมพิวเตอร์ทางทหารของแต่ละหน่วย ซึ่งเป็นเครื่องคอมพิวเตอร์ต่างชนิดกันให้สามารถติดต่อกันได้ โครงการนี้มีชื่อว่า ARPANET (Advanced Research Projects Agency Network) ซึ่งการรับส่งข้อมูลจะแบ่งข้อมูลออกเป็นส่วนย่อยๆ เรียกว่า “แพ็กเก็ต” (Packet) ข้อมูลแต่ละส่วนนี้จะถูกส่งไปให้คอมพิวเตอร์ผู้รับปลายทางผ่านสายส่งข้อมูล โดยแต่ละส่วนอาจใช้เส้นทางในการส่งข้อมูลคนละทางก็ได้ คอมพิวเตอร์ปลายทางจะนำข้อมูลที่ได้รับมาต่อรวมกันตามลำดับจนครบ หากเส้นทางที่ส่งข้อมูลเสียหายหรือเครื่องคอมพิวเตอร์บางส่วนในเครือข่ายเสียหาย ข้อมูลจะถูกส่งไปใหม่โดยใช้เส้นทางอื่นโดยอัตโนมัติ

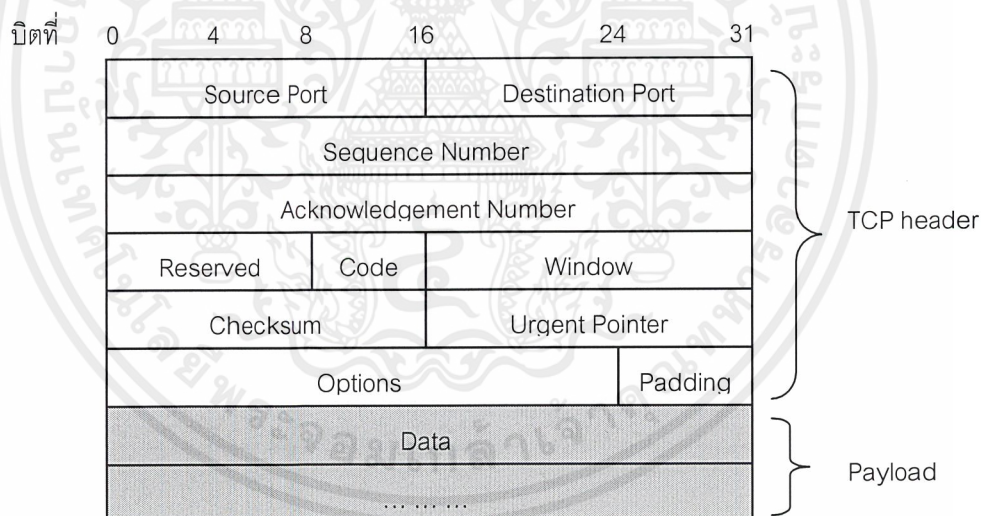
ซอฟต์แวร์ที่ใช้ควบคุมการรับส่งข้อมูลของ ARPANET ประกอบด้วย ส่วนหลักๆ 2 ส่วน คือ Transmission Control Protocol หรือ TCP และ Internet Protocol หรือ IP ซึ่ง TCP มีหน้าที่ตรวจสอบการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ให้ได้รับข้อมูลถูกต้องครบถ้วน ส่วน IP จะมีหน้าที่เลือกเส้นทางที่ใช้รับส่งข้อมูลผ่านระบบเครือข่าย ในปี ค.ศ. 1983 TCP/IP ถูกกำหนดให้เป็นมาตรฐานการรับส่งข้อมูลของกระทรวงกลาโหมประเทศสหรัฐอเมริกา และได้ถูกรวมเข้าเป็นส่วนหนึ่งของระบบปฏิบัติการ UNIX

2.1.1.2 โพรโตคอล TCP

โพรโตคอล TCP (Transmission Control Protocol) เป็นโพรโตคอลที่มีการรับส่งข้อมูลแบบไม่คำนึงถึงปริมาณข้อมูลที่จะส่งไป แต่จะแบ่งข้อมูลเป็นส่วนย่อยๆ ก่อน แล้วจึงจะส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล (Stream Oriented Protocol) ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งสูญหายไป ก็จะมีส่งข้อมูลส่วนนั้นใหม่อีกครั้ง ส่วนปลายทางก็จะทำหน้าที่จัดเรียงส่วนของข้อมูลตามลำดับใหม่ให้ต่อเนื่องกันและประกอบกลับเป็นข้อมูลทั้งหมด ซึ่งจะทำการแยกข้อมูลที่ส่งไปไม่ถูกต้องออก

การติดต่อกันจะเป็นแบบต้องมีการสร้างการติดต่อเป็น Session ทั้ง 2 ด้านเสียก่อน แล้วจึงจะรับส่งข้อมูลไปได้พร้อมกัน (Full Duplex) และเมื่อจะเลิกการติดต่อก็จะต้องมีการส่งสัญญาณแจ้งให้ทราบทั้ง 2 ด้านด้วย (Connection Oriented)

ในระหว่างการรับส่งข้อมูล โพรโตคอล TCP จะมีกระบวนการตรวจสอบข้อมูลเพื่อให้ข้อมูลมีความถูกต้อง โดยการส่งสัญญาณสอบทานข้อมูล (Acknowledgement) และส่งข้อมูลให้ใหม่อีกครั้ง ถ้าปลายทางไม่ได้รับหรือเกิดความผิดพลาดขึ้น



รูปที่ 2-1 แสดงรูปแบบของ TCP packet

ส่วนของ TCP header มีการแบ่งย่อยเพื่อระบุพารามิเตอร์ในการใช้งานต่างๆ ดังนี้

- Source Port

มีขนาด 16 บิต บอกหมายเลขพอร์ตของสถานีต้นทาง

- Destination Port

มีขนาด 16 บิต บอกหมายเลขพอร์ตของสถานีปลายทาง

- Sequence Number

มีขนาด 32 บิต บอกตำแหน่งไบต์แรกของข้อมูลที่ปรากฏอยู่ในส่วน

ของเขต Data

- Acknowledgement Number

มีขนาด 32 บิต โพรโตคอล TCP ใช้หลักการของ Positive ACK ในการตอบกลับไปยังต้นทางเพื่อบอกว่าปลายทางได้รับข้อมูลเรียบร้อยแล้ว โดยหมายเลขของ Acknowledge Number จะมีค่าเท่ากับ ไบต์สุดท้าย บวกหนึ่ง ของข้อมูลที่ได้รับ

- Reserved

มีขนาด 4 บิต เป็นส่วนที่ถูกสำรองไว้ใช้ในอนาคต

- Code

ประกอบด้วย 6 เขตย่อย ซึ่งแต่ละเขตย่อยมีขนาด 1 บิต

มีรายละเอียดดังนี้

- เขตย่อย URGent ถ้าบิตนี้เป็น 1 หมายความว่าเขต Urgent Pointer บรรจุข่าวสารที่มีความหมายและต้องนำมาตีความด้วย
- เขตย่อย ACKnowledgement ถ้าบิตนี้เป็น 1 หมายความว่าเขต Acknowledgement Number มีความหมายต้องนำมาพิจารณาในการตรวจสอบลำดับในการรับส่งข้อมูลด้วย
- เขตย่อย PUSH ถ้าบิตนี้เป็น 1 หมายความว่าทันทีที่สถานีปลายทางรับ TCP เซกเมนต์ได้ ต้องรับส่งข้อมูลไปยังแอปพลิเคชันทันที
- เขตย่อย RESET ถ้าบิตนี้เป็น 1 หมายความว่ามีการผิดพลาดเกิดขึ้นระหว่างคู่สถานีที่มีการเชื่อมต่อถึงกันอยู่
- เขตย่อย SYNchronize ถ้าบิตนี้เป็น 1 หมายความว่าขอเริ่มต้นการเชื่อมต่อระหว่างสถานี
- เขตย่อย FIN ถ้าบิตนี้เป็น 1 หมายความว่าขอยกเลิกการเชื่อมต่อระหว่างสถานี

- Window

มีขนาด 16 บิต สถานีปลายทางใช้เขตนี้ในการบอกถึงขนาดของบัฟเฟอร์ (จำนวนไบต์) ที่สามารถรับข้อมูลจากต้นทางได้

- Checksum

มีขนาด 16 บิต เป็นเขตที่ใช้ในการตรวจสอบความผิดพลาดเฉพาะส่วนของ TCP header

- Urgent Pointer

มีขนาด 16 บิตบอกถึงตำแหน่งของไบต์สุดท้ายภายใน TCP เซกเมนต์ที่เป็นข้อมูลเร่งด่วน ค่าที่บรรจุในเขตนี้จะมีความหมายก็ต่อเมื่อเขตย่อย Urgent เป็น 1

- Options

เป็นฟิลด์เก็บข้อมูลที่มีขนาดไม่แน่นอน ใช้สำหรับกำหนดค่าพารามิเตอร์ส่วนประกอบปลีกย่อยซึ่งส่วนใหญ่ไม่มีการนำไปใช้งาน

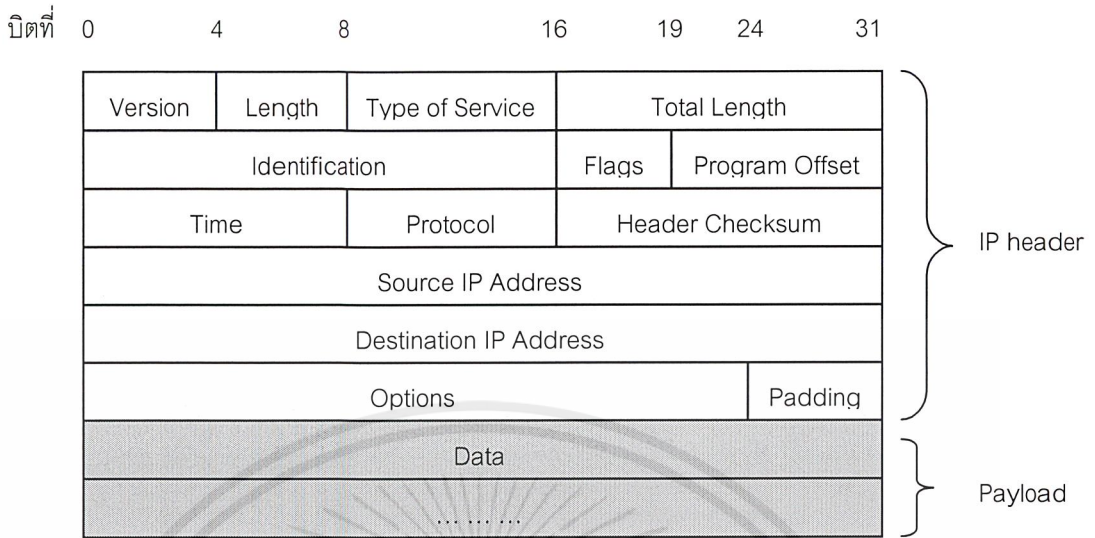
- Padding

ทำหน้าที่เป็นส่วนข้อมูลเติมเต็มเพื่อให้ TCP header เต็มครบ 32 ไบต์ ซึ่งเป็นผลมาจาก Options ที่มีขนาดไม่แน่นอนนั่นเอง

2.1.1.3 โพรโตคอล IP

โพรโตคอล IP (Internet Protocol) ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจาก Host-to-Host Layer ของ TCP/IP สแต็ก เพื่อส่งข้ามไปยังเครือข่ายใดๆ ได้อย่างถูกต้อง โดยจะทำงานแบบมีการส่งข้อมูลผ่านสวิทช์ไปยังปลายทาง ข้อมูลจะเดินทางไปยังเครือข่ายต่างๆ ผ่านสวิทช์ไปเรื่อยๆ จนกว่าจะถึงปลายทาง (Packet Switching)

การติดต่อกันจะเป็นแบบที่ผู้ส่งข้อมูลไม่ต้องตรวจสอบว่าผู้รับพร้อมรับข้อมูลหรือไม่ และการส่งข้อมูลไปให้จะไม่มีการตรวจสอบว่าผู้รับได้รับข้อมูลหรือไม่ (Connectless)



รูปที่ 2-2 แสดงรูปแบบของ IP ดาต้าแกรมประกอบด้วยส่วน header และ payload

ดังนี้

ส่วนของ IP header มีการแบ่งย่อยเพื่อระบุพารามิเตอร์ในการทำงานต่างๆ

IPv4 (IP เวอร์ชัน 4)

- Version

มีขนาด 4 บิต ถูกกำหนดค่าเป็น 4 ในกรณีที่ให้หมายเลข IP เป็น

- Length

มีขนาด 4 บิต เป็นค่าความยาวของ IP header นี้

- Type of Service

มีขนาด 8 บิต บอกให้ทราบว่าจะดำเนินการกับข้อมูลนี้อย่างไร เช่น low delay, high throughput เป็นต้น แต่ในความเป็นจริงตัวเราเตอร์ (Router) จะไม่สนใจข้อมูลนี้

- Total Length

มีขนาด 16 บิต เก็บข้อมูลแสดงค่าความยาวสุทธิของ IP ดาต้าแกรม เป็นจำนวนไบต์

- Identification

เป็นข้อมูลที่บอกให้ทราบว่า IP ดาต้าแกรมนั้นมาจากที่ใด โดยเฉพาะกรณีที่ข้อมูลถูกแยกออกเป็นส่วนๆ แล้ว

- Flags และ Fragment Offset

เป็นข้อมูลที่ใช้ระบุการแยกและรวมข้อมูลเพื่อให้ข้อมูลที่ถูกรวมออกเป็นข้อมูลย่อย (fragment) สามารถกลับมารวมกันใหม่ตามลำดับได้ถูกต้อง

- Time หรือ Time to Live

เป็นข้อมูลแสดงจำนวนเวลามากที่สุดของ IP ดาต้าแกรม ซึ่งสามารถจะส่งผ่านเครือข่ายไปยังปลายทางได้ โดยมีหน่วยเป็นวินาที และตามปกติจะมีค่าเป็น 32

- Protocol

เป็นข้อมูลการระบุโปรโตคอลที่ทำงานในเลเยอร์ (Layer) ข้างบนซึ่งผ่นลงมาใน IP ดาต้าแกรม

- Header Checksum

เป็นส่วนที่ใช้ตรวจสอบความถูกต้องของข้อมูลเฉพาะในส่วนของ IP header โดยไม่เกี่ยวกับส่วนของ Payload ซึ่งการตรวจสอบความถูกต้องของข้อมูลนี้ โปรโตคอล IP จะทำหน้าที่ในการคำนวณและตรวจสอบ โดยกรณีที่เกิดความผิดพลาดของข้อมูล IP ดาต้าแกรมนั้น จะถูกยกเลิกหรือไม่รับข้อมูลมาใช้งาน

- Source IP Address

เป็นส่วนเก็บข้อมูลของหมายเลข IP ต้นทางที่ IP ดาต้าแกรมนั้นถูกส่งมา

- Destination IP Address

เป็นส่วนเก็บข้อมูลของหมายเลข IP ปลายทางที่เป็นผู้รับข้อมูล IP ดาต้าแกรมนั้น

- Options

เป็นฟิลด์เก็บข้อมูลที่มีขนาดไม่แน่นอน ใช้สำหรับกำหนดค่าพารามิเตอร์ส่วนประกอบปลีกย่อย ซึ่งส่วนใหญ่ไม่มีการนำไปใช้งาน

- Padding

ทำหน้าที่เป็นส่วนข้อมูลเติมเต็มเพื่อให้ TCP header เต็มครบ 32 ไบต์ ซึ่งเป็นผลมาจาก Options ที่มีขนาดไม่แน่นอนนั่นเอง

2.1.2 ช่องทางของการสื่อสารข้อมูล (Socket)

Socket ต้องมีประเภท (type) และต้องรวมอยู่ในกระบวนการ (process) ที่กำลังทำงานอยู่ โดยทั่วไป Socket จะสื่อสารข้อมูลกับ Socket ตัวอื่นๆ (ที่กำลังทำงานอยู่) ในช่องของการสื่อสารเดียวกัน (Communication Domain) เดียวกัน โดยใช้ IP ช่วย Socket สามารถแบ่งได้เป็น 2 ประเภท คือ

- Socket แบบสตรีม

เป็น Socket ที่ทำงานกับข้อมูลที่จะส่งไปโดยไม่กำหนดขอบเขตของเรคอร์ดหรือไม่เป็นเรคอร์ด (ข้อมูลที่ส่งเรียกว่า A stream of byte) และสตรีมจะรับประกันว่าข้อมูลถูกส่งถึงปลายทางอย่างถูกต้อง โดยข้อมูลไม่สลับกันและไม่มีการส่งซ้ำซ้อน

- Socket แบบดาต้าแกรม

เป็น Socket ที่สนับสนุนการส่งข้อมูลเป็นเรคอร์ดแต่ไม่รับประกันว่าข้อมูลได้ส่งไปถึงปลายทางและข้อมูลอาจจะมีการสลับกันหรืออาจมีการส่งข้อมูลซ้ำซ้อนกันได้

2.1.2.1 Socket แบบสตรีม

Socket แบบสตรีม ใช้ในการส่งข้อมูลในลักษณะที่ไม่เป็นเรคอร์ด ซึ่งสตรีมของไบนารีสามารถส่งได้ทั้งสองทาง (ส่งและรับใน Socket เดียวกัน) สตรีมนี้สามารถเชื่อถือได้ว่าข้อมูลจะถูกส่งอย่างถูกต้อง ไม่สลับกัน ไม่ซ้ำ และส่งไปถึงปลายทางอย่างแน่นอน โดยสตรีมยังสามารถจัดการส่งและรับข้อมูลขนาดใหญ่ได้ดีอีกด้วย เพราะโปรโตคอลใน Network Layer อาจแตกข้อมูลให้อยู่ในขนาดที่เหมาะสม

การสื่อสารข้อมูลแบบสตรีมนั้นเป็นการสื่อสารข้อมูลแบบต้องมีการสร้างการติดต่อเป็น Session ทั้ง 2 ด้านก่อน สตรีมนั้นต้องมีการเชื่อมต่ออย่างชัดเจน ถ้า Socket A ต้องการเชื่อมต่อกับ Socket B จะต้องยอมรับหรือปฏิเสธการร้องขอนั้นๆ เหมือนกับการโทรศัพท์ซึ่งต้องมีผู้เรียก แล้วผู้รับจะต้องรับสายจึงจะคุยกันได้ และการสนทนาจะไม่มีเสียงซ้ำกัน (ถ้าผู้พูดไม่ได้พูดซ้ำ) และลำดับของเสียงที่พูดจะตรงกับลำดับของเสียงที่ผู้ฟังได้ยิน และที่สำคัญที่สุดคือ เสียงที่ผู้พูดพูดไปนั้นจะถูกส่งไปถึงผู้รับอย่างแน่นอน ด้วยเหตุนี้จึงทำให้ Socket แบบสตรีมเป็นที่นิยมมากกว่า Socket แบบดาต้าแกรม

2.1.2.2 Socket แบบดาต้าแกรม

Socket แบบดาต้าแกรม สนับสนุนการส่งข้อมูลสองทาง โดยไม่รับประกันว่าข้อมูลจะถูกส่งไปถึงปลายทางอย่างถูกต้องหรือไม่ และข้อมูลอาจมีการสลับกันหรือซ้ำซ้อน แต่

ขนาดของเรคอร์ดจะได้รับการปกป้องรักษา แม้ว่าขนาดของเรคอร์ดจะใหญ่กว่าขนาดที่ผู้รับกำหนดก็ตาม

การสื่อสารข้อมูลแบบดาต้าแกรมเป็นการสื่อสารข้อมูลแบบไม่จำเป็นต้องสร้างการเชื่อมต่อโดยชัดเจน เพียงส่งข้อความดาต้าแกรม (Datagram Message) ไปยัง Socket ที่กำหนด ก็สามารถสื่อสารข้อมูลกันได้แล้ว และ Socket แบบดาต้าแกรมยังมีความสามารถในการส่งข้อมูลแบบกระจาย (Broadcasting Message)

2.1.2.3 ขั้นตอนการเปิด Socket โดยภาษาจาวา

ฝั่ง Client

เปิด Socket ได้โดยมีส่วนของโปรแกรมดังนี้

```
Socket MyClient;
try{
    MyClient = new Socket("Machine name", PortNumber);
}
catch (IOException (e){
    System.out.println(e);
}
```

ฝั่ง Server

เปิด Socket ได้โดยมีส่วนของโปรแกรมดังนี้

```
ServerSocket MyService;
try{
    MyService = new ServerSocket(PortNumber);
}
catch (IOException (e){
    System.out.println(e);
}
```

2.1.2.4 ขั้นตอนการสร้างสตรีมข้อมูลเข้า (Input Stream) โดยภาษาจาวา

ฝั่ง Client

สามารถใช้ คลาส DataInputStream ในการสร้างสตรีมข้อมูลเข้าเพื่อรับการตอบสนองจาก Server โดยมีส่วนของโปรแกรมดังนี้

```
DataOutputStream input;
try{
    input = new DataInputStream(MyClient.getInputStream());
}
catch (IOException e){
    System.out.println(e);
}
```

ฝั่ง Server

ใช้ คลาส DataInputStream ในการสร้างสตรีมข้อมูลเข้าเพื่อรับข้อมูลเข้าจาก Client โดยมีส่วนของโปรแกรมดังนี้

```
DataInputStream input;
try{
    input = new DataInputStream(ServiceSocket.
getInputStream());
}
catch (IOException e){
    System.out.println(e);
}
```

2.1.2.5 ขั้นตอนการสร้างสตรีมข้อมูลออก (Output Stream) โดยภาษาจาวา

ฝั่ง Client

สร้างสตรีมข้อมูลออกเพื่อส่งข้อมูลไปที่ Server โดยใช้คลาส PrintStream หรือ DataOutputStream ของ java.io โดยมีส่วนของโปรแกรมดังนี้

```

PrintStream output;
try{
    output = new PrintStream(MyClient.getOutputStream());
}
catch (IOException e) {
    System.out.println(e);
}

```

แต่ถ้าใช้คลาส `DataOutputStream` จะมีส่วนของโปรแกรมดังนี้

```

DataOutputStream output;
try{
    output = new DataOutputStream(MyClient.getOutputStream
());
}
catch (IOException e) {
    System.out.println(e);
}

```

ฝั่ง Server

ใช้คลาส `PrintStream` เพื่อส่งข้อมูลไปให้กับ Client โดยมีส่วนของโปรแกรม

```

PrintStream output;
try{
    output = new PrintStream(serviceSocket.getOutputStream());
}
catch (IOException e){
    System.out.println(e);
}

```

ดังนี้

2.1.2.6 ขั้นตอนการปิด Sockets โดยภาษาจาวา

เราจะต้องทำการปิดสตรีมข้อมูลเข้าและสตรีมข้อมูลออกก่อน แล้วจึงจะทำการปิด Socket ดังตัวอย่างต่อไปนี้

ฝั่ง Client

โดยมีส่วนของโปรแกรมดังนี้

```
try {
    output.close();
    input.close();
    MyClient.close();
}
catch (IOException) {
    System.out.println(e);
}
```

ฝั่ง Server

โดยมีส่วนของโปรแกรมดังนี้

```
try {
    output.close();
    input.close();
    serviceSocket.close();
    MyService.close();
}
catch (IOException) {
    System.out.println(e);
}
```

2.2 โครงสร้างข้อมูล (Data Structure)

การจัดแบ่งข้อมูลสามารถทำได้หลายวิธีแตกต่างกัน โดยการจัดโครงสร้างข้อมูลเชิงคณิตศาสตร์หรือ ลอจิก (Logic) ของข้อมูลนั้น เราจะเรียกว่าโครงสร้างข้อมูล การเลือกรูปแบบการจัดเก็บข้อมูลมีหลักการพิจารณา ดังนี้

1. ต้องมีโครงสร้างที่สมบูรณ์เพียงพอที่จะแสดงให้เห็นถึงความสัมพันธ์ของข้อมูลในปัจจุบัน
2. โครงสร้างที่จะนำมาจัดเก็บข้อมูลจะต้องง่ายต่อการดำเนินการ

2.2.1 อาร์เรย์ (Array)

โครงสร้างของข้อมูลสามารถแบ่งเป็น 2 แบบ คือ แบบต่อเนื่อง และ แบบไม่ต่อเนื่อง (Linear and Nonlinear) โครงสร้างข้อมูลที่เป็นแบบต่อเนื่องจะมีการเก็บข้อมูลแต่ละอิลิเมนต์ต่อเนื่องกันหรือเรียกว่าเป็นรายการต่อเนื่อง โครงสร้างข้อมูลแบบต่อเนื่องที่เก็บในหน่วยความจำนั้น สามารถแสดงความต่อเนื่องได้ 2 แบบ

1. ใช้ตำแหน่งของหน่วยความจำที่ต่อเนื่องกันในการเก็บข้อมูล ซึ่งโครงสร้างแบบนี้ เรียกว่า อาร์เรย์
2. แสดงความต่อเนื่องกันโดยใช้พอยน์เตอร์ (Pointer) หรือ ลิงค์ ซึ่งจะเรียกโครงสร้างข้อมูลแบบนี้ว่า ลิงค์ลิสต์ (Linked-List)

สำหรับข้อมูลแบบไม่ต่อเนื่องได้แก่ โครงสร้างแบบทรี และกราฟ ซึ่งจะไม่กล่าวถึงในที่นี้

การทำงานกับโครงสร้างแบบต่อเนื่อง ที่เป็นอาร์เรย์หรือลิงค์ลิสต์จะมีดังต่อไปนี้

1. การเข้าถึงข้อมูล (Traversal) เป็นการเข้าไปติดต่อกับข้อมูลในแต่ละอิลิเมนต์ในลิสต์
2. การค้นหา (Search) เป็นการหาตำแหน่งของอิลิเมนต์ หรือ เรคอร์ดที่กำหนดคีย์ให้
3. การแทรก (Insertion) เป็นการเพิ่มข้อมูลใหม่เข้าไปในลิสต์
4. การลบ (Deletion) เป็นการนำข้อมูล 1 อิลิเมนต์ออกจากลิสต์
5. การจัดเรียง (Sorting) เป็นการจัดเรียงข้อมูลตามลำดับที่ต้องการ

โครงสร้างที่เลือกใช้จะขึ้นอยู่กับความถี่ของการทำงานเหล่านี้ ในที่นี้จะเป็นการกล่าวถึงอาร์เรย์ เนื่องจากอาร์เรย์ง่ายต่อการเข้าถึงข้อมูล ค้นหาและจัดเรียง อาร์เรย์จะใช้เก็บข้อมูลที่มีความสัมพันธ์ของข้อมูลคงที่ แต่ถ้าหากขนาดของโครงสร้างข้อมูลเปลี่ยน โครงสร้างข้อมูลแบบอาร์เรย์อาจไม่เหมาะเท่ากับโครงสร้างแบบลิงค์ลิสต์

2.3 ส่วนควบคุมการประมวลผลโปรแกรม

2.3.1 Exceptions

Exceptions เป็นข้อยกเว้นอย่างหนึ่งที่ใช้ในการนิยามสถานะที่ผิดปกติหรือไม่ถูกต้องของโปรแกรม โดย Exception จะถูกส่ง (Thrown) ออกมาจากโปรแกรมและสามารถตรวจจับได้และนำ Exception นั้นไปตรวจสอบได้ด้วย

Exception จะต่างจากข้อผิดพลาด (Error) ตรงที่ข้อผิดพลาดจะไม่สามารถตรวจสอบได้เพราะมักทำให้โปรแกรมหยุดทำงานไปเลย แต่ถ้าข้อผิดพลาดใดสามารถตรวจจับและนำไปตรวจสอบต่อได้ก็จะถือว่าเป็น Exception

ประโยชน์ของการตรวจจับข้อผิดพลาด คือ จะไม่ทำให้โปรแกรมหยุดชะงักและเราสามารถสั่งให้โปรแกรมทำงานอื่นๆ ต่อไปได้อีก

2.3.1.1 Exception Messages

โดยปกติถ้าไม่มีการตรวจจับข้อผิดพลาดของโปรแกรมโดยใช้ Exception แล้วหากมีสิ่งผิดปกติเกิดขึ้นกับโปรแกรม จะทำให้โปรแกรมหยุดทำงาน (Terminate) และแสดงข้อความที่อธิบายถึงสิ่งผิดปกติที่เกิดขึ้น ข้อความนี้เรียกว่า Exception Message ซึ่งลักษณะเช่นนี้เรามักพบเห็นอยู่เสมอเมื่อทำการรันโปรแกรม

จาก Exception Message จะทำให้เราทราบว่าทำไมจึงเกิดข้อผิดพลาดขึ้นและบอกด้วยว่าเกิดขึ้นที่บรรทัดใดของโปรแกรม (รวมบรรทัดว่างด้วย)

2.3.1.2 ตรวจสอบโดยใช้คำสั่ง try, catch และ finally

คำสั่ง try ใช้ในการระบุช่วงของคำสั่ง (Block of statements) ที่จะตรวจสอบ
คำสั่ง catch ใช้สำหรับกำหนดข้อผิดพลาดที่เป็นไปได้
คำสั่งหลังบรรทัด catch ใช้กำหนดว่า ถ้าข้อผิดพลาดเกิดขึ้นจริง จะให้ทำอะไร

คำสั่ง finally ใช้ในการปลดปล่อยทรัพยากรที่โปรแกรมยึดครองไว้ในขณะทำงาน

รูปแบบของคำสั่งเป็นดังนี้

```
try {
    // โค้ดที่อาจจะทำให้เกิด Exception
}
catch (Type1 id1) {
    // โค้ดที่ใช้จัดการ Exception ของ Type1
}
catch (Type2 id2) {
    // โค้ดที่ใช้จัดการ Exception ของ Type2
}
catch (Type3 id3) {
    // โค้ดที่ใช้จัดการ Exception ของ Type3
}
finally {
    // โค้ดที่ใช้ปลดปล่อยทรัพยากร
```

โดย Type1, Type2, Type3 หมายถึง ชื่อคลาส Exception ที่ใช้ในการ
ตรวจสอบ
id1, id2, id3 หมายถึง ชื่อออบเจกต์ของ Type1, Type2, Type3
ตามลำดับ ซึ่งจะกำหนดเป็นชื่ออะไรก็ได้

2.3.1.3 Exception Propagation

ตามปกติ ถ้าตรวจจับข้อผิดพลาดและตรวจพบได้ภายในโปรแกรมแล้ว
โปรแกรมก็จะทำงานได้ตามปกติ แต่ถ้าเป็นข้อผิดพลาดที่เกิดขึ้นและไม่ได้ตรวจจับไว้ในเมทอด
(Method) ที่กำลังใช้งานอยู่ โปรแกรมจะส่งข้อผิดพลาดไปยังเมทอดที่เรียกใช้งานมาตามลำดับ
จนกว่าจะตรวจจับข้อผิดพลาดได้ แต่ถ้าเรียกไปจนถึงเมทอดหลัก (Main Method) แล้วยังไม่
สามารถตรวจจับได้ โปรแกรมก็จะหยุดทำงาน แล้วส่ง Exception Message ออกมาให้ทางจอภาพ
ลักษณะการส่งต่อเพื่อการตรวจจับข้อผิดพลาดนี้ เรียกว่า Exception Propagation

ในการตรวจจับข้อผิดพลาดตามลำดับการเรียกใช้เมทอดแบบนี้จำเป็นต้องนำเมทอดที่ต้องการตรวจจับมาใส่ไว้ในคำสั่ง try ด้วย เช่น ถ้ามีเมทอดชื่อ “level3”, “level2” และ “level1” โดย level1 จะเรียกใช้ level2 และ level2 เรียกใช้ level3 ตามลำดับ ทั้งนี้ที่ level3 เกิดข้อผิดพลาดขึ้นแต่ level3 ไม่มีการตรวจจับข้อผิดพลาดไว้ ดังนั้นโปรแกรมจึงส่งกลับมาที่ level2 แต่ level2 ก็ไม่มีการตรวจจับข้อผิดพลาดไว้เช่นกันจึงส่งไปที่ level1 ที่ level1 มีการตรวจจับข้อผิดพลาดไว้ จึงสามารถตรวจจับข้อผิดพลาดได้ และสามารถสั่งให้โปรแกรมทำงานอื่นๆ ต่อได้อีก

2.3.2 Threads

ในการทำงานของโปรแกรมโดยทั่วไปจะเป็นการเรียงลำดับก่อนหลัง (Sequential) นั่นคือ คำสั่ง (Statement) หนึ่งเมื่อประมวลผลเสร็จ อีกคำสั่งหนึ่งจึงจะถูกนำไปประมวลผลเป็นลำดับต่อไป

Thread เป็นกลไกของโปรแกรมอย่างหนึ่งที่อนุญาตให้สามารถประมวลผลงานมากกว่า 1 งานในโปรแกรมเดียวกันและเวลาเดียวกัน โดยงานหลายๆ งานนี้จะเข้าไปประมวลผลร่วมกันในหน่วยประมวลผลกลางตัวเดียวกัน (Concurrent Processing)

2.3.2.1 การสร้าง Thread

ในจาวาสสามารถสร้าง Thread ได้ 2 วิธี ได้แก่

1. ถ่ายทอด (Extend) มาจากคลาส Thread โดยจะต้องสร้างเมทอด “run” ขึ้นมาเป็น Overridden method ด้วย
2. Implement มาจากอินเตอร์เฟส Runnable ซึ่งจะต้องสร้างเมทอด “run” ขึ้นมาใหม่เสมอ จากนั้นจึงส่งออบเจกต์ไปยังคอนสตรักเตอร์ของ Thread ออบเจกต์ที่สร้างขึ้นใหม่

2.3.2.2 ปัญหาของ Thread

หากในโปรแกรมหนึ่งมีการสร้าง Thread ขึ้นมามากกว่า 1 Thread และจะต้องนำผลลัพธ์ที่ได้จากแต่ละ Thread มาใช้งานร่วมกัน อาจเกิดข้อผิดพลาดหรือผลลัพธ์ที่ได้อาจผิดปกติไปก็ได้

ตัวอย่าง มีสามีภรรยาคนหนึ่งเปิดบัญชีร่วมกัน โดยสามารถถอนเงินได้ทั้งคู่ผ่านทาง ATM ทั้งคู่มีเงินในบัญชีขณะนั้น 53,100 บาท ในเวลาเดียวกัน หากสามีซึ่งอยู่ที่สาขาที่ 1 ถอนเงินโดย ATM จำนวน 30,000 บาทและภรรยาซึ่งอยู่ที่สาขาที่ 2 ถอนเงิน 30,000 บาท จะเกิด

อะไรขึ้นเมื่อทั้งคู่ถอนเงินพร้อมกัน ดังนั้นโปรแกรมจึงตรวจสอบว่ามียอดถอนได้ทั้งคู่ แต่สุดท้ายจะกลายเป็นว่ายอดติดลบ คือถอนเงินเกินบัญชีนั่นเอง

สาขาที่1 (สามี)	สาขาที่2 (ภรรยา)
งาน : ถอนเงิน 30,000 ยอดเงินคงเหลือ	งาน : ถอนเงิน 30,000
ยอดถอน <= ยอดคงเหลือ → 53100 ←	ยอดถอน <= ยอดคงเหลือ
Yes	Yes
ยอดคงเหลือ - = 30000 → 23100	
	-6900 ← ยอดคงเหลือ -=30000

2.3.2.3 Synchronization

ปัญหาที่เกิดจากการใช้ข้อมูลร่วมกันจะหมดไป ถ้าเราสามารถรับประกันได้ว่า จะไม่มี Thread ใดที่ถูกเรียกไปทำงานพร้อมๆ กัน จาว่าจะมีคำสั่ง Synchronization ให้เรานำไปใช้ร่วมกับเมธอดเพื่อกำหนดว่าให้มีเพียง 1 Thread เท่านั้นที่สามารถเรียกใช้เมธอดนี้ในเวลาเดียวกันได้ นั่นคือต้องรอให้ Thread หนึ่งทำงานกับเมธอดนี้เสร็จก่อน Thread อื่นจึงจะสามารถมาทำงานต่อในเมธอดนี้ได้

2.3.2.4 การควบคุมการทำงานของ Thread

เมธอดที่ใช้ควบคุมการทำงานของ Thread มี 3 เมธอดที่สำคัญ คือ

- suspend () ใช้หยุดการทำงานของ Thread ชั่วคราว
- resume () ใช้สั่งให้ Thread ทำงานต่อไปหลังจากที่ถูกให้หยุดชั่วคราวด้วยเมธอด suspend () ทั้งสองเมธอดนี้จะใช้งานร่วมกันเสมอ
- sleep (long milliseconds) ใช้สั่งให้ Thread หยุดการทำงานชั่วคราวเป็นระยะเวลาตามที่ระบุ (มีหน่วยเป็นเศษ 1 ส่วน 100 ของวินาที)

2.4 คลาส Properties

คลาส Properties เป็นคลาสที่ถ่ายทอดมาจากคลาส Hashtable ที่สามารถอ่านข้อมูลจากสตรีมหรือเขียนข้อมูลลงสตรีม ช่วยบรรจุกลุ่มของค่าต่างๆที่จะใช้ คลาส Properties อนุญาตให้

ออบเจกต์ของคลาส มีออบเจกต์ของ Properties โดยอัตโนมัติ ซึ่งจะส่งกลับคุณลักษณะต่างๆ ที่กำหนดไว้ด้วยตัวของมันเอง

คลาส Properties ให้การสนับสนุนสองคอนสตรัคเตอร์ คอนสตรัคเตอร์หนึ่งจะไม่มีพารามิเตอร์ และอีกคอนสตรัคเตอร์หนึ่งยอมรับคุณลักษณะต่างๆ ที่เกี่ยวข้องกับออบเจกต์ของคลาส Properties ที่ถูกสร้างขึ้น นอกจากนี้คลาส Properties ยังสนับสนุน Overriding Method อีกด้วย

คลาส Properties มีเมธอดต่างๆ ดังนี้

- `getProperties ()` อนุญาตให้นำคุณลักษณะออกมาแสดงโดยใช้ออบเจกต์ของคลาส String เป็นคีย์ โดยสามารถใช้ `getProperties ()` เมธอดในอีกรูปแบบหนึ่ง โดยอนุญาตให้สตริงของค่าที่ใช้เป็นค่าปกติ คือ เป็นค่าที่กำหนดให้โดยอัตโนมัติ ในกรณีที่คีย์ไม่ได้บรรจุในออบเจกต์ของคลาส Properties จะคืนค่าเป็นค่าว่าง (Null)
- `load ()` ใช้โหลดออบเจกต์ของคลาส Properties จากสตรีมข้อมูลเข้าและบันทึกออบเจกต์ให้สตรีมข้อมูลออก
- `save ()` อนุญาตให้บันทึกข้อความสำหรับอธิบายโปรแกรม โดยให้เริ่มต้นที่ตำแหน่งของออบเจกต์ในสตรีมข้อมูลออกที่บันทึกไว้
- `put ()` ทำการพิมพ์ตาราง Hash โดยอัตโนมัติ สำหรับออบเจกต์ต่างๆ ของคลาส Hashtable
- `get ()` ใช้แสดงค่าต่างๆ ของตาราง Hash อีกครั้ง เพื่อรับค่าต่างๆ ที่ตอบสนองคีย์ที่ต้องการ
- `remove ()` ใช้ย้ายคู่ของคีย์และค่าที่เกี่ยวข้องกับคีย์

2.5 การติดต่อกับส่วนผู้ใช้ (Graphics User Interface : GUI)

User Interface เป็นการสร้างจอภาพให้ผู้ใช้สามารถติดต่อกับโปรแกรมได้อย่างง่ายดายและสะดวกยิ่งขึ้น ทั้งนี้นิยมสร้างให้เป็นจอภาพแบบกราฟิก จึงเรียกว่า “Graphics User Interface” หรือ GUI เครื่องมือต่างๆ เหล่านี้เป็นคลาสที่อยู่ในแพ็คเกจชื่อ AWT (Abstract Windowing Toolkit) ทั้งนี้แต่ละคลาสจะมีเมธอดคอนสตรัคเตอร์ของตนเองที่จะถูกเรียกใช้เมื่อมีการสร้างออบเจกต์ใหม่ภายใต้คลาสต่างๆ

2.5.1 การเขียนโปรแกรมที่มีการตรวจสอบการทำงานของผู้ใช้ (Events-Driven Programming)

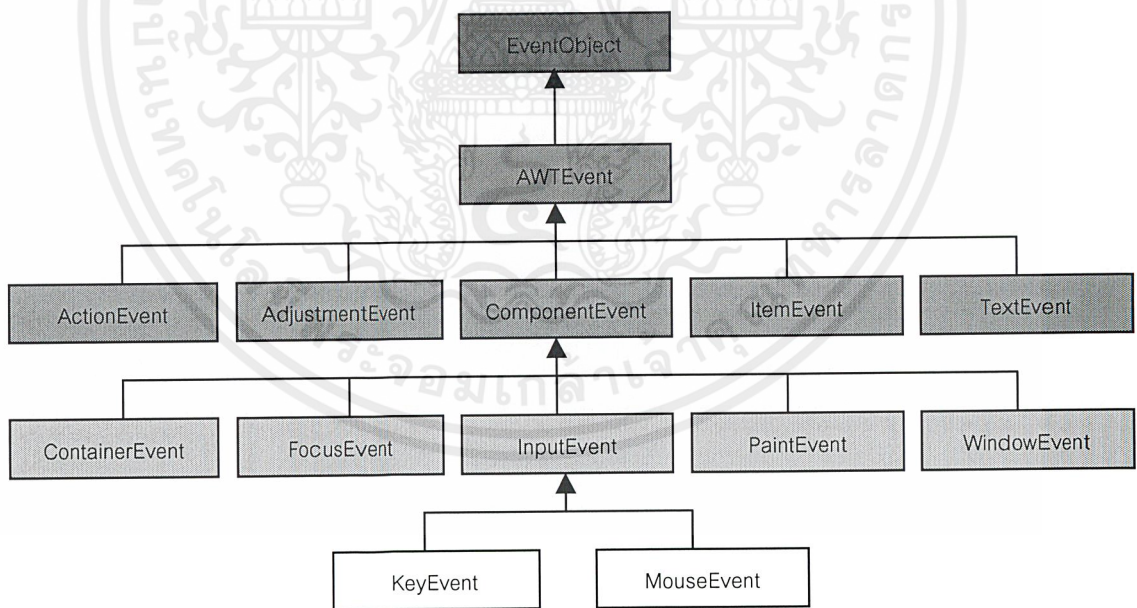
เป็นการเขียนโปรแกรมที่มีการตรวจสอบการทำงานของผู้ใช้ ไม่ว่าจะเป็นการกดแป้นพิมพ์ การเลือกตัวเลือกต่างๆ บนจอภาพของโปรแกรมที่เขียนขึ้น การกระทำต่างๆ ของผู้ใช้ เรียกว่า “Event” หรือเหตุการณ์ เมื่อมีเหตุการณ์หนึ่งเกิดขึ้น ต้องมีการตรวจสอบและกำหนดเงื่อนไขว่าต่อไปจะต้องทำอะไร เช่น ถ้ามีการเลือกตัวเลือกที่ 1 ให้แสดงสีจอภาพเป็นสีดำ แต่ถ้าเลือกตัวเลือกที่ 2 ให้แสดงสีจอภาพเป็นสีขาว เป็นต้น ลักษณะการกำหนดเงื่อนไขดังกล่าว เรียกว่า “Event Handling”

2.5.2 เหตุการณ์ (Event)

สามารถมีเหตุการณ์ใดๆ เกิดขึ้นในแต่ละคอมโพเนนต์ที่ได้ เช่น เมื่อกดปุ่ม (Button) โดยการคลิกด้วย เมาส์ (Mouse Click Event) เหตุการณ์ที่เกิดขึ้นกับปุ่ม คือ “Click Mouse” เป็นต้น แต่ละคอมโพเนนต์จะมีเหตุการณ์ที่เกิดขึ้นคล้ายๆ กัน โดยที่เหตุการณ์ทั้งหมดจะอยู่ในแพ็คเกจชื่อ “java.awt.Event”

ชนิดของเหตุการณ์

เหตุการณ์ที่มีในจาวามีทั้งหมด 14 เหตุการณ์ตามระดับ ดังรูป



รูปที่ 2-3 แสดงชนิดของเหตุการณ์

สำหรับเหตุการณ์ที่สำคัญ ได้แก่

- ActionEvents เกิดขึ้นจากการกระทำต่างๆของผู้ใช้ เช่น การคลิกปุ่ม ด้วยเมาส์ การดับเบิลคลิกเมาส์ที่รายการใดๆ ในลิสต์หรือการเลือกตัวเลือกในเมนู
- AdjustmentEvents เกิดจากการเลื่อนตำแหน่งของสไลด์บาร์
- ComponentEvents เกิดจากการกระทำใดๆ ที่เกิดขึ้นกับคอมโพเนนท์ (คอมโพเนนท์ คือ ออบเจ็กต์ใดๆ บนจอภาพ) เช่นการเคลื่อนย้าย การเปลี่ยนแปลงขนาด การซ่อน หรือการแสดงคอมโพเนนท์
- ItemEvents เกิดขึ้นจากการที่ผู้ใช้กระทำการเปลี่ยนแปลงคอมโพเนนท์ใดๆ
- MouseEvents เกิดขึ้นเมื่อมีการกดปุ่มหรือปล่อยปุ่มเมาส์ หรือมีการเคลื่อนย้ายตัวชี้เมาส์ไปในบริเวณใดๆ บนจอภาพ
- WindowEvents เกิดขึ้นเมื่อผู้ใช้มีการเปิด (Open) , ปิด (Close) , ขยาย (Maximize) หรือ ย่อ (Minimize) จอภาพ

2.5.3 การกำหนดเหตุการณ์ตอบสนอง (Event Handling)

การที่แต่ละคอมโพเนนท์ตอบสนองต่อเหตุการณ์ที่เกิดขึ้นอย่างไรขึ้นอยู่กับผู้เขียนโปรแกรมว่าต้องการให้ทำอะไร การกำหนดเหตุการณ์ตอบสนอง เรียกว่า “Driven” หรือ “Event Handling” ดังนั้นการเขียนโปรแกรมประเภทนี้ เรียกว่า “Event-Driven Programming”

ลักษณะของ Event-Driven จะเกิดการดำเนินงานที่ขั้นตอนใดก็ได้ขึ้นอยู่กับการกระทำของผู้ใช้โปรแกรม โดยปกติการเกิดเหตุการณ์ต่างๆ จะถูกตรวจจับโดยระบบปฏิบัติการ (Operating System) ที่ใช้งานอยู่และจะตรวจสอบอยู่ตลอดเวลา เหตุการณ์ที่เกิดขึ้นจะถูกนำไปเก็บไว้ในคิว (Queue) ไว้ก่อน จากนั้นโปรแกรมจาวาที่เขียนขึ้นมาจึงจะไปเรียกใช้เหตุการณ์ต่างๆ เหล่านี้จากในคิว (ในรูปแบบเข้าก่อนออกก่อน : First in First out)

ในการเขียนโปรแกรมเพื่อตรวจสอบเหตุการณ์ต่างๆ ภายในคิวนั้น ในจาวาจะกระทำโดย “Java Virtual Machine” (JVM) ซึ่งเป็นกลไกการทำงานของจาวาซึ่งจะตรวจสอบว่ามีเหตุการณ์ใดเกิดขึ้น จากนั้นจะสร้างออบเจ็กต์ขึ้นมาจากคลาสที่ถ่ายทอดมาจากคลาส EventObject ที่เกี่ยวข้องกับชนิดของเหตุการณ์ที่เกิดขึ้น ณ จุดนี้ หน้าทีของโปรแกรมจาวาคือ ต้องมีโค้ดสำหรับการ “จับตาดู” หรือที่เรียกว่า “Listens” เหตุการณ์เหล่านั้น ตัวอย่างเช่น ถ้าผู้ใช้กระทำการปิดจอภาพการทำงาน

โปรแกรมจะต้องไปทำงานตามที่ผู้ใช้สั่ง ดังนั้นจะต้องเขียนโปรแกรมในส่วนของคำสั่งให้โปรแกรมจับตามดูว่ามีเหตุการณ์ชนิดใดเกิดขึ้นบ้าง

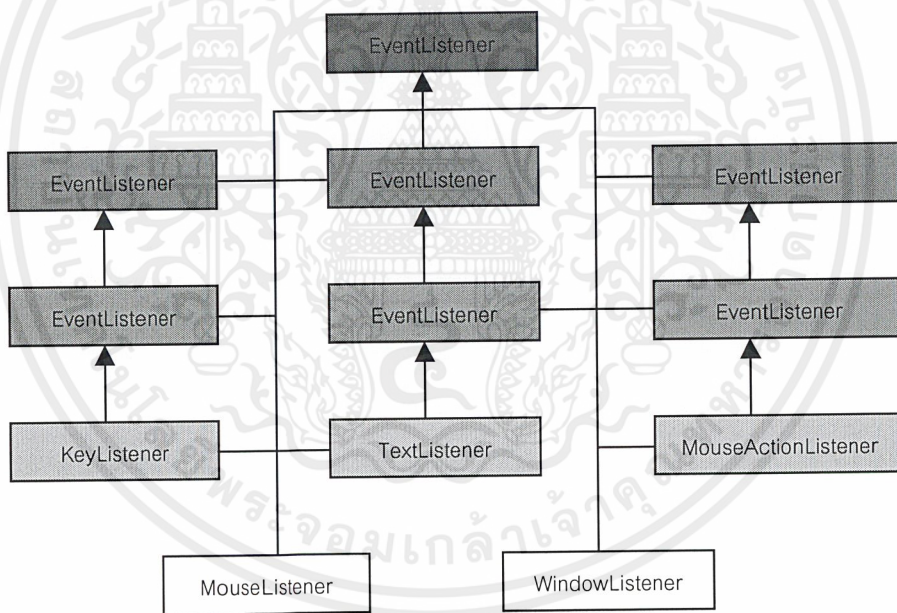
2.5.4 Listener

เป็นลักษณะของการเขียนโปรแกรมแบบ Event-Driven ในจาวาเวอร์ชัน 1.2 ขึ้นไป โดยการตรวจสอบเหตุการณ์ที่เกิดขึ้นด้วย "Listener"

การสร้างและการลงทะเบียน Listener Objects

เหตุการณ์ต่างๆ ที่เกิดขึ้น จะต้องถูกส่งมาให้กับออบเจกต์ที่เกี่ยวข้อง เรียกออบเจกต์นั้นว่า "Listener Objects" ซึ่ง Listener Objects เหล่านี้ถูกสร้างขึ้นมาจากคลาสที่เรียกว่า "Listener Interface"

Listener Interface มีทั้งหมด 11 ชนิด โดยแบ่งตามระดับ ดังรูป



รูปที่ 2-4 แสดงListener Interface

อินเตอร์เฟซเหล่านี้จะต้องทำการ import เข้ามาในโปรแกรมด้วย โดยใช้คำสั่ง `import java.awt.event.*` ในการที่จะทำให้ Listener Object ใช้งานได้นั้น จะต้องมี การ "Register" หรือลงทะเบียน Listener Object เสียก่อน การลงทะเบียนในแต่ละออบเจกต์จะไม่เหมือนกัน ขึ้นอยู่กับเหตุการณ์ด้วย แต่จะได้ง่ายๆ ว่า จะเป็นในรูปแบบ `addXXXListener` ซึ่ง XXX หมายถึง ชนิดของ Listener ที่จะลงทะเบียน

2.5.5 การตรวจสอบเหตุการณ์ (Handle Event)

เมื่อกำหนดการจับตามเหตุการณ์ต่างๆ แล้ว หากมีเหตุการณ์ใดเกิดขึ้นจะสามารถตรวจสอบได้ว่าเป็นเหตุการณ์อะไร และจะปฏิบัติอย่างไร (action) ทั้งนี้ในการเขียนโปรแกรมเพื่อทำงานดังกล่าวในจาวารุ่นเดิมและรุ่นใหม่จะเขียนอยู่ในรูปแบบที่ต่างกันดังนี้

- ถ้าเป็นจาวารุ่นเดิมจะใช้เมทอดชื่อ "action" ซึ่งมีรูปแบบการเขียนดังนี้

```
public boolean action (Event evt, Object arg) {
```

```
    if (arg == "Test Button") {
```

```
        if (Color == Color.black)
```

```
            color = Color.red;
```

```
    Else {
```

```
        color = Color.black;
```

```
        repaint ();
```

```
        return true;
```

```
    }
```

```
    return false;
```

```
}
```

- ถ้าเป็นจาวารุ่นใหม่ จะใช้เมทอดชื่อ "actionPerformed" ซึ่งมีรูปแบบการเขียน

ดังนี้

```
public void actionPerformed (ActionEvent event) {
```

```
    String arg = event.getActionCommand ();
```

```
    If (arg == "Test Button") {
```

```
        if (Color == Color.black)
```

```
            color = Color.red;
```

```
        Else
```

```
            color = Color.black;
```

```
        repaint ();
```

```
    }
```

```
}
```

2.5.6 เหตุการณ์ของเมาส์ (Mouse Events)

จาวาจัดเตรียมเมทอดที่จะรับข้อมูลจากเหตุการณ์ของเมาส์ที่ผู้ใช้กระทำกับเมาส์ โดยตั้งชื่อเมทอดตามลักษณะเหตุการณ์ เมทอดต่างๆ มีดังนี้

- public boolean mouseDown (Event event, int x, int y) เหตุการณ์เมื่อปุ่มเมาส์ถูกกดลง
- public boolean mouseUp (Event event, int x, int y) เหตุการณ์เมื่อปล่อยปุ่มเมาส์

เมทอด mouseDown () และ mouseUp () ใช้สามพารามิเตอร์ ได้แก่ เหตุการณ์ของมัน, พิกัด x และ y ตรงที่เกิดเหตุการณ์กดปุ่มและปล่อยปุ่มเมาส์

- public boolean mouseEnter (Event event, int x, int y)
- public boolean mouseExit (Event event, int x, int y)

เมทอด mouseEnter () และ mouseExit () จะถูกเรียกใช้งานเมื่อทำการใส่พอยน์เตอร์ของเมาส์เข้าไปใน applet หรือออกจากเมทอดของ applet โดยทั้งสองแบบนี้ จะมีรูปแบบการเขียนคำสั่งคล้ายกัน คือ ใช้พารามิเตอร์สามตัว ได้แก่ ออบเจกต์ของเหตุการณ์ , พิกัด x และ พิกัด y ของจุดที่เมาส์ถูกใส่หรือออกจาก applet

- public boolean mouseMove (Event event, int x, int y) เมาส์เคลื่อนที่ แต่ปุ่มของเมาส์ไม่ได้ถูกกดลง
- public boolean mouseDrag (Event event, int x, int y) เมาส์เคลื่อนที่พร้อมปุ่มของเมาส์ถูกกดลง

ทุกๆ ครั้งที่เมาส์เคลื่อนที่ไปตรงพิกเซล (Pixel) ใดพิกเซลหนึ่งบนจอภาพจะทำให้เกิดเหตุการณ์การเคลื่อนที่ ซึ่งจะทำให้เกิด 2 เหตุการณ์ คือ mouseDrag () และ mouseMove () โดยจะต้องรวมเมทอด mouseDrag () และ เมทอด mouseMove () ไว้ใน applet เพื่อที่จะสามารถ intercept และเชื่อมโยงกับเหตุการณ์ต่างๆ ของการเคลื่อนที่ของเมาส์ได้

คลาส InputEvent สามารถใช้แยกการกดปุ่มของเมาส์ว่ามาจากปุ่มใดได้

- isMetaDown () เมทอดนี้จะคืนค่าเป็น True เมื่อผู้กดปุ่มทางขวามือ
- isAltDown () เมทอดนี้จะคืนค่าเป็น True เมื่อผู้กดปุ่มเมาส์ตรงกลาง ในกรณี

ที่ไม่มีปุ่มตรงกลางผู้ใช้สามารถจำลองเหตุการณ์ดังกล่าว โดยการกดปุ่ม Alt พร้อมกับ การกดปุ่มเมาส์

Mouse Event Adapter Classes

อินเทอร์เฟซ Event-listener หลายอินเทอร์เฟซได้เตรียมหลายเมทอด เช่น อินเทอร์เฟซ MouseListener และ อินเทอร์เฟซ MouseMotionListener ซึ่งบางครั้งเราอาจต้องการใช้เพียงบางเมทอดในอินเทอร์เฟซเท่านั้น เราจึงอยากทำการให้คำนิยามเฉพาะเมทอดที่ต้องการ ด้วยเหตุนี้แพ็คเกจ java.awt.event จึงเตรียมคลาส Event-listener Adapter เอาไว้ ทุกๆ Event-listener interface ที่มีมากกว่า เมทอดจะมีคลาส Adapter ควบคุมด้วย คลาส Adapter จะต้องเตรียม Default Implementation ของทุกเมทอดในอินเทอร์เฟซ

Event adapter class	Implements
ComponentAdapter	ComponentListener
ContainerAdapter	ContainerListener
FocusAdapter	FocusListener
KeyAdapter	KeyListener
MouseAdapter	MouseListener
MouseMotionAdapter	MouseMotionListener
WindowAdapter	WindowListener

ตารางที่ 2-1 แสดง Mouse Event Adapter

ผู้เขียนโปรแกรมสามารถถ่ายทอดจากคลาส Adapter เพื่อถ่ายทอด Default Implementation ของทุกๆ เมทอดและกำหนดรายละเอียดเฉพาะเมทอดที่เราต้องการใช้ได้

2.6 กฎกติกาสากลของการเล่นเกมต่อคำศัพท์ภาษาอังกฤษ

อ้างอิงจาก J.W. Spear & Sons LTD. , “Rules For Playing Scrabble” , Enfield , Middlesex , England

เกมต่อคำศัพท์ภาษาอังกฤษ หรือ Scrabble นั้นเป็นเกมที่สามารถเล่นกันได้ ตั้งแต่ 2 ถึง 4 คน แต่ถ้าต้องการแข่งขันกันแบบสากลจะต้องเป็นการเล่นกัน 2 คน ทักษะของการเล่นคือ การประกอบตัวอักษรที่มีอยู่ให้เป็นคำในภาษาอังกฤษและต่อลงในช่องตารางบนกระดานให้เกิดผลดีที่สุด ซึ่งคำศัพท์นั้นต้องปรากฏอยู่ในพจนานุกรม "The Standard Dictionary For Crossword Game Players"

(3rd Tournament edition) เมื่อสิ้นสุดการแข่งขัน ผู้ที่ได้คะแนนมากที่สุดจะเป็นผู้ชนะ คะแนนจะเกิดจากตัวอักษรในคำและช่องตาราง ซึ่งมีค่าแตกต่างกันไป

ก่อนที่เราจะเริ่มเล่นเกมต่อคำศัพท์ภาษาอังกฤษนั้นเราจะต้องรู้ส่วนประกอบต่างๆ , อุปกรณ์และความหมาย และ วิธีการเล่นเสียก่อน ดังมีรายละเอียดต่อไปนี้

2.6.1 อุปกรณ์และความหมาย

- กระดาน (Board) กระดานจะมีขนาด 15x15 ช่อง = 225 ช่อง แบ่งออกเป็นช่องคะแนนธรรมดา 164 ช่อง (สีเขียว) และช่องคะแนนพิเศษ 61 ช่อง (สีแดง , สีชมพู , สีน้ำเงิน และสีฟ้า)
 - สีแดง (Triple word score) หมายถึง ตัวอักษรตัวใดที่ผู้เล่นลงทับช่องนี้จะส่งผลให้คำที่เกิดขึ้นจากตัวอักษรนี้ได้คะแนนเป็น 3 เท่า
 - สีชมพู (Double word score) หมายถึง ตัวอักษรใดที่ผู้เล่นลงทับช่องนี้จะส่งผลให้คำที่เกิดขึ้นจากตัวอักษรนี้ได้คะแนนเป็น 2 เท่า
 - สีน้ำเงิน (Triple letter score) หมายถึง ตัวอักษรใดที่ผู้เล่นลงทับช่องนี้จะส่งผลให้เฉพาะตัวอักษรนั้นได้คะแนนเป็น 3 เท่าของคะแนนเดิม
 - สีฟ้า (Double letter score) หมายถึง ตัวอักษรใดที่ผู้เล่นลงทับช่องนี้ จะส่งผลให้เฉพาะตัวอักษรนั้นได้คะแนนเป็น 2 เท่าของคะแนนเดิม

2. ตัวอักษร (Tiles) มีทั้งสิ้น 100 ตัว นับได้ตั้งแต่ A ถึง Z รวมทั้งตัว "Blank" โดยตัวอักษรแต่ละตัวจะมีจำนวนและคะแนนปรากฏอยู่ตามความยากง่ายของการเล่น ดังนี้

ตัวอักษร A	มีทั้งหมด 9 ตัว แต่ละตัวมีคะแนน 1 คะแนน
ตัวอักษร B	มีทั้งหมด 2 ตัว แต่ละตัวมีคะแนน 3 คะแนน
ตัวอักษร C	มีทั้งหมด 2 ตัว แต่ละตัวมีคะแนน 3 คะแนน
ตัวอักษร D	มีทั้งหมด 4 ตัว แต่ละตัวมีคะแนน 2 คะแนน
ตัวอักษร E	มีทั้งหมด 12 ตัว แต่ละตัวมีคะแนน 1 คะแนน
ตัวอักษร F	มีทั้งหมด 2 ตัว แต่ละตัวมีคะแนน 4 คะแนน
ตัวอักษร G	มีทั้งหมด 3 ตัว แต่ละตัวมีคะแนน 3 คะแนน
ตัวอักษร H	มีทั้งหมด 2 ตัว แต่ละตัวมีคะแนน 4 คะแนน
ตัวอักษร I	มีทั้งหมด 9 ตัว แต่ละตัวมีคะแนน 1 คะแนน
ตัวอักษร J	มีทั้งหมด 1 ตัว แต่ละตัวมีคะแนน 8 คะแนน
ตัวอักษร K	มีทั้งหมด 1 ตัว แต่ละตัวมีคะแนน 5 คะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษร L	มีทั้งหมด 4 ตัว	แต่ละตัวมีคะแนน 1 คะแนน
ตัวอักษร M	มีทั้งหมด 2 ตัว	แต่ละตัวมีคะแนน 3 คะแนน
ตัวอักษร N	มีทั้งหมด 6 ตัว	แต่ละตัวมีคะแนน 1 คะแนน
ตัวอักษร O	มีทั้งหมด 8 ตัว	แต่ละตัวมีคะแนน 1 คะแนน
ตัวอักษร P	มีทั้งหมด 2 ตัว	แต่ละตัวมีคะแนน 3 คะแนน
ตัวอักษร Q	มีทั้งหมด 1 ตัว	แต่ละตัวมีคะแนน 10 คะแนน
ตัวอักษร R	มีทั้งหมด 6 ตัว	แต่ละตัวมีคะแนน 1 คะแนน
ตัวอักษร S	มีทั้งหมด 4 ตัว	แต่ละตัวมีคะแนน 1 คะแนน
ตัวอักษร T	มีทั้งหมด 6 ตัว	แต่ละตัวมีคะแนน 1 คะแนน
ตัวอักษร U	มีทั้งหมด 4 ตัว	แต่ละตัวมีคะแนน 1 คะแนน
ตัวอักษร V	มีทั้งหมด 2 ตัว	แต่ละตัวมีคะแนน 4 คะแนน
ตัวอักษร W	มีทั้งหมด 2 ตัว	แต่ละตัวมีคะแนน 4 คะแนน
ตัวอักษร X	มีทั้งหมด 1 ตัว	แต่ละตัวมีคะแนน 8 คะแนน
ตัวอักษร Y	มีทั้งหมด 2 ตัว	แต่ละตัวมีคะแนน 4 คะแนน
ตัวอักษร Z	มีทั้งหมด 1 ตัว	แต่ละตัวมีคะแนน 10 คะแนน
ตัวอักษร Blank	มีทั้งหมด 2 ตัว	แต่ละตัวมีคะแนน 0 คะแนน

2.6.2 การเริ่มเกม

1. เลือกว่าผู้เล่นคนใดจะได้เล่นก่อนโดยกระทำการดังนี้ ให้ผู้เล่นแต่ละคน สุ่มจับตัวอักษรขึ้นมาคนละ 1 ตัว โดยมีหลักว่าใครได้ตัวอักษรที่ใกล้ที่สุดคนนั้นเป็นผู้เริ่มเล่นก่อน โดยเปรียบเทียบดังนี้ Blank , A , B , C , ... ไปจนถึง Z
2. เมื่อรู้ว่าผู้ใดเป็นผู้เล่นก่อนแล้วก็ให้คืนตัวอักษรนั้นกลับเข้าไปในถุงหรือกล่อง จากนั้นให้ผู้เล่นสุ่มจับตัวอักษรขึ้นมาคนละ 7 ตัว โดยให้ผู้ที่ได้เริ่มเล่นก่อนเป็นคนจับก่อน
3. เมื่อผู้เล่นทุกคนจับตัวอักษรครบแล้ว ผู้เล่นที่ได้เริ่มเล่นก่อนจะต้องผสมอักษรตั้งแต่ 2 ตัวขึ้นไปให้เป็นคำในภาษาอังกฤษ (ที่ปรากฏในพจนานุกรมที่กล่าวไว้ข้างต้นแล้ว) แล้วนำคำที่ได้นั้นไปลงบนกระดานในแนวนอนหรือแนวตั้งก็ได้ โดยจะต้องมีตัวอักษรตัวใดตัวหนึ่งของคำที่บออยู่บนดาวกลางกระดานและคำที่ลงครั้งแรกนี้จะได้คะแนนเป็น 2 เท่า เพราะช่องดาวบนกระดานก็คือช่องสี่เหลี่ยมนั่นเอง

4. เมื่อผู้เล่นได้ลงตัวอักษรไปบนกระดานแล้ว ก็ให้จับตัวอักษรตัวใหม่ที่อยู่ในถุงหรือกล่องขึ้นมาแทนให้เท่ากับตัวอักษรที่ใช้ลงไปบนกระดาน (จำนวนตัวอักษรในมือผู้เล่นจะเท่ากับ 7 ตัวเสมอ ยกเว้นกรณีที่อยู่ในกล่องหรือในถุงนั้นมีตัวอักษรไม่ครบ หรือหมดแล้ว) จากนั้นจะเป็นตาเล่นของผู้เล่นคนต่อไปซึ่งจะต้องต่อตัวอักษรที่มีอยู่ให้เป็นคำโดยมีตัวอักษรอย่างน้อยหนึ่งตัวสัมผัสกับตัวอักษรที่มีอยู่ในกระดานแล้วจากนั้นก็คิดคะแนนตามคำใหม่ที่เกิดขึ้นมาทุกคำจากการเล่นในตานี้ โดยผู้เล่นจะต้องลงตัวอักษรในแนวใดแนวหนึ่งเป็นแนวเดียวติดต่อกันเท่านั้น คำ ๆ ใหม่อาจเกิดขึ้นในกรณีดังนี้

- ผสมตัวอักษรหนึ่งตัวหรือมากกว่า กับคำหรือตัวอักษรที่มีอยู่ในกระดานแล้ว เช่น การทำให้เป็นพหูพจน์ การทำให้เป็นขั้นกว่า หรือขั้นสูงสุด เป็นต้น
- ใส่คำในแนวตั้งกับคำที่มีอยู่ในกระดาน โดยมีตัวอักษรตัวใดตัวหนึ่งของคำใหม่สัมผัสหรือเพิ่มเข้าไปในคำที่มีอยู่บนกระดาน
- ใส่คำที่สมบูรณ์แล้วในแนวนอนขนาน โดยตัวอักษรในคำใหม่ก่อให้เกิดคำใหม่อื่น ๆ ขึ้นมาจากการสัมผัสกับคำที่อยู่บนกระดานอยู่แล้ว

5. การขอเปลี่ยนตัว ผู้เล่นสามารถขอเปลี่ยนตัวอักษรได้ โดยต้องเสียตาเล่น 1 ตา การเปลี่ยนตัวอักษรสามารถเปลี่ยนได้ตั้งแต่ 1-7 ตัวแต่จะไม่สามารถเปลี่ยนตัวได้โดยเด็ดขาดเมื่อตัวอักษรในถุงเหลือไม่ถึง 7 ตัว

2.6.3 การคิดและคำนวณคะแนน

1. คะแนนที่ทำได้จะเกิดจากคำที่เกิดขึ้นใหม่ทุกคำในการลงเล่นตานี้ ซึ่งอาจเป็นการเปลี่ยนรูปให้เป็นพหูพจน์โดยใส่ S หรือทำให้เป็นขั้นกว่า หรือ คำในรูปอดีตกาล ย่อมถือเป็นคำที่เกิดขึ้นใหม่ทั้งนั้น คะแนนแต่ละครั้งจะเท่ากับคะแนนของตัวอักษรที่เล่นในคำที่สร้างขึ้นใหม่ประกอบด้วยช่องพิเศษต่างๆ จากตัวอักษรที่วางทับช่องนั้นๆ

2. จากช่องคะแนนพิเศษ 4 แบบ คือ ช่องสีฟ้า (คูณ 2 เฉพาะค่าของตัวอักษรนั้น) , สีน้ำเงิน (คูณ 3 เฉพาะค่าของตัวอักษรนั้น) , สีชมพู (คูณ 2 ทั้งคำหลังจากรวมค่าแต่ละตัวอักษรของคำนั้นแล้ว) และสีแดง (คูณ 3 ทั้งคำหลังจากรวมค่าแต่ละตัวอักษรของคำนั้นแล้ว) หากเกิดกรณีที่ผู้เล่นลงคำซึ่งมีตัวอักษรที่เล่นใหม่ทับช่องพิเศษมากกว่า 1 ช่องแล้ว คะแนนที่ได้จะนับคะแนนพิเศษให้ตัวอักษรก่อนแล้วค่อยมารวมสำหรับคะแนนพิเศษของคำ ตัวอย่าง เช่น หากผู้เล่นฝ่ายหนึ่งลงคำว่า "CHAIR" โดยมีตัว C ซึ่งมีค่า 3 แต้ม ทับบนช่องสีน้ำเงิน และ ตัว R ซึ่งมีค่า 1 แต้มทับช่องสีชมพูแล้ว คะแนนที่คิดออกมาจะเป็นดังนี้ $C(3 \times 3) + H(4) + A(1) + I(1) + R(1) = 16 \times 2 = 32$ แต้ม

3. ช่องตัวอักษรพิเศษและช่องคำพิเศษนั้นสามารถใช้ได้ในการเล่นที่ลงไปในการเล่นครั้งแรกเท่านั้น ในการเล่นครั้งต่อมาให้นับเฉพาะค่าของตัวอักษรอย่างเดียวไม่สามารถนำช่องพิเศษนั้นมาใช้ได้อีก
4. เมื่อตัวอักษรนั้นตกบนช่องสี่ชมพู , แดง ค่าของค่านั้นจะถูกคูณ 2 หรือ คูณ 3 ถึงแม้ว่าจะตัวอักษรที่ทับช่องนั้นจะเป็นตัว Blank ซึ่งไม่มีคะแนนก็ตาม
5. เมื่อคำ 2 คำขึ้นไปถูกสร้างขึ้นในการเล่นครั้งเดียวกัน คำแต่ละคำที่เกิดขึ้นใหม่จะถูกคิดคะแนน โดยตัวอักษรที่ทับช่องพิเศษจะถูกคิดคะแนนสำหรับแต่ละคำ (ด้วยคำพิเศษของแต่ละคำ)
6. การบิงโก (Bingo) ผู้เล่นจะได้คะแนนบวกพิเศษอีก 50 คะแนน นอกเหนือจากคะแนนที่ได้ตามปกติ

2.6.4 การสิ้นสุดเกม และการได้ผู้ชนะ

1. เกมจะสิ้นสุดเมื่อผู้เล่นคนใดคนหนึ่งใช้ตัวอักษรที่ตนมีอยู่จนหมด (หลังจากที่ตัวอักษรในถุงหมด)
2. หากผู้เล่นฝ่ายตรงข้ามยังคงมีตัวอักษรเหลืออยู่ ให้หาคะแนนรวมของตัวอักษรนั้นแล้วคูณด้วย 2 และนำไปบวกให้กับผู้เล่นที่เป็นคนลงตัวอักษรหมดก่อน (ยกเว้น Blank ไม่คิดลบ)
3. ในกรณีที่ผู้เล่นทุกคนไม่สามารถเล่นตัวอักษรที่เหลืออยู่ในมือของเขาได้ และบอกผ่านครบ 3 ครั้งติดต่อกันก็ถือว่าเกมการแข่งขันสิ้นสุดลง การนับคะแนนจะทำโดยเอาคะแนนรวมของตัวอักษรที่เหลืออยู่ในเป็นลบออกจากคะแนนของตัวเองโดยไม่ต้องคูณ 2 (คะแนนของ Blank เท่ากับศูนย์)
4. ผู้ชนะคือผู้ที่มีคะแนนสูงสุดหลังจากสิ้นสุดการเล่นแล้ว

2.6.5 ตัวอย่างการคิดคะแนน

สมมุติว่า จิตราเป็นผู้เริ่มเล่นคนแรก ปิ่นทิพย์เป็นคนที่สอง และสิริพรเป็นคนที่สาม (จากรูปที่ 2-6)

- จิตราลงคำว่า "HIRE" จะได้คะแนนเท่ากับ 8 คะแนน จากแต่ละตัวอักษรมี 1 คะแนน รวมกันเป็น 4 คะแนน และมีตัวอักษรหนึ่งตัวที่อยู่บนช่องสี่ชมพู (ตัว I) ดังนั้นคะแนนที่ได้จะเท่ากับ 2 เท่าของคะแนนเดิม คือ $2 \times 4 = 8$

- ปิ่นทิพย์ลงคำว่า "SIRE" จะได้คะแนนเท่ากับ 7 คะแนน จากคะแนนของตัวอักษรทั้ง 4 ตัวรวมกัน $4 + 1 + 1 + 1$ จะเห็นว่าถึงแม้ว่าปิ่นทิพย์จะมีตัวอักษรที่อยู่บนช่องสี่ชมพู (ตัว I) แต่ตามกติกาจะไม่มี การคิดคะแนนพิเศษให้เนื่องจากคะแนนของช่องสี่ชมพูนี้ไม่ได้ใช้คิดให้จิตราไปแล้ว

- สิริพลลงคำว่า "GREAT" จะได้คะแนนเท่ากับ 12 คะแนน จากตัว R,E,และ A ที่แต่ละตัวมี 1 คะแนน รวมกันเป็น 3 คะแนน และตัว G กับตัว T ที่อยู่บนช่องสีน้ำเงิน ดังนั้นคะแนนของตัวอักษรจะเพิ่มเป็น 6 และ 3 ตามลำดับ (คิดเป็น 3 เท่าของคะแนนของตัวอักษรที่อยู่บนช่องสีน้ำเงิน คือ 3×2 และ 3×1) นั่นคือ คะแนนของสิริพลจะเท่ากับ $6+1+1+1+3=12$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRIPLE WORD SCORE			DOUBLE LETTER SCORE				TRIPLE WORD SCORE				DOUBLE LETTER SCORE			TRIPLE WORD SCORE
	DOUBLE WORD SCORE				TRIPPLE LETTER SCORE				TRIPPLE LETTER SCORE				DOUBLE WORD SCORE	
		DOUBLE WORD SCORE				DOUBLE LETTER SCORE		DOUBLE LETTER SCORE					DOUBLE WORD SCORE	
DOUBLE LETTER SCORE			DOUBLE WORD SCORE				DOUBLE LETTER SCORE					DOUBLE WORD SCORE		DOUBLE LETTER SCORE
				DOUBLE WORD SCORE							DOUBLE WORD SCORE			
	TRIPPLE LETTER SCORE				TRIPPLE LETTER SCORE				TRIPPLE LETTER SCORE					TRIPPLE LETTER SCORE
		DOUBLE LETTER SCORE				DOUBLE LETTER SCORE		DOUBLE LETTER SCORE					DOUBLE LETTER SCORE	
TRIPLE WORD SCORE			DOUBLE LETTER SCORE				DOUBLE WORD SCORE				DOUBLE LETTER SCORE			TRIPLE WORD SCORE
		DOUBLE LETTER SCORE				DOUBLE LETTER SCORE		DOUBLE LETTER SCORE				DOUBLE LETTER SCORE		
		TRIPPLE LETTER SCORE			TRIPPLE LETTER SCORE				TRIPPLE LETTER SCORE					TRIPPLE LETTER SCORE
				DOUBLE WORD SCORE							DOUBLE WORD SCORE			
DOUBLE LETTER SCORE			DOUBLE WORD SCORE				DOUBLE LETTER SCORE				DOUBLE WORD SCORE			DOUBLE LETTER SCORE
		DOUBLE WORD SCORE				DOUBLE LETTER SCORE		DOUBLE LETTER SCORE				DOUBLE WORD SCORE		
	DOUBLE WORD SCORE				TRIPPLE LETTER SCORE				TRIPPLE LETTER SCORE				DOUBLE WORD SCORE	
TRIPLE WORD SCORE			DOUBLE LETTER SCORE				TRIPLE WORD SCORE				DOUBLE LETTER SCORE			TRIPLE WORD SCORE

รูปที่ 2-5 แสดงกระดาน SCRABBLE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRIPLE WORD SCORE			DOUBLE LETTER SCORE				TRIPLE WORD SCORE				DOUBLE LETTER SCORE			TRIPLE WORD SCORE
	DOUBLE WORD SCORE				TRIPPLE LETTER SCORE			TRIPPLE LETTER SCORE					DOUBLE WORD SCORE	
		DOUBLE WORD SCORE				DOUBLE LETTER SCORE		DOUBLE LETTER SCORE					DOUBLE WORD SCORE	
DOUBLE LETTER SCORE			DOUBLE WORD SCORE				DOUBLE LETTER SCORE				DOUBLE WORD SCORE			DOUBLE LETTER SCORE
				DOUBLE WORD SCORE							DOUBLE WORD SCORE			
	TRIPPLE LETTER SCORE				TRIPPLE LETTER SCORE				TRIPPLE LETTER SCORE					TRIPPLE LETTER SCORE
		DOUBLE LETTER SCORE				DOUBLE LETTER SCORE	H ₄	DOUBLE LETTER SCORE				DOUBLE LETTER SCORE		
TRIPLE WORD SCORE			DOUBLE LETTER SCORE			S ₁	I ₁	R ₁	E ₁			DOUBLE LETTER SCORE		TRIPLE WORD SCORE
		DOUBLE LETTER SCORE				DOUBLE LETTER SCORE	R ₁	DOUBLE LETTER SCORE				DOUBLE LETTER SCORE		
	TRIPPLE LETTER SCORE					G ₂	R ₁	E ₁	A ₁	T ₁				TRIPPLE LETTER SCORE
				DOUBLE WORD SCORE							DOUBLE WORD SCORE			
DOUBLE LETTER SCORE			DOUBLE WORD SCORE				DOUBLE LETTER SCORE				DOUBLE WORD SCORE			DOUBLE LETTER SCORE
		DOUBLE WORD SCORE				DOUBLE LETTER SCORE		DOUBLE LETTER SCORE				DOUBLE WORD SCORE		
	DOUBLE WORD SCORE				TRIPPLE LETTER SCORE				TRIPPLE LETTER SCORE				DOUBLE WORD SCORE	
TRIPLE WORD SCORE			DOUBLE LETTER SCORE				TRIPLE WORD SCORE				DOUBLE LETTER SCORE			TRIPLE WORD SCORE

รูปที่ 2-6 แสดงตัวอย่างการคิดคะแนนของคำที่ลง

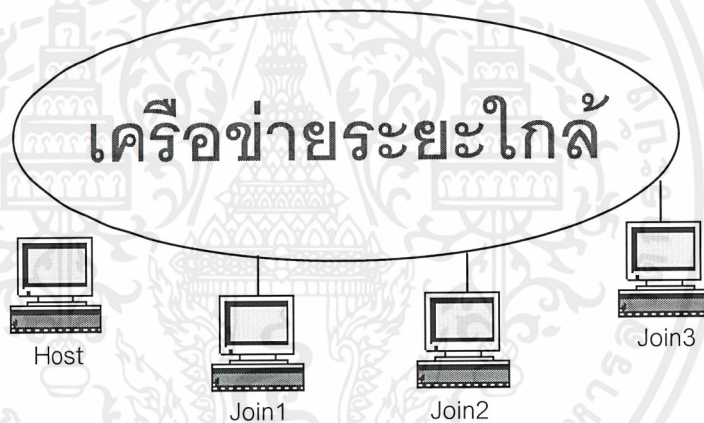
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบโปรแกรม

3.1 รูปแบบการเชื่อมต่อของคอมพิวเตอร์

เชื่อมต่อกันด้วยคอมพิวเตอร์จำนวน 2 ถึง 4 เครื่อง บนเครือข่ายระยะใกล้ผ่านโปรโตคอล TCP/IP โดยจะมีเครื่อง 1 เครื่องทำหน้าที่เปรียบเหมือน Server เรียกว่า "Host" และเครื่องจำนวน 1 ถึง 3 เครื่องทำหน้าที่เปรียบเหมือน Client เรียกว่า "Join" โดย Host จะรอรับการติดต่อจาก Join จากนั้นจะทำการตอบตกลงกลับไป ซึ่งจะทำให้ Join สามารถเล่นเกมได้ เครื่องคอมพิวเตอร์ทั้งหมดจะสามารถเล่นเกมต่อคำศัพท์ภาษาอังกฤษนี้ได้ก็ต่อเมื่อมีคอมพิวเตอร์ภาษาจาวาอยู่ในเครื่องเรียบร้อยแล้วเท่านั้น



รูปที่ 3-1 แสดงรูปแบบการเชื่อมต่อของคอมพิวเตอร์

3.2 อัลกอริทึมและแผนผังการทำงาน

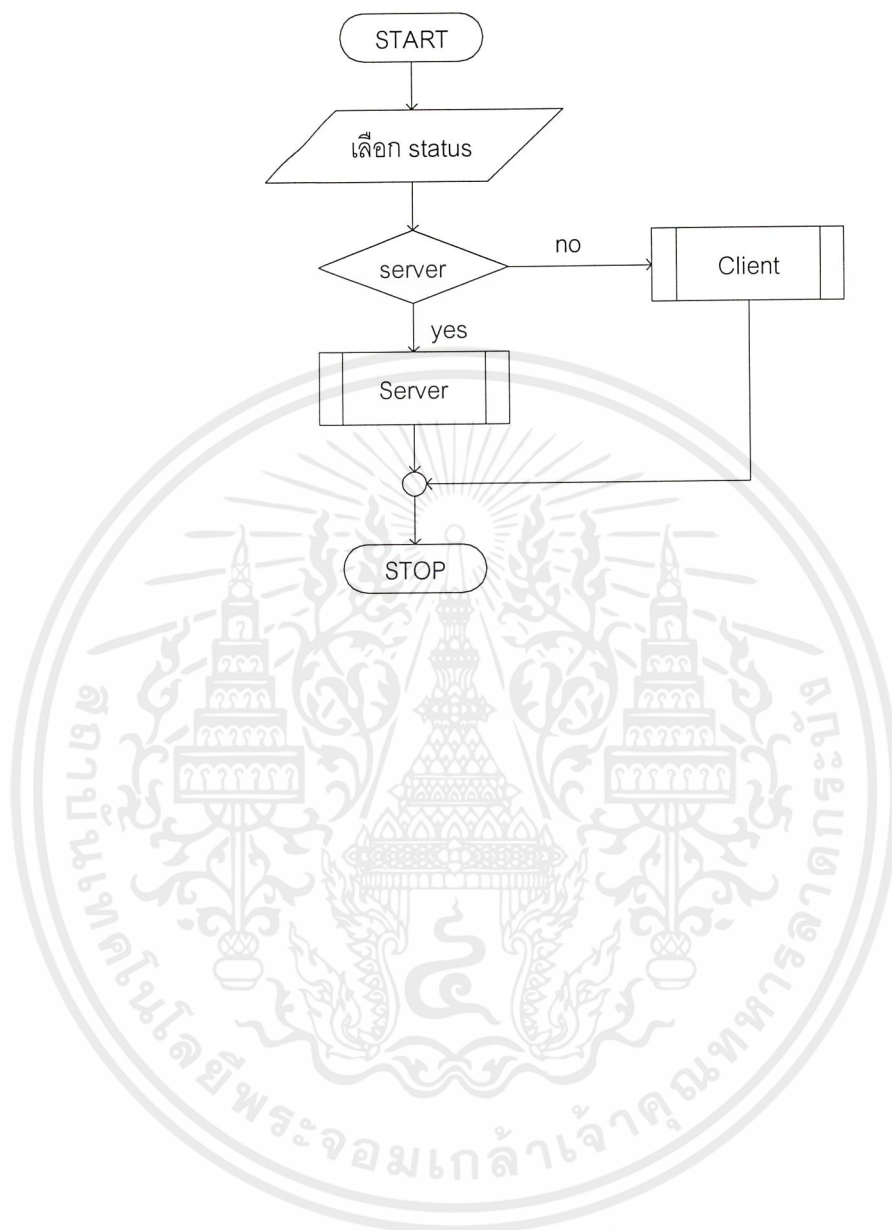
3.2.1 อัลกอริทึมในการหาผู้ชนะ

```

CountOfPass=0;
Finish=False;
While(true) {
    if (letterInBag == 0) and (remainLetterOfNowPlayer == 0) {
        Finish=True;
    }
    Else
        If (player1_Pass == true) and (player2_Pass == true) and
        (player3_Pass == true ){
            CountOfPass += 1;
        }
        Else {
            CountOfPass=0;
        }
    }
    If (CountOfPass == 3 or Finish==true) {
        Winnner = bestScore;
    }
    break;
}

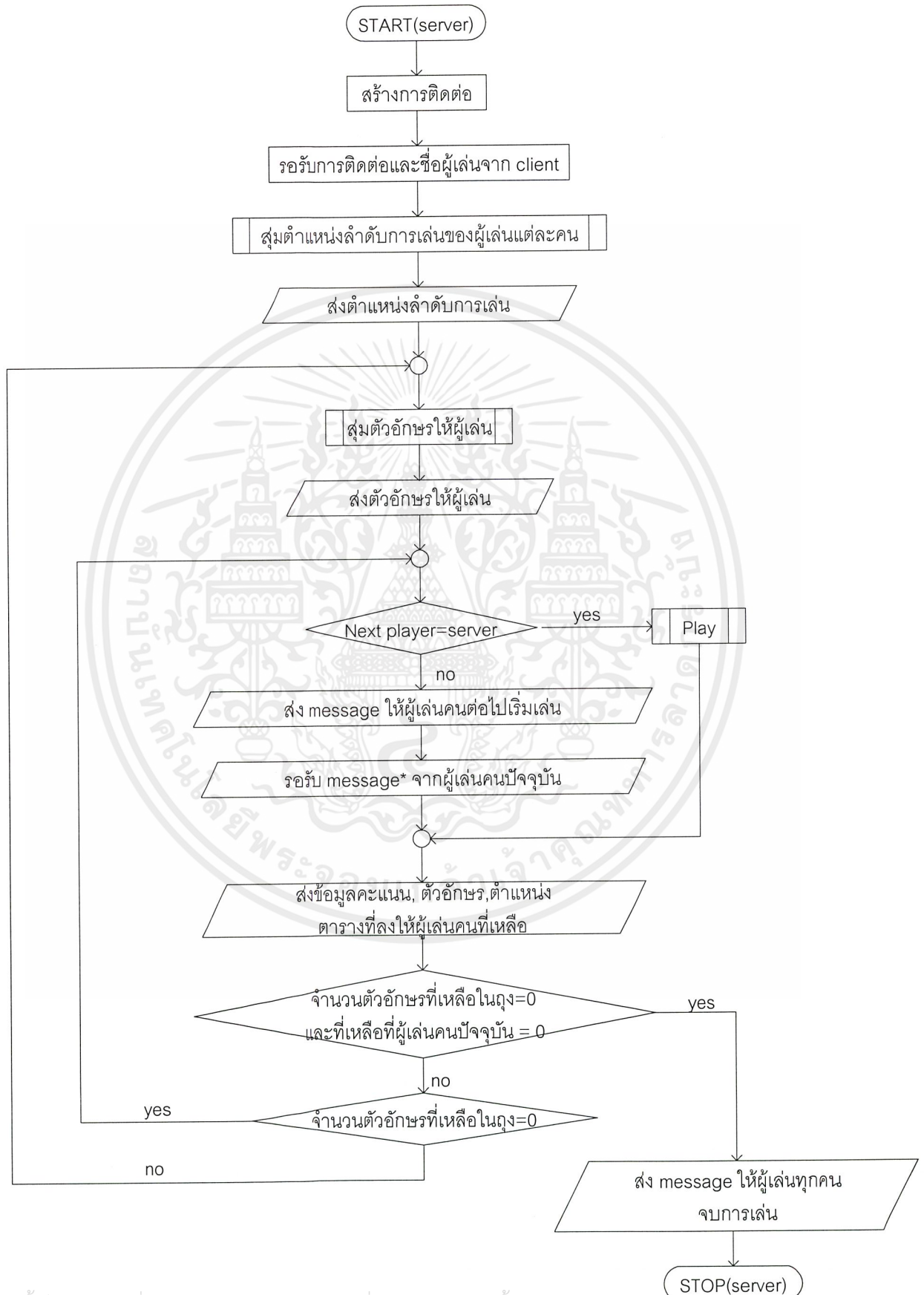
```

3.2.2 แผนผังแสดงการทำงานของระบบ



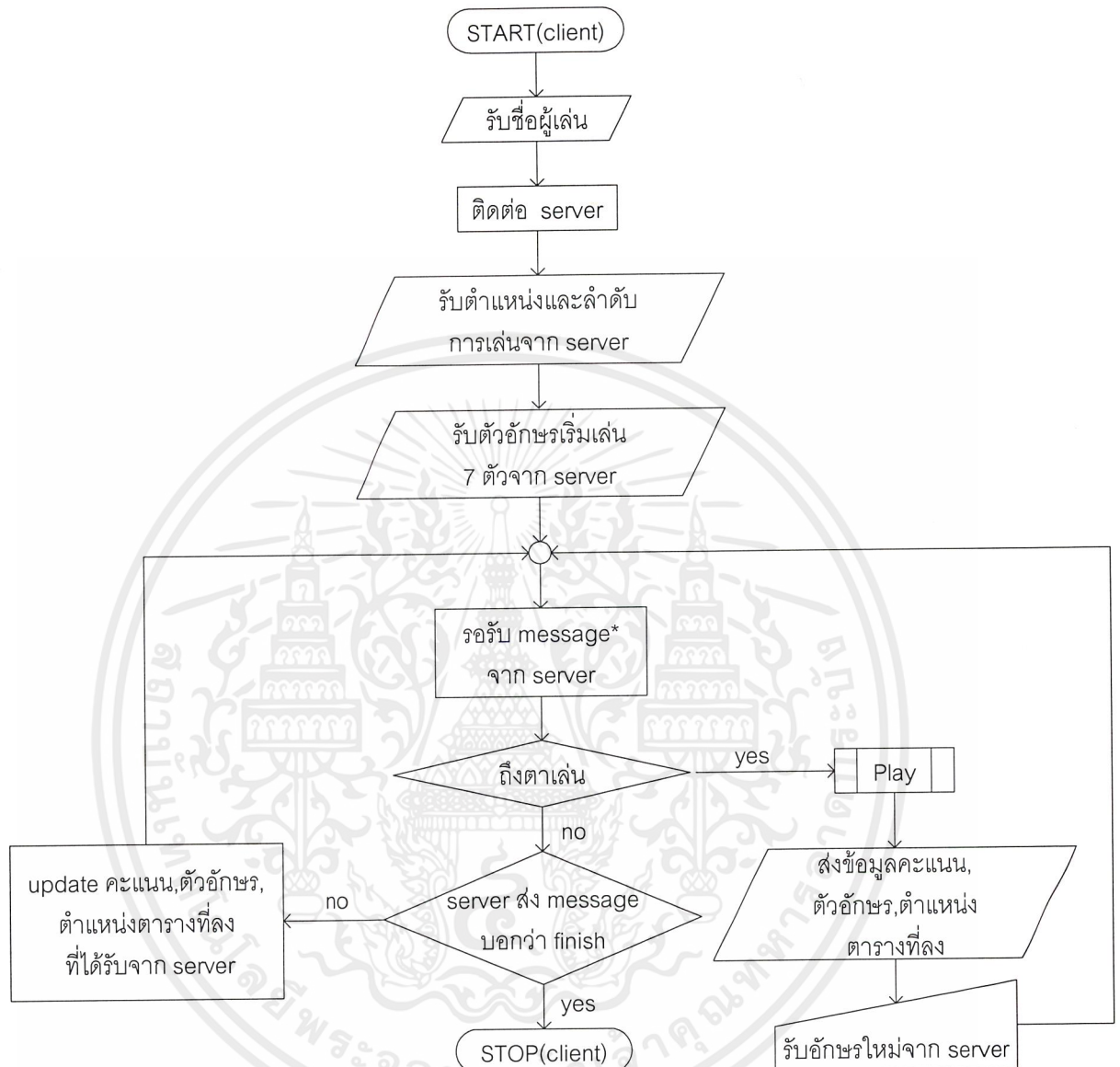
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.1 แผนผังแสดงการทำงานของ Server



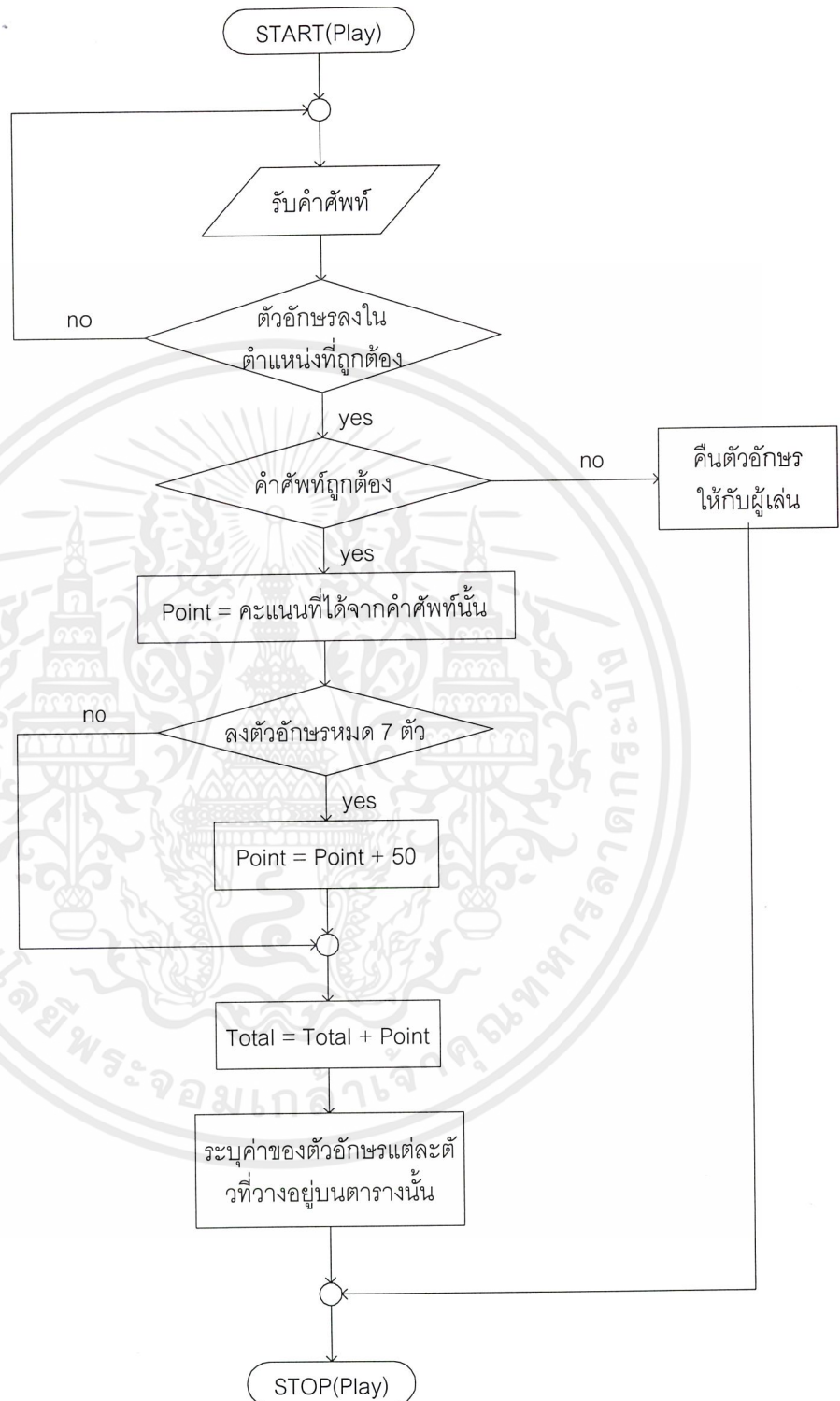
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า message* ได้แก่ คะแนน, ตัวอักษร, ตำแหน่ง ตารางที่ส่งตัวอักษร ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 แผนผังแสดงการทำงานของ Client



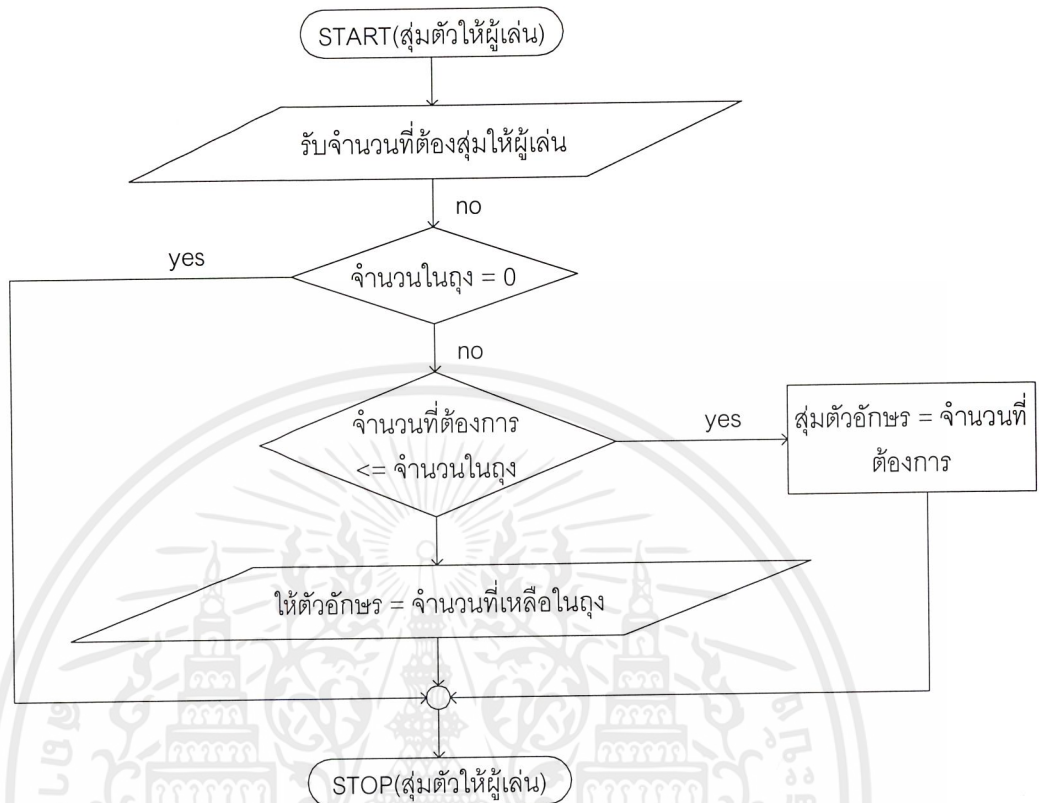
message* คือ ที่บอกว่าถึงตาเล่นหรือยัง

3.2.2.3 แผนผังแสดงการทำงานของ Play



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

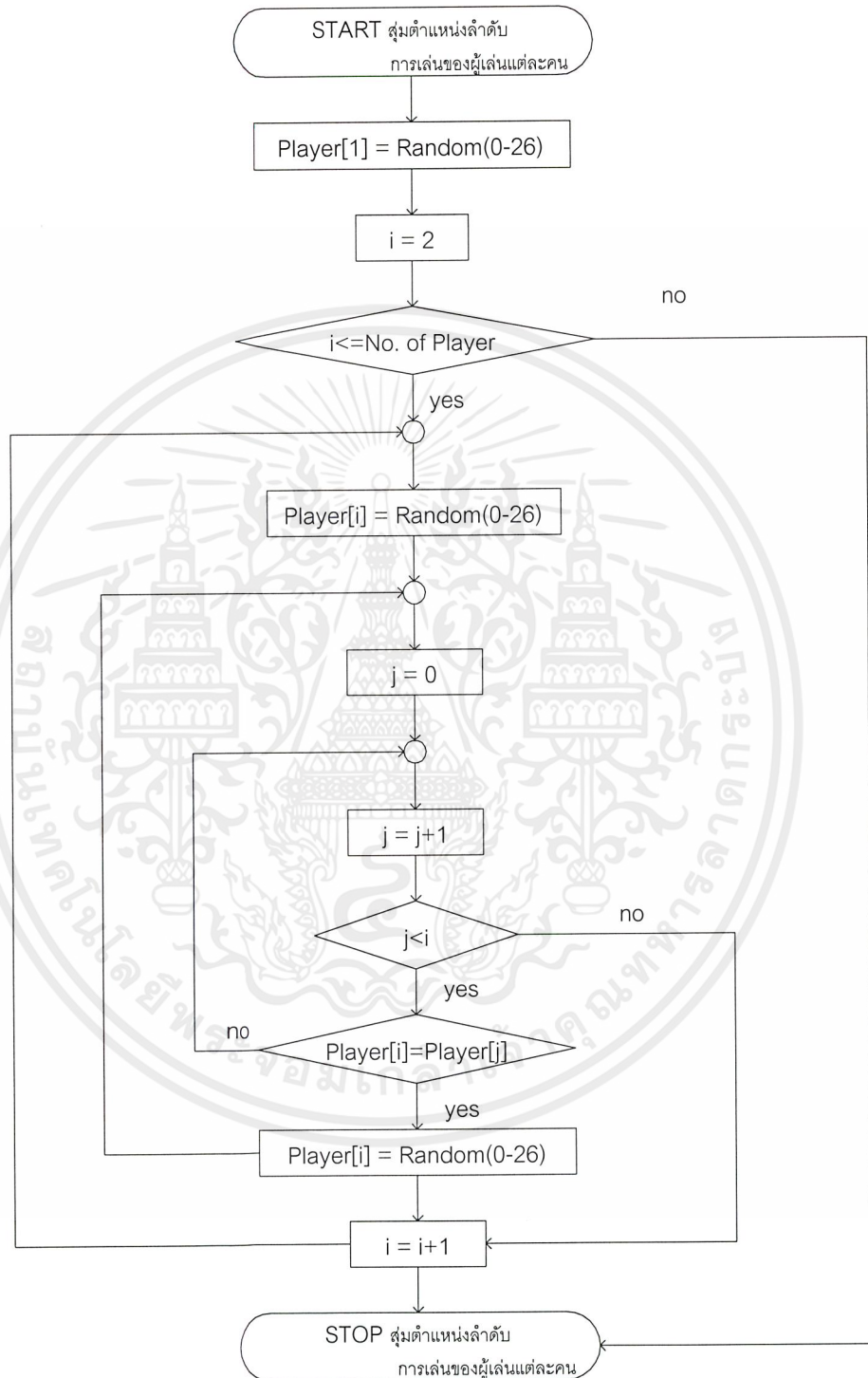
3.2.2.4 แผนผังแสดงการทำงานของการทำงานการสุ่มตัวให้ผู้เล่น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.5 แผนผังแสดงการทำงานของการทำงานการสุ่มตำแหน่งลำดับการเล่นของผู้เล่น

แต่ละคน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ประเมินและอภิปรายผล

4.1 ลักษณะของโปรแกรม

โปรแกรมเกมต่อคำศัพท์ภาษาอังกฤษบนเครือข่ายระยะใกล้ ถูกพัฒนาขึ้นโดยใช้ภาษาจาวา (JDK 1.2.2) ผ่านโปรโตคอล TCP/IP มีความสามารถดังต่อไปนี้

- 1) ตรวจสอบความถูกต้องของคำศัพท์ที่ผู้เล่นแต่ละคนลงบนกระดาน หากมีคำศัพท์ปรากฏอยู่ในพจนานุกรม โปรแกรมจะแสดงความหมายของคำศัพท์ออกทางหน้าจอ
- 2) เพิ่มคำศัพท์ที่ต้องการ โดยผู้เล่นที่เป็น Host เท่านั้น และผู้เล่นที่เป็น Join สามารถเห็นคำศัพท์ที่ Host เพิ่มเข้าไปได้ด้วย
- 3) ลบคำศัพท์ที่ต้องการจากพจนานุกรมฉบับที่ผู้เล่นทำการเพิ่มคำศัพท์เข้าไปเอง โดยผู้เล่นที่เป็น Host เท่านั้น
- 4) เลือกประเภทของพจนานุกรมที่ต้องการใช้ในการตรวจสอบคำศัพท์ได้ โดยแบ่งเป็นพจนานุกรมฉบับมาตรฐาน และฉบับที่ผู้เล่นทำการเพิ่มคำศัพท์เข้าไปเอง
- 5) ทำการคิดคะแนนที่เกิดจากการลงคำศัพท์ในแต่ละตาเล่นของผู้เล่น
- 6) สามารถผ่านตาเล่นตานั้นได้ หากต้องการ
- 7) สามารถเปลี่ยนตัวอักษรได้ตั้งแต่ 1 ถึง 7 ตัว
- 8) สามารถกำหนดตัว Blank ให้เป็นตัวอักษรใดก็ได้ตามที่ผู้เล่นต้องการ โดยจะมีคะแนนเป็นศูนย์
- 9) สามารถลากตัวอักษรไปวางที่ตำแหน่งใดๆ บนจอภาพได้ ในพื้นที่ที่กำหนด
- 10) จอภาพแสดงสถานะของผู้เล่นคนปัจจุบัน
- 11) สุ่มตัวอักษรให้กับผู้เล่นแต่ละคนโดยจะทำการเรียงลำดับจากน้อยไปมากให้กับผู้เล่นแต่ละคนเลย รวมถึงการสุ่มหาว่าผู้เล่นคนใดจะเป็นผู้เริ่มเล่น
- 12) รูปแบบอินเตอร์เฟส มีฟังก์ชันการใช้งานแบบเมนูและสร้างให้ใช้งานง่าย โดยใช้ Button , CheckBox , RadioButton , TextField
- 13) มีการเก็บสถิติคะแนนสูงสุด 5 คน โดยแบ่งเป็น 3 แพ้มข้อมูลตามจำนวนผู้เล่น (2 , 3 และ 4 คน)

4.2 ข้อกำหนดของการประเมิน

ติดตั้งโปรแกรมไว้ในเครื่องคอมพิวเตอร์โดยเครื่องนั้นจะต้องมีโปรแกรม Scrabble.exe อยู่ และ คอมพิวเตอร์ต้องเชื่อมต่อบนเครือข่ายระยะใกล้ผ่านโปรโตคอล TCP/IP การเล่นเกมในแต่ละครั้งจะมีผู้เล่นจำนวน 2 ถึง 4 คน โดยจะมีคอมพิวเตอร์เพียงเครื่องเดียวเท่านั้นที่มีสถานะเป็น Host ส่วนเครื่องอื่นจะมีสถานะเป็น Join

คอมพิวเตอร์แต่ละเครื่องจะทำงานภายใต้ระบบปฏิบัติการวินโดวส์ 98 หรือสูงกว่า , ซีพียูความเร็ว Pentium II 350 เมกะเฮิร์ต หรือสูงกว่า , หน่วยความจำหลักอย่างน้อย 128 เมกะไบต์ และมีพื้นที่ในหน่วยความจำสำรองอย่างน้อย 12 เมกะไบต์

4.3 การประเมิน

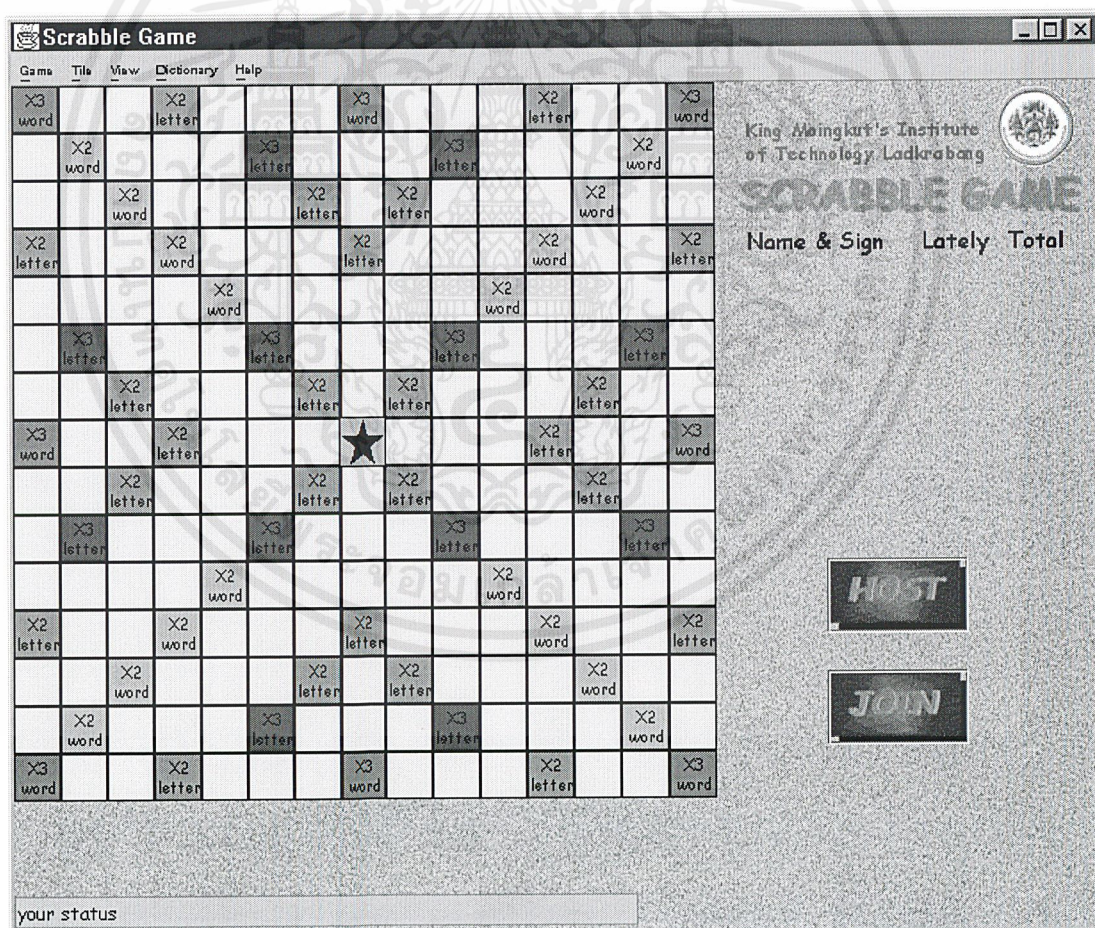
- 1) เริ่มการใช้งานโปรแกรม ให้ผู้เล่นเลือกสถานะของผู้เล่นว่าต้องการมีสถานะเป็น Host หรือ Join
- 2) ถ้าผู้เล่นเลือกเป็น Host (โดยการกดปุ่ม Host) ให้ผู้เล่นใส่ชื่อของตนและเลือกสัญลักษณ์แทนตนเอง
- 3) ถ้าผู้เล่นเลือกเป็น Join (โดยการกดปุ่ม Join) ให้ผู้เล่นใส่หมายเลข IP ของ Host , ใส่ชื่อของตน และ เลือกสัญลักษณ์แทนตนเอง
- 4) เมื่อผู้เล่นที่เป็น Host ได้ใส่ชื่อ และ เลือกสัญลักษณ์แทนตนเองและกดปุ่ม OK แล้ว จะรอรับการติดต่อจากผู้เล่นที่เป็น Join
- 5) เมื่อผู้เล่นฝ่าย Join ทำการใส่หมายเลข IP ของ Host , ชื่อของตน , เลือกสัญลักษณ์แทนตนเอง และ กดปุ่ม OK แล้ว จะรอให้ผู้เล่นคนอื่น ๆ ฝ่าย Join เชื่อมต่อกับ Host และ รอให้ Host เป็นผู้กดปุ่มเพื่อเริ่มเล่นเกม (Start)
- 6) เมื่อผู้เล่นฝ่าย Host กดปุ่มเริ่มเล่นเกม ให้ผู้เล่นฝ่าย Host เลือกว่าจะใช้พจนานุกรมคำศัพท์แบบมาตรฐาน หรือ พจนานุกรมที่มีคำศัพท์ที่ผู้เล่นได้เพิ่มเข้าไปอยู่ด้วย
- 7) หลังจากเลือกประเภทของพจนานุกรมแล้ว แสดงว่าเริ่มเล่นเกมแล้ว โดยลำดับการเล่นของผู้เล่นแต่ละคนเกิดจากผู้เล่นฝ่าย Host ทำการสุ่มให้
- 8) เมื่อผู้เล่นกดปุ่ม Change ให้ผู้เล่นเลือกตัวอักษรที่ผู้เล่นต้องการจะเปลี่ยน โดยสามารถเปลี่ยนได้ตั้งแต่ 1 ถึง 7 ตัวอักษร หลังจากนั้นให้ผู้เล่นกดปุ่ม OK โปรแกรมจะทำการสุ่มตัวอักษรตัวใหม่ให้กับผู้เล่น
- 9) หากผู้เล่นลากตัวอักษรไปวางที่กระดานแล้ว ผู้เล่นจะไม่สามารถเปลี่ยนตัวอักษรได้จนกว่าจะนำตัวอักษรออกมามองตารางหมัดก่อนจึงจะสามารถเปลี่ยนตัวอักษรได้

10) ถ้าผู้เล่นทำการกดยืนยันการลงโดยกด Done แล้วโปรแกรมจะทำการหาตำแหน่งว่าผู้เล่นลงถูกตำแหน่งหรือไม่ และ คำศัพท์ที่ลงไปนั้นมีความหมายในพจนานุกรมหรือไม่ หากคำศัพท์นั้นมีความหมาย จึงทำการแปลงตัวอักษรที่ลงไปให้เป็นสีน้ำตาล เพื่อแสดงว่าผู้เล่นได้ลงคำศัพท์นั้นเรียบร้อยแล้วและจะทำการสุ่มตัวอักษรใหม่ให้กับผู้เล่นเป็นจำนวนเท่ากับจำนวนตัวอักษรที่ลงไป หรือเท่ากับจำนวนตัวอักษรทั้งหมดที่เหลือในถุง (หากจำนวนตัวอักษรที่ผู้เล่นลงไปมีมากกว่าจำนวนตัวอักษรที่เหลือทั้งหมด)

4.4 ผลการทำงานของโปรแกรม

โปรแกรมเกมต่อคำศัพท์ภาษาอังกฤษบนเครือข่ายระยะใกล้ มีผลการทำงานของโปรแกรมโดยจะแสดงพอลังเชปดังนี้

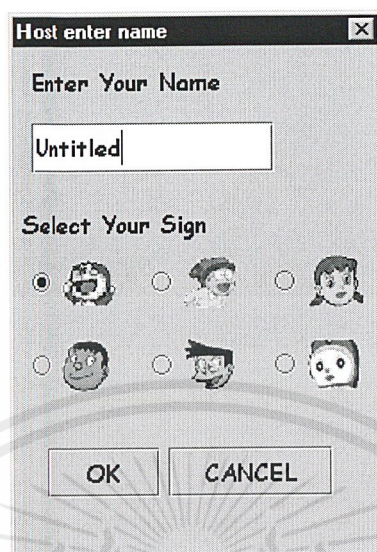
1) เมื่อทำการรันโปรแกรม โปรแกรมจะแสดงจอภาพดังรูปที่ 4-1 เพื่อให้ผู้เล่นเลือกสถานะของผู้เล่นว่าต้องการมีสถานะเป็น Host หรือ Join



รูปที่ 4-1 แสดงจอภาพเพื่อให้ผู้เล่นเลือกสถานะของตน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ถ้าผู้เล่นเลือกเป็น Host (โดยการกดปุ่ม Host) โปรแกรมจะแสดงจอภาพดังรูปที่ 4-2 เพื่อให้ผู้เล่น ใส่ชื่อของตน และ เลือกสัญลักษณ์แทนตนเอง



รูปที่ 4-2 แสดงจอภาพเพื่อให้ผู้เล่นที่เป็น Host ใส่ชื่อและเลือกสัญลักษณ์แทนตนเอง

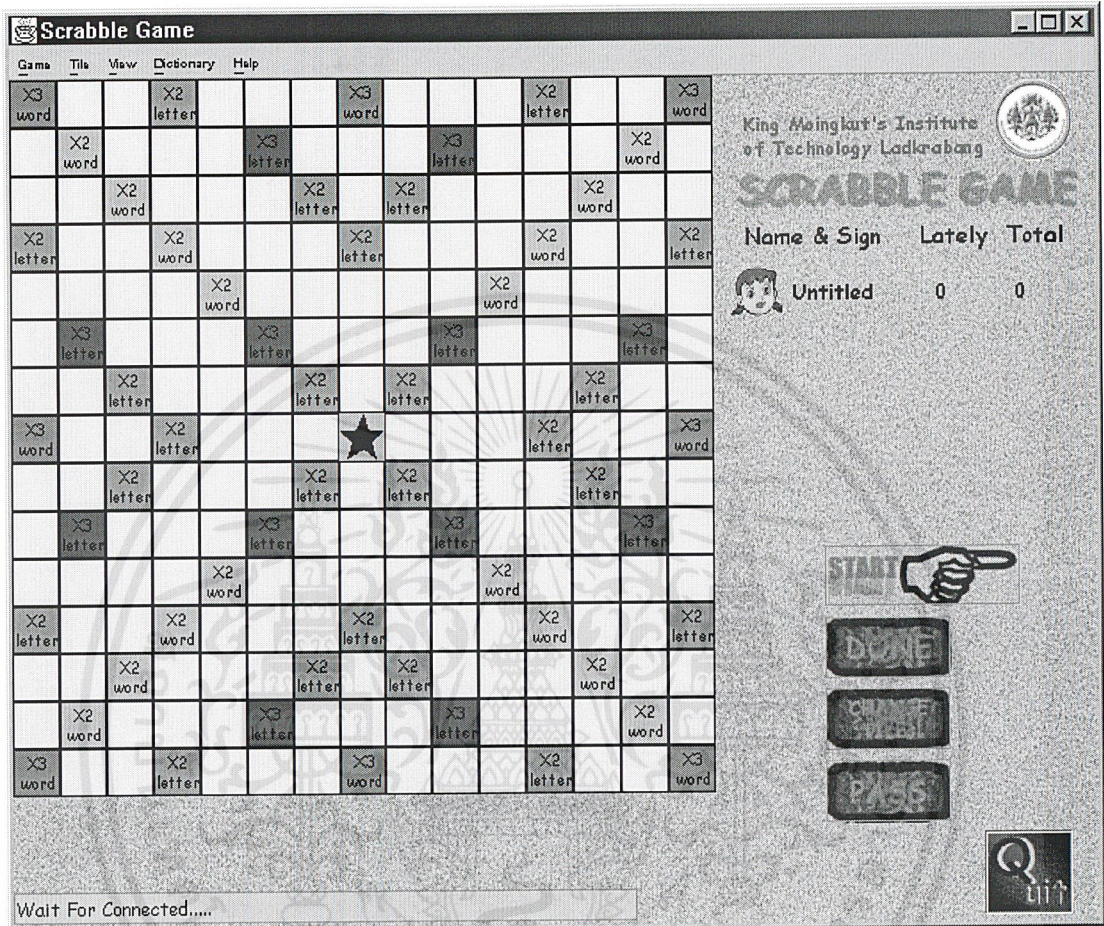
ถ้าผู้เล่นเลือกเป็น Join (โดยการกดปุ่ม Join) โปรแกรมจะแสดงจอภาพดังรูปที่ 4-3 เพื่อให้ผู้เล่นใส่หมายเลข IP ของ Host ,ใส่ชื่อของตน และ เลือกสัญลักษณ์แทนตนเอง



รูปที่ 4-3 แสดงจอภาพเพื่อให้ผู้เล่นที่เป็น Join ใส่หมายเลข IP ของ Host , ชื่อ และเลือกสัญลักษณ์ตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

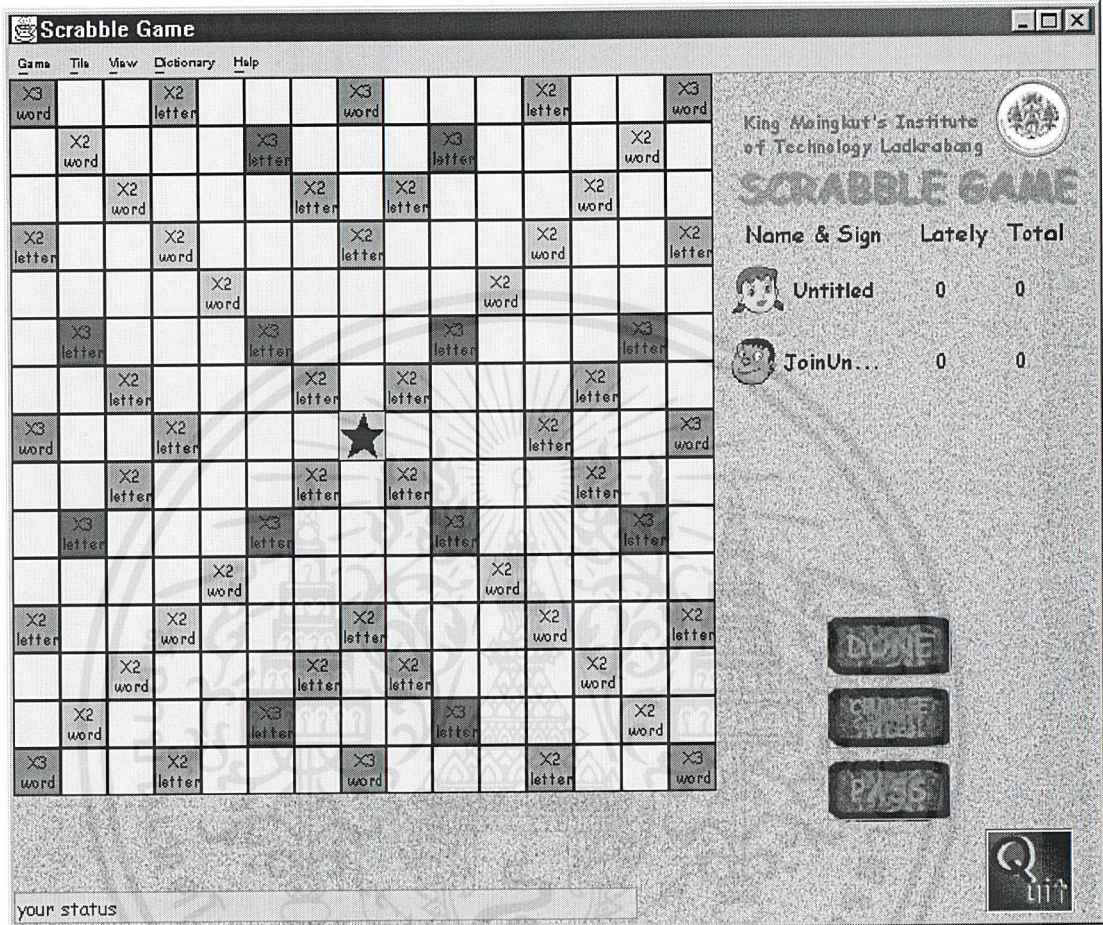
3) จากรูปที่ 4-2 เมื่อผู้เล่นที่เป็น Host ได้ใส่ชื่อ และ เลือกสัญลักษณ์แทนตนและ กดปุ่ม OK แล้ว โปรแกรมจะทำการแสดงจอภาพต่อไปดังรูปที่ 4-4 เพื่อรอรับการติดต่อจากผู้เล่น ที่เป็น Join



รูปที่ 4-4 แสดงจอภาพของผู้เล่นฝ่าย Host เพื่อรอรับการติดต่อจาก Join

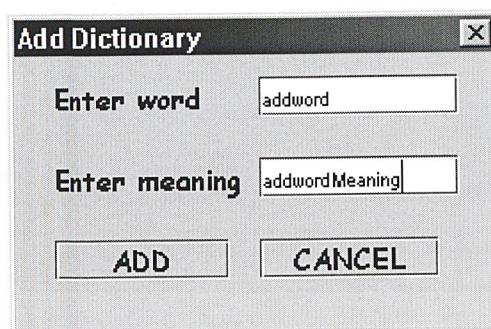
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4-3 เมื่อผู้เล่นฝ่าย Join ทำการใส่หมายเลข IP ของ Host , ชื่อของตน , เลือกสัญลักษณ์แทนตนเอง และ กดปุ่ม OK แล้ว โปรแกรมจะแสดงจอภาพดังรูปที่ 4-5 เพื่อรอให้ผู้เล่นคนอื่น ๆ ฝ่าย Join เชื่อมต่อกับ Host และ รอให้ Host เป็นผู้กดปุ่มเพื่อเริ่มเล่นเกม (Start)



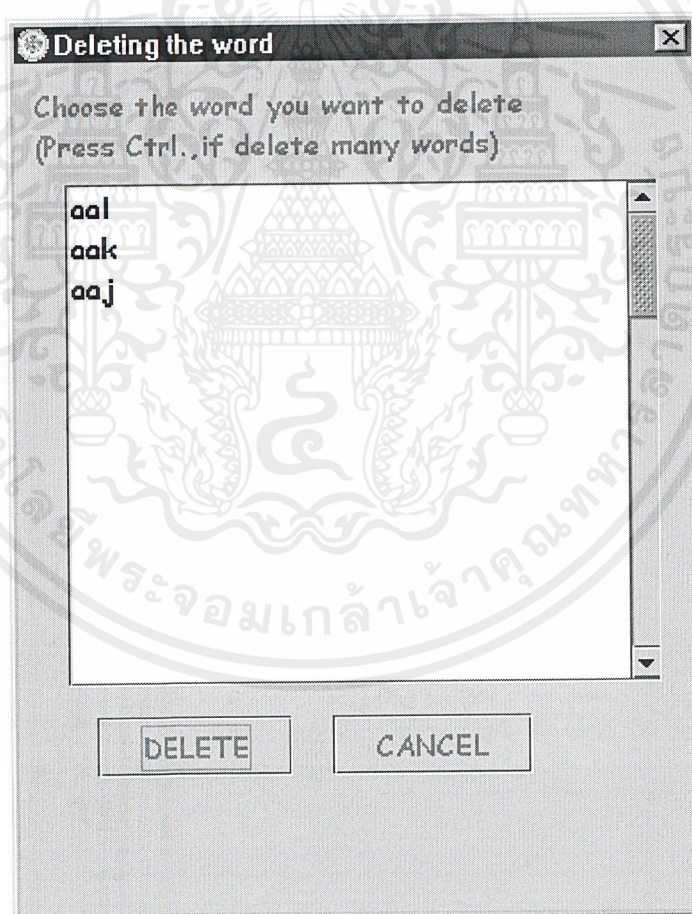
รูปที่ 4-5 แสดงจอภาพของผู้เล่นฝ่าย Join เพื่อรอให้ผู้เล่นอื่นฝ่าย Join เชื่อมต่อกับ Host และรอให้ Host เป็นผู้กดปุ่มเริ่มเล่นเกม (Start)

4) ผู้เล่นฝ่าย Host สามารถทำการเพิ่ม และ/หรือ ลบคำศัพท์ได้ ก่อนกดปุ่มเริ่มเล่นเกมเท่านั้น โดยหากผู้เล่นทำการเพิ่มคำศัพท์ โปรแกรมจะแสดงจอภาพดังรูปที่ 4-6



รูปที่ 4-6 แสดงจอภาพเมื่อผู้เล่นฝ่าย Host ทำการเพิ่มคำศัพท์

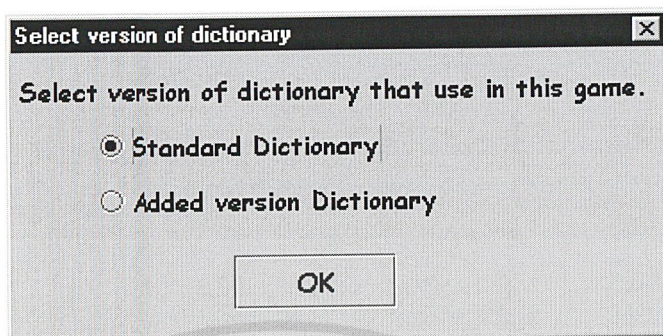
แต่หากผู้เล่นทำการลบคำศัพท์ โปรแกรมจะแสดงจอภาพดังรูปที่ 4-7



รูปที่ 4-7 แสดงจอภาพเมื่อผู้เล่นฝ่าย Host ทำการลบคำศัพท์

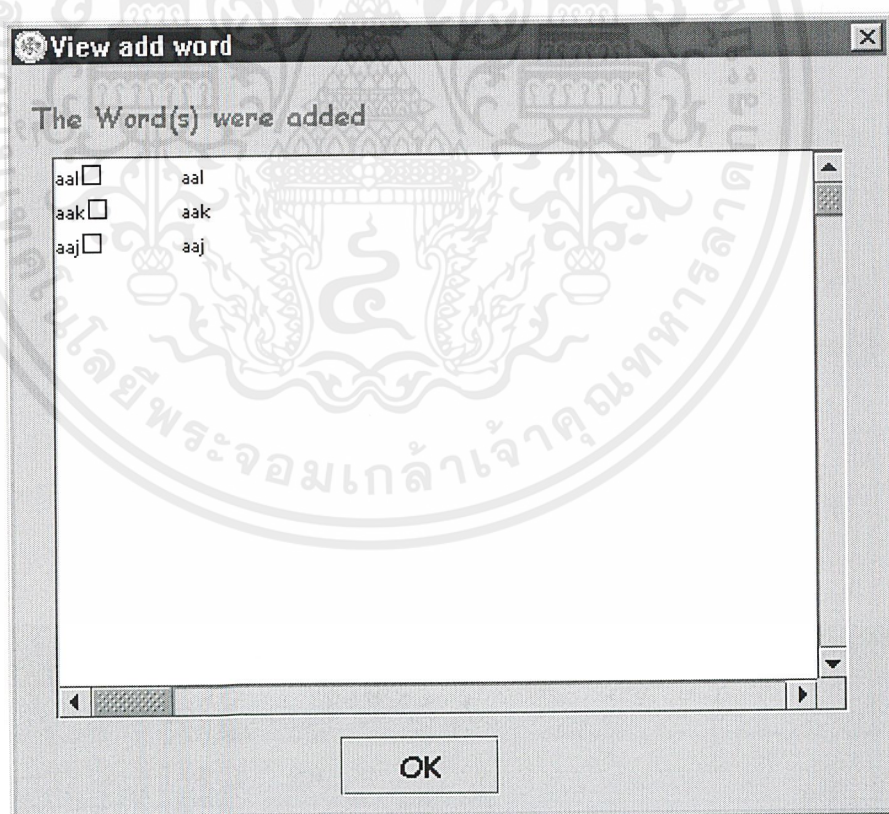
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) จากรูปที่ 4-4 เมื่อผู้เล่นฝ่าย Host กดปุ่มเริ่มเล่นเกม โปรแกรมจะแสดงจอภาพดังรูปที่ 4-8 เพื่อให้ผู้เล่นฝ่าย Host เลือกว่าจะใช้พจนานุกรมคำศัพท์แบบมาตรฐาน หรือ พจนานุกรมที่มีคำศัพท์ที่ผู้เล่นได้เพิ่มเข้าไปอยู่ด้วย



รูปที่ 4-8 แสดงจอภาพเพื่อให้ผู้เล่นฝ่าย Host ทำการเลือกประเภทของพจนานุกรม

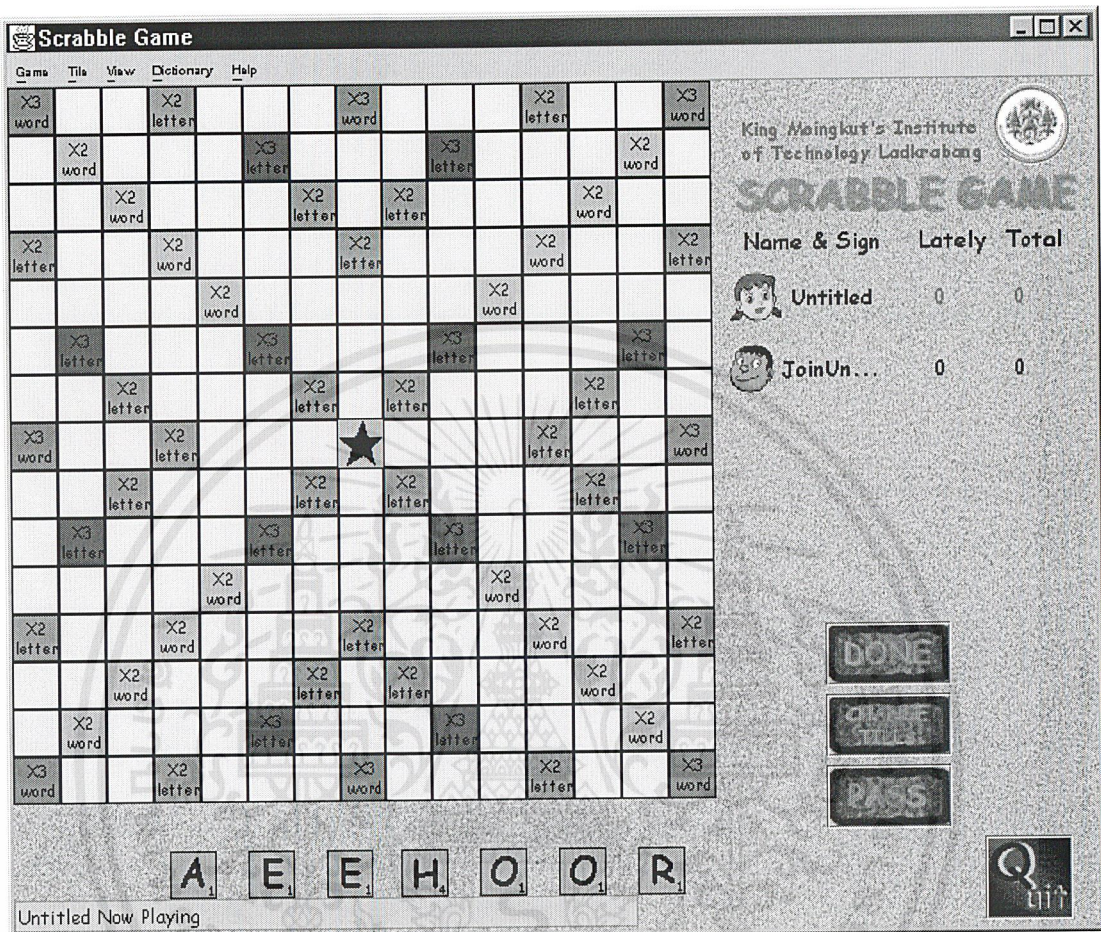
หากผู้เล่นฝ่าย Host เลือกรายการ Added version Dictionary โปรแกรมจะแสดงจอภาพดังรูปที่ 4-9 เพื่อบอกให้ผู้เล่นทุกคนทราบว่า มีคำศัพท์ใดบ้างที่อยู่ภายในพจนานุกรมฉบับที่ไม่ได้เป็นมาตรฐาน



รูปที่ 4-9 แสดงจอภาพเมื่อผู้เล่นฝ่าย Host เลือกใช้พจนานุกรมฉบับเพิ่มเติมคำศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

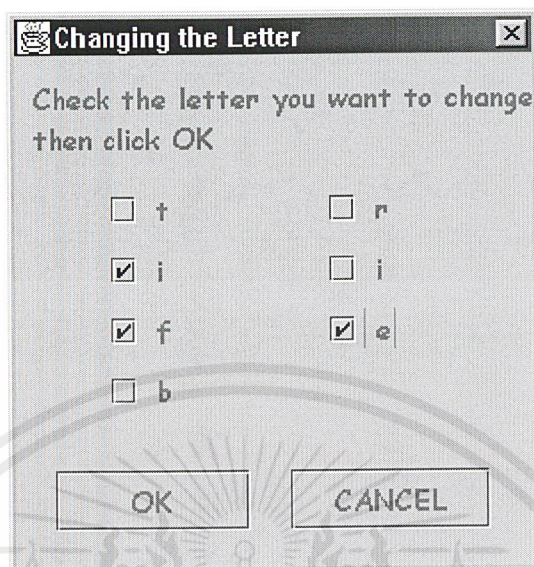
6) หลังจากเลือกประเภทของพจนานุกรมแล้ว โปรแกรมจะแสดงจอภาพดังรูปที่ 4-10 แสดงว่าเริ่มเล่นเกมแล้ว โดยลำดับการเล่นของผู้เล่นแต่ละคนเกิดจากผู้เล่นฝ่าย Host ทำการสุ่มให้ จากภาพผู้เล่นฝ่าย Host ได้เริ่มเล่นก่อน



รูปที่ 4-10 แสดงจอภาพของผู้เล่นในขณะที่เล่นเกม โดยในที่นี้ Host เป็นผู้เริ่มเล่นก่อน

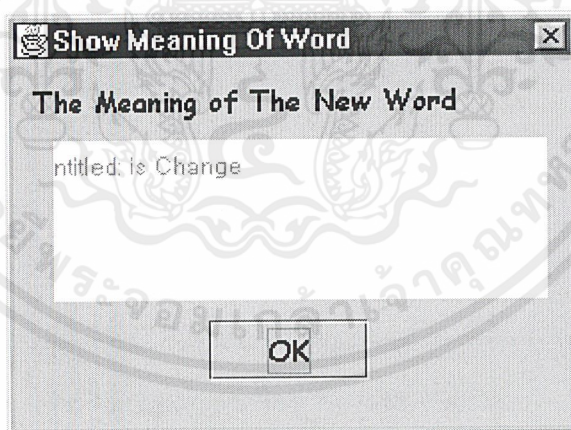
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) เมื่อผู้เล่นกดปุ่ม Change โปรแกรมจะแสดงจอภาพดังรูปที่ 4-11 เพื่อให้ผู้เล่นเลือกตัวอักษรที่ผู้เล่นต้องการจะเปลี่ยน



รูปที่ 4-11 แสดงจอภาพที่ใช้ในการเปลี่ยนตัวอักษรของผู้เล่น

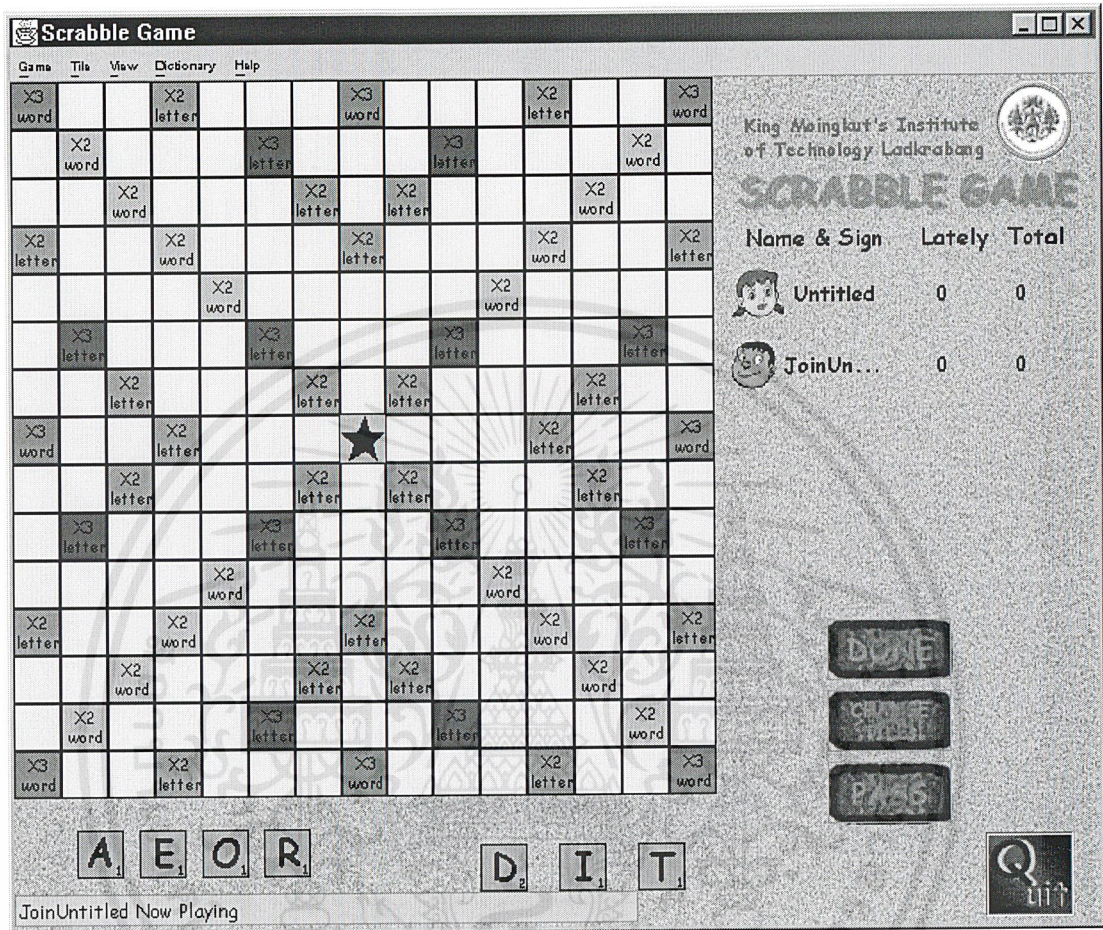
เมื่อผู้เล่นกดปุ่ม OK ที่เครื่องของผู้เล่นคนอื่นๆ โปรแกรมจะแสดงจอภาพดังรูป 4-12 เพื่อแสดงว่าผู้เล่นที่อะไรได้ทำการเปลี่ยนตัวอักษร



รูปที่ 4-12 แสดงจอภาพเมื่อมีผู้เล่นคนใดคนหนึ่งทำการเปลี่ยนตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

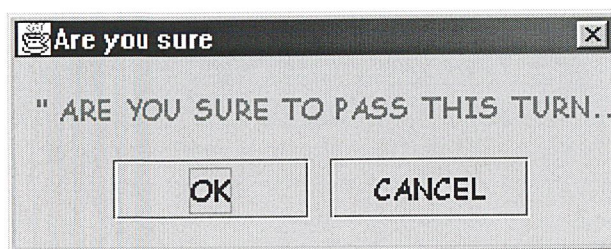
ส่วนที่เครื่องคอมพิวเตอร์ของผู้เล่นที่เป็นผู้เปลี่ยนตัวอักษรเอง โปรแกรมจะแสดงจอภาพ ดังรูปที่ 4-13 จะเห็นว่าตัวอักษรเดิมที่ผู้เล่นไม่ได้เปลี่ยนกับตัวอักษรใหม่ที่เกิดขึ้นจากการสุ่มให้ โดยคอมพิวเตอร์ จะแยกออกจากกันอย่างชัดเจน



รูปที่ 4-13 แสดงจอภาพหลังจากที่คอมพิวเตอร์ทำการสุ่มตัวอักษรให้ใหม่

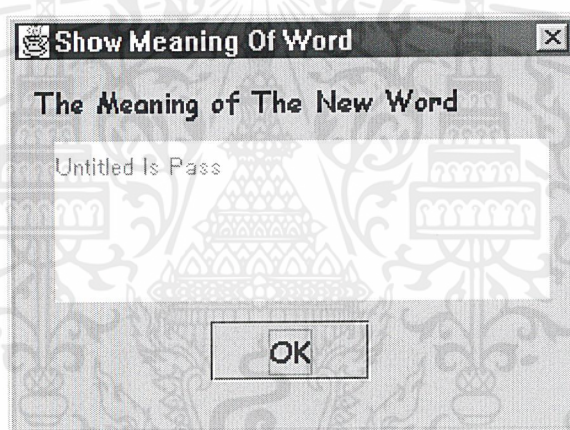
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) เมื่อผู้เล่นกดปุ่ม Pass โปรแกรมจะแสดงจอภาพดังรูปที่ 4-14 เพื่อให้ผู้เล่นทำการยืนยันว่าต้องการจะผ่านตาเล่นนั้นจริงๆ



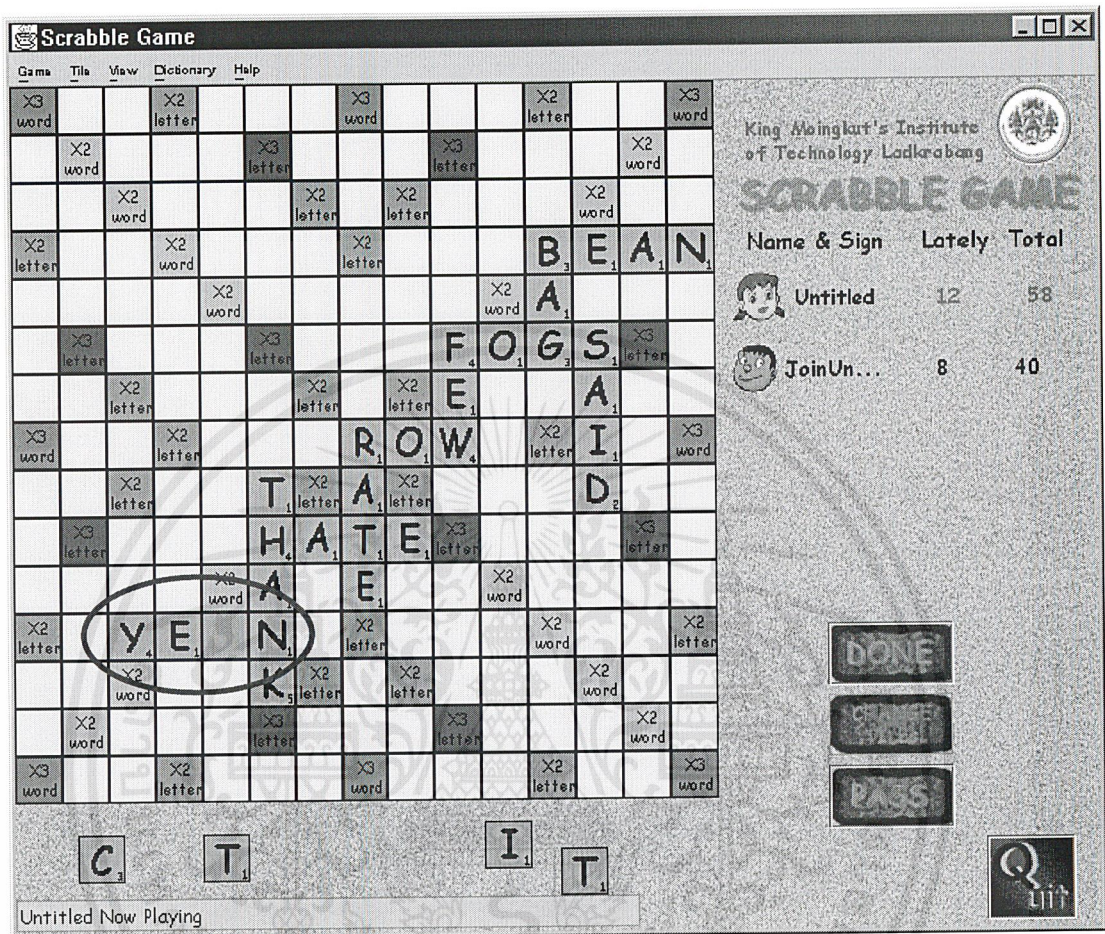
รูปที่ 4-14 แสดงจอภาพเพื่อให้ผู้เล่นยืนยันการผ่านตาเล่น

เมื่อผู้เล่นกดปุ่ม OK ที่เครื่องของผู้เล่นคนอื่นๆ โปรแกรมจะแสดงจอภาพดังรูป 4-15 เพื่อแสดงว่าผู้เล่นที่อะไรได้ทำการผ่านตาเล่นในครั้งนั้น



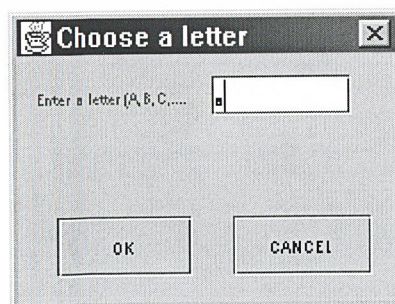
รูปที่ 4-15 แสดงจอภาพเมื่อผู้เล่นคนใดคนหนึ่งผ่านตาเล่น

9) ถ้าผู้เล่นเลือกตัวอักษรไปวางที่กระดาน โปรแกรมจะแสดงจอภาพดังรูปที่ 4-16 โดยที่ถ้าผู้เล่นนำตัวอักษรไปวางในกระดานแล้วจะไม่สามารถเปลี่ยนตัวอักษรได้จนกว่าจะนำตัวอักษรออกมามองตารางหมดก่อนจึงจะสามารถเปลี่ยนตัวอักษรได้



รูปที่ 4-16 แสดงจอภาพกรณีที่ผู้เล่นนำตัวอักษรไปวางบนกระดาน

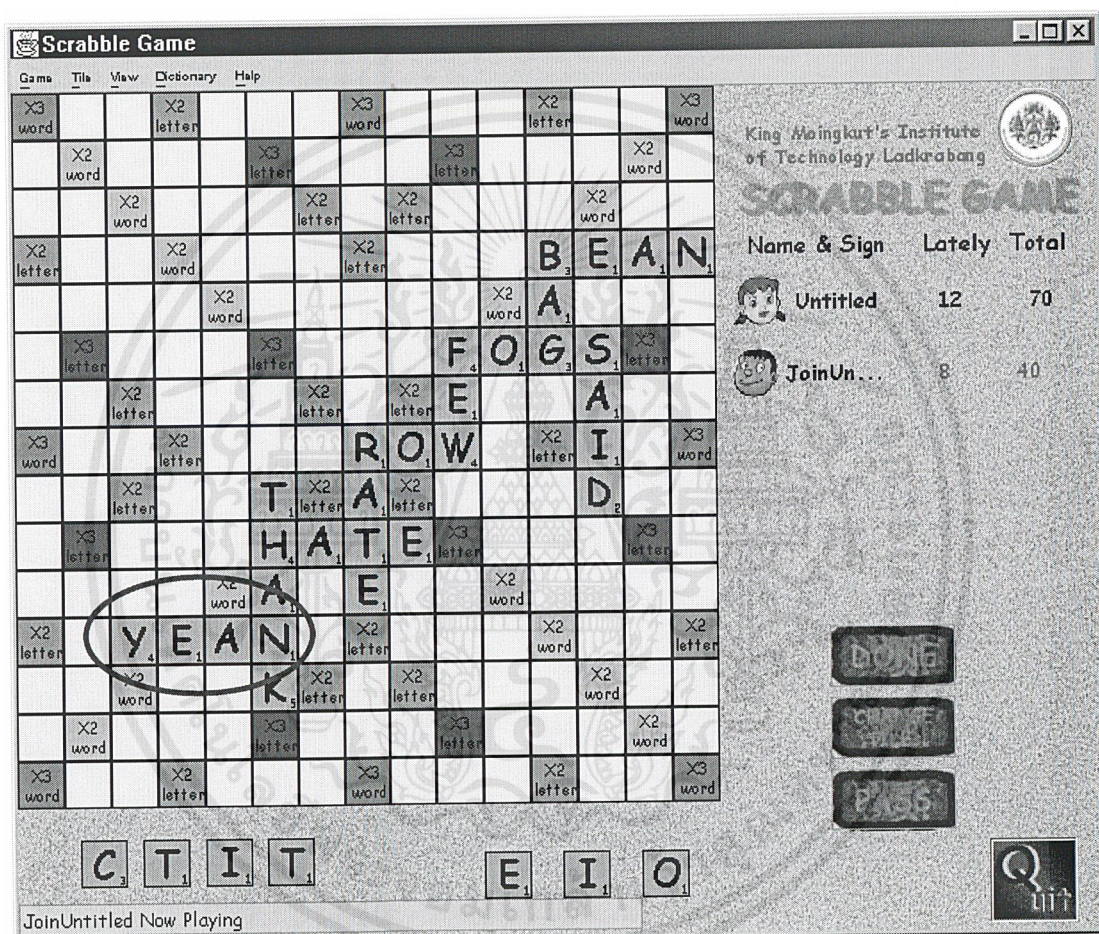
หากตัวอักษรที่ผู้เล่นลากขึ้นไปวางบนตารางเป็นตัว Blank โปรแกรมจะแสดงจอภาพดังรูปที่ 4-17 เพื่อให้ผู้เล่นกรอกตัวอักษรที่ต้องการให้ตัว Blank นั้นเป็นลงไป



รูปที่ 4-17 แสดงจอภาพให้ผู้เล่นกรอกตัวอักษรที่ต้องการแทนตัว Blank

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

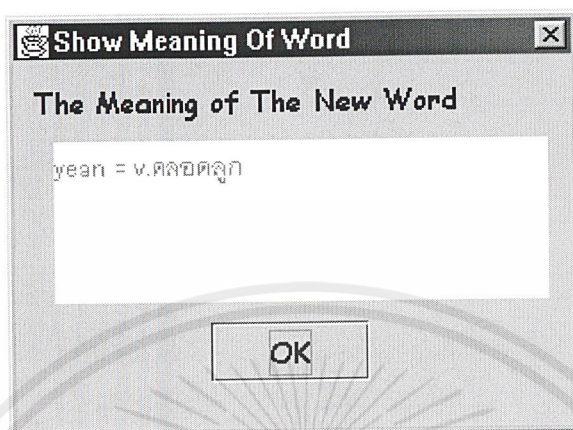
10) ถ้าผู้เล่นทำการกดยืนยันการลงโดยกด Done แล้วโปรแกรมจะทำการหาตำแหน่งว่าผู้เล่นลงถูกตำแหน่งและคำศัพท์ที่ลงไปนั้นมีความหมายในพจนานุกรมหรือไม่ หากถูกต้อง โปรแกรมจะทำการแปลงตัวอักษรที่ลงไปให้เป็นสีน้ำตาล เพื่อแสดงว่าผู้เล่นได้ลงคำศัพท์นั้นแล้ว ดังรูปที่ 4-18 จากนั้นโปรแกรมจะทำการสุ่มตัวอักษรใหม่ให้กับผู้เล่นเป็นจำนวนเท่ากับจำนวนตัวอักษรที่ลงไป หรือเท่ากับจำนวนตัวอักษรทั้งหมดที่เหลือในถุง (หากจำนวนตัวอักษรที่ผู้เล่นลงไปมีมากกว่าจำนวนตัวอักษรที่เหลือทั้งหมด) โดยโปรแกรมจะทำการแยกส่วนของตัวอักษรที่คอมพิวเตอร์ทำการสุ่มให้ใหม่กับตัวอักษรเดิมที่มีอยู่ไว้อย่างชัดเจน



รูปที่ 4-18 แสดงจอภาพเมื่อคำศัพท์ที่ผู้เล่นลงถูกต้อง

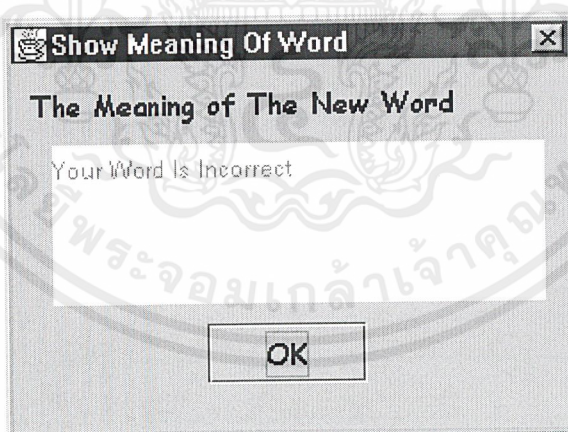
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และที่เครื่องคอมพิวเตอร์ของผู้เล่นทุกคนจะปรากฏจอภาพแสดงความหมายของคำศัพท์ทุกคำที่เกิดขึ้น ดังรูปที่ 4-19 รวมถึงโปรแกรมจะทำการคำนวณคะแนนที่เกิดขึ้นจากการลงคำศัพท์ที่ค้ำนั้นๆ ด้วย



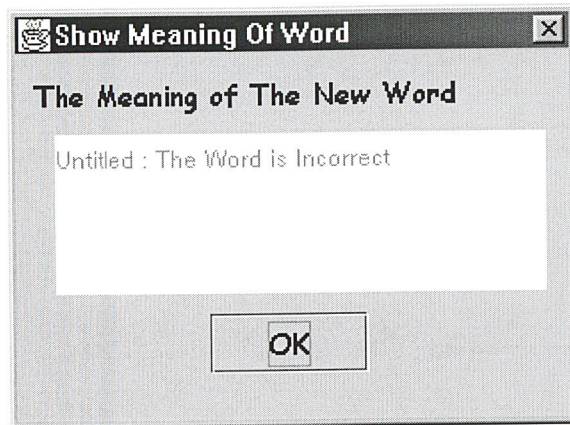
รูปที่ 4-19 แสดงจอภาพเพื่อบอกความหมายของคำศัพท์ทุกคำที่เกิดขึ้น

แต่หากคำศัพท์ที่ผู้เล่นลงมามีคำใดคำหนึ่งไม่ถูกต้อง หรือมีการลงผิดตำแหน่ง โปรแกรมจะแสดงจอภาพดังรูปที่ 4-20 (ที่เครื่องคอมพิวเตอร์ของผู้เล่นที่เป็นผู้ลงคำศัพท์นั้น) และ 4-21 (ที่เครื่องคอมพิวเตอร์ของผู้เล่นคนอื่นๆ ที่เหลือ)



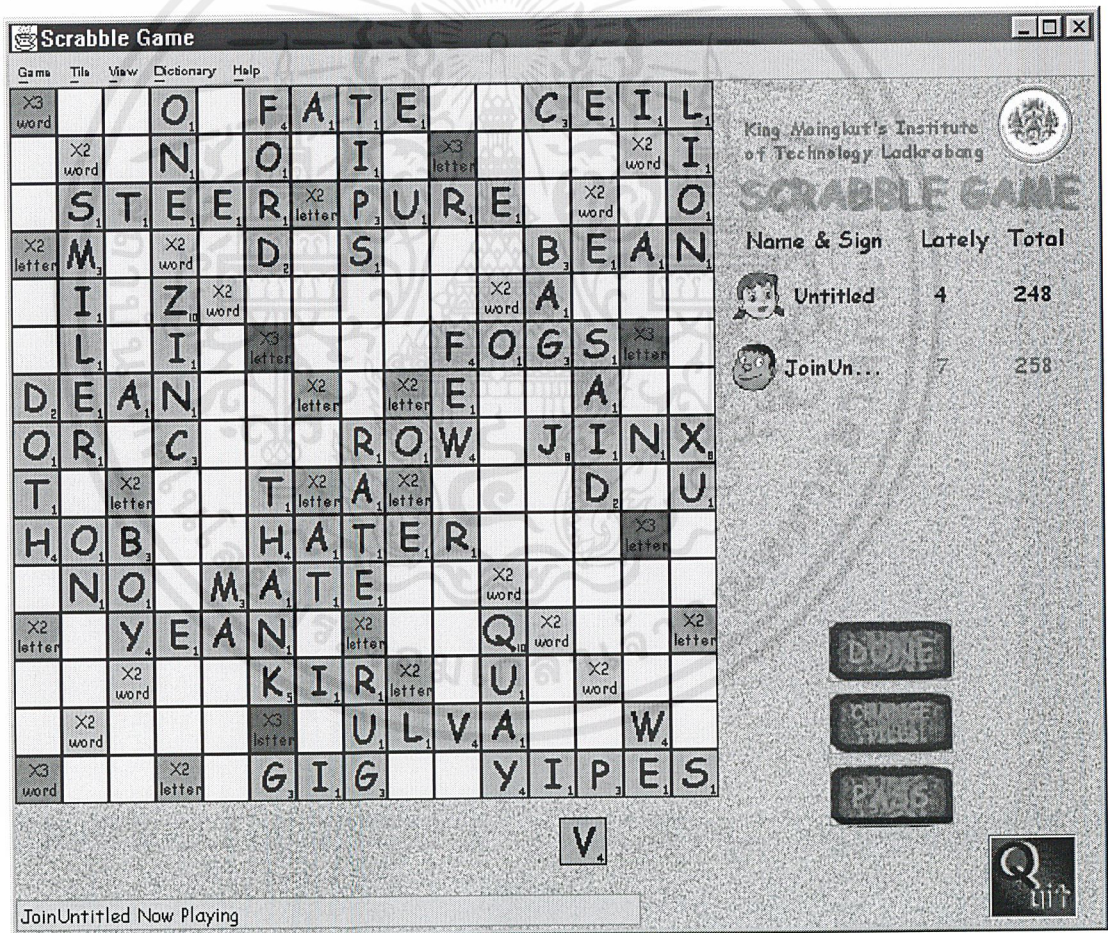
รูปที่ 4-20 แสดงจอภาพเพื่อบอกว่าผู้เล่นลงคำศัพท์ไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-21 แสดงจอภาพเมื่อมีผู้เล่นคนใดคนหนึ่งลงคำศัพท์ที่ไม่ถูกต้อง

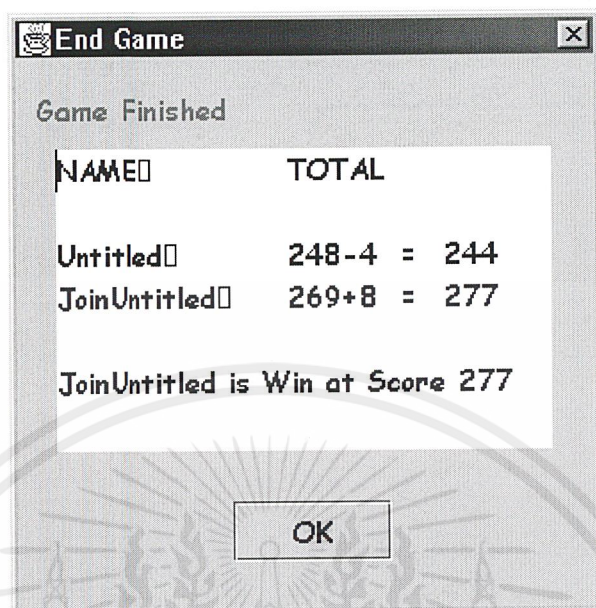
หลังจากผู้เล่นแต่ละคนได้เล่นเกมตามขั้นตอนต่างๆ ไปสักระยะหนึ่ง จะเป็นดังรูปที่ 4-22



รูปที่ 4-22 แสดงจอภาพหลังจากเล่นเกมไปได้สักระยะหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11) เมื่อเกมจบ (ทั้งที่จบตามกติกาหรือจบแบบผ่านตาเล่น 3 ครั้งติดกัน) โปรแกรมจะแสดงจอภาพดังรูปที่ 4-23 เพื่อบอกคะแนนของผู้เล่นแต่ละคน



รูปที่ 4-23 แสดงจอภาพสรุปคะแนนเมื่อเกมจบ

และโปรแกรมจะแสดงจอภาพดังรูปที่ 4-24 เพื่อแสดงลำดับคะแนนสูงสุด 5 ลำดับตามจำนวนของผู้เล่นในครั้งนั้น



รูปที่ 4-24 แสดงจอภาพเพื่อบอกถึงลำดับคะแนนสูงสุด 5 อันดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การอภิปรายผล

หลังจากทดสอบเล่นเกมต่อคำศัพท์ภาษาอังกฤษเรียบร้อยแล้ว พบว่าผลที่ได้คือ โปรแกรมสามารถทำงานได้อย่างถูกต้องตามขั้นตอนที่ได้กำหนดไว้ แต่จะเกิดความล่าช้าใน ขั้นตอนการส่งข้อความไปยังเครื่องคอมพิวเตอร์แต่ละเครื่อง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

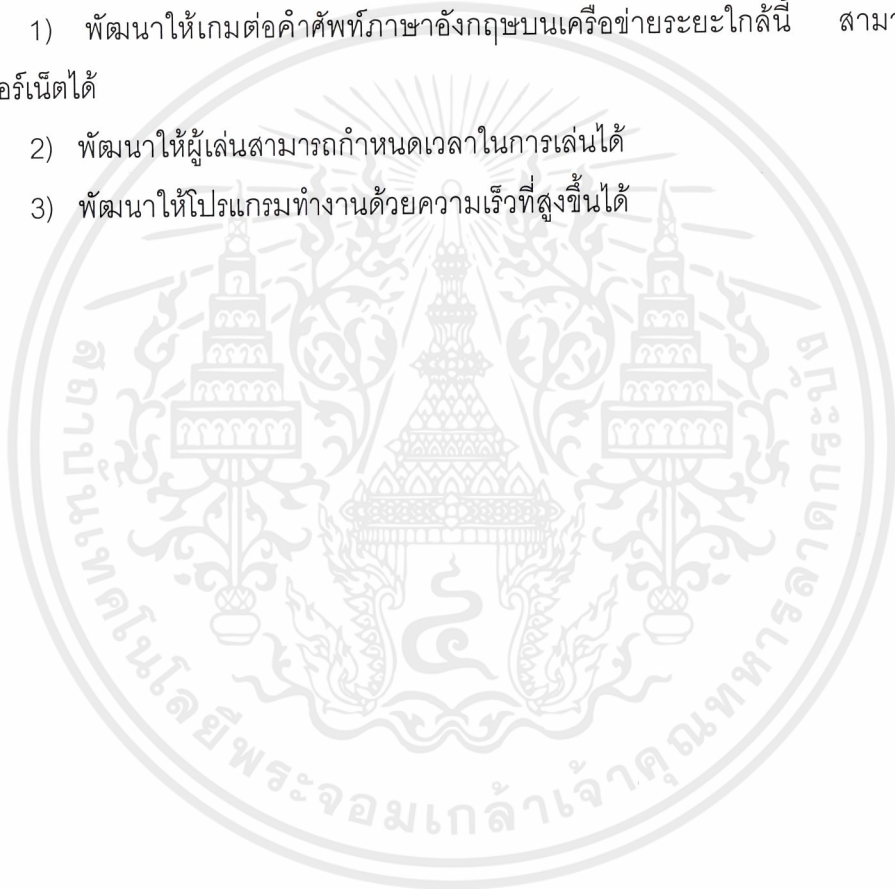
บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

การพัฒนาโปรแกรมเกมต่อคำศัพท์ภาษาอังกฤษบนเครือข่ายระยะใกล้ได้ดำเนินการตามวัตถุประสงค์ คือ สามารถรองรับผู้เล่นจำนวน 2 ถึง 4 คนผ่านโปรโตคอล TCP/IP ได้ ซึ่งผู้เล่นจะเกิดทักษะทางด้านภาษาอังกฤษมากขึ้นหรือน้อยขึ้นอยู่กับผู้เล่นเอง

อนึ่ง การพัฒนาโปรแกรมเกมต่อคำศัพท์ภาษาอังกฤษบนเครือข่ายระยะใกล้มีข้อเสนอแนะซึ่งเป็น แนวทางสำหรับผู้ที่ต้องการพัฒนาต่อไป ดังนี้

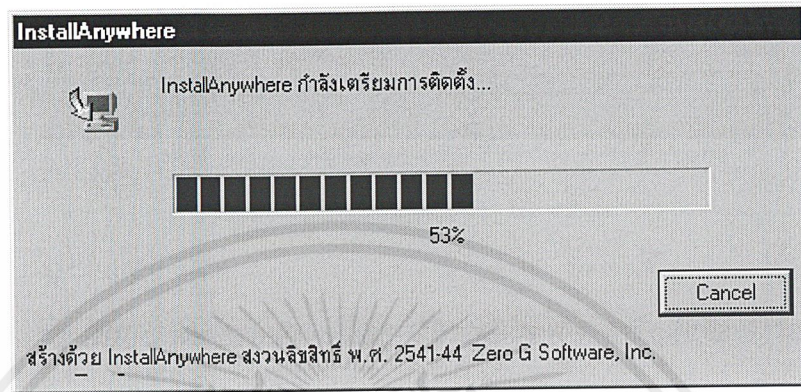
- 1) พัฒนาให้เกมต่อคำศัพท์ภาษาอังกฤษบนเครือข่ายระยะใกล้นี้ สามารถเล่นผ่านอินเทอร์เน็ตได้
- 2) พัฒนาให้ผู้เล่นสามารถกำหนดเวลาในการเล่นได้
- 3) พัฒนาให้โปรแกรมทำงานด้วยความเร็วที่สูงขึ้นได้



ภาคผนวก ก

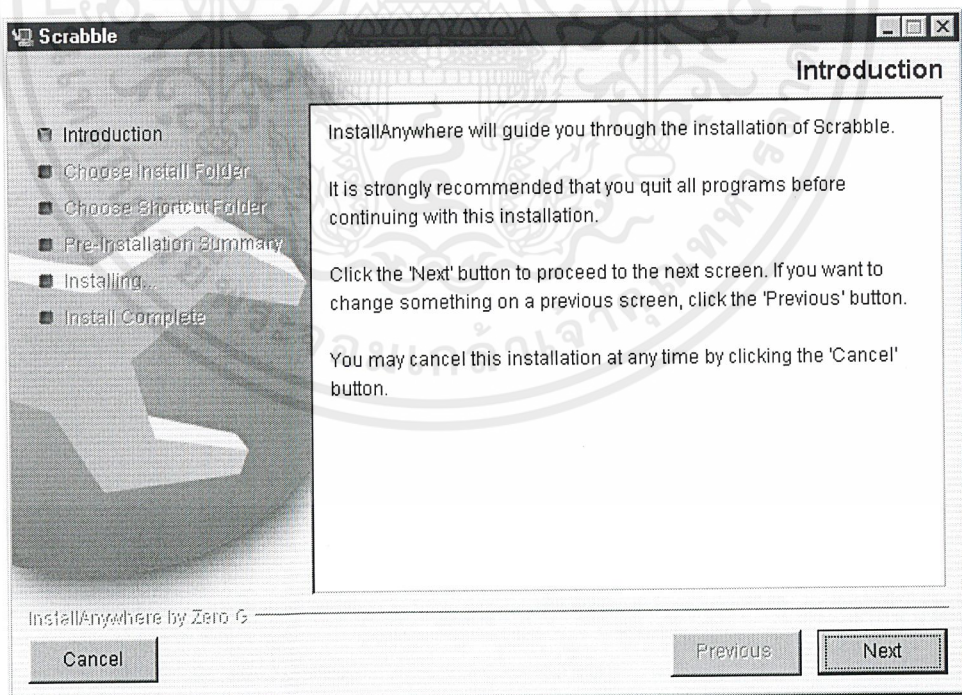
การติดตั้งโปรแกรมต่อคำศัพท์ภาษาอังกฤษ

- 1) ดับเบิลคลิกไอคอนที่รูป Install จะปรากฏจอภาพดังรูปที่ ก-1



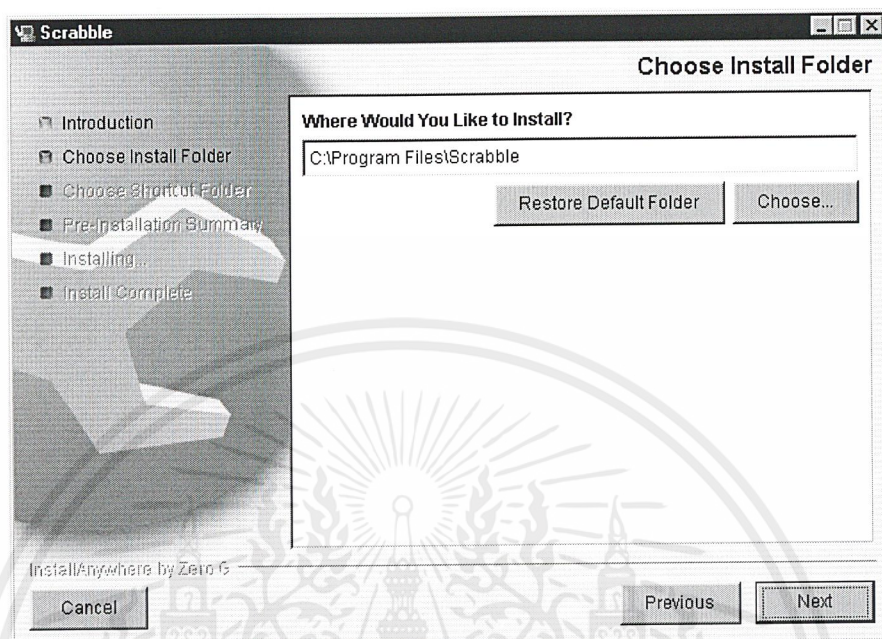
รูปที่ ก-1 แสดงจอภาพแรกเมื่อเริ่มทำการติดตั้งโปรแกรม

- 2) เมื่อโปรแกรมทำงานเรียบร้อยแล้วจะปรากฏจอภาพ Introduction ดังรูปที่ ก-2 จากนั้นให้กดปุ่ม Next



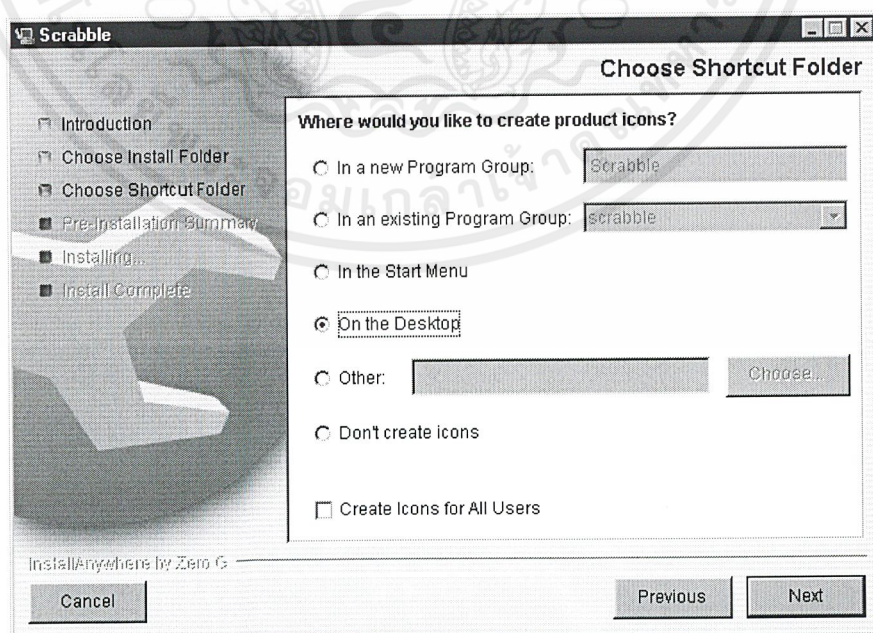
รูปที่ ก-2 แสดงจอภาพ Introduction

- 3) เลือกโฟลเดอร์ที่ต้องการติดตั้งโปรแกรม Scrabble.exe โดยโฟลเดอร์ที่กำหนดให้อัตโนมัติ คือ C:\Program Files\Scrabble แต่ถ้าต้องการติดตั้งที่โฟลเดอร์อื่นคลิก Choose เมื่อเลือกเรียบร้อยแล้ว



รูปที่ ก-3 แสดงจอภาพให้เลือกโฟลเดอร์ที่ต้องการติดตั้งโปรแกรม Scrabble.exe

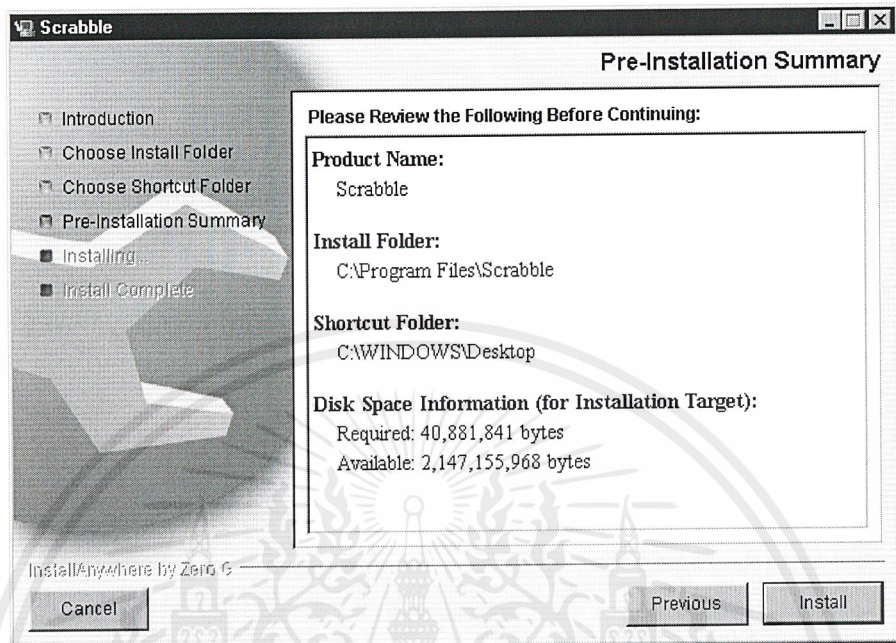
- 4) เลือกโฟลเดอร์ที่ต้องการให้ Shortcut ของโปรแกรม Scrabble.exe อยู่ โดยที่ค่าอัตโนมัติ คือ On the Desktop หลังจากนั้นคลิกที่ Next



รูปที่ ก-4 แสดงจอภาพให้เลือก Shortcut Folder

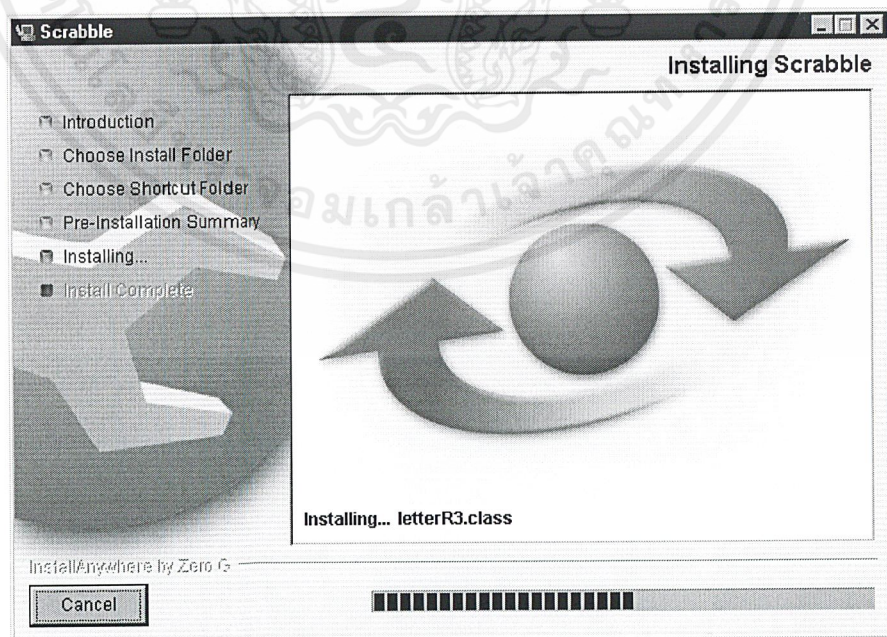
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) จากรูปที่ ก-5 เป็นการแสดงรายละเอียดต่างๆ คือ ชื่อโปรแกรมที่ติดตั้ง , โฟลเดอร์ที่ติดตั้ง , โฟลเดอร์ของ shortcut , ขนาดของโปรแกรม เมื่อดูรายละเอียดว่าถูกต้องแล้ว ให้กดปุ่ม Install



รูปที่ ก-5 แสดงจอภาพที่มีรายละเอียดต่างๆ ของการติดตั้ง

6) โปรแกรมจะทำการติดตั้งโปรแกรม Scrabble.exe ไปยังโฟลเดอร์ที่เลือกไว้ เมื่อติดตั้งเรียบร้อยแล้ว จะปรากฏจอภาพดังรูปที่ ก-6



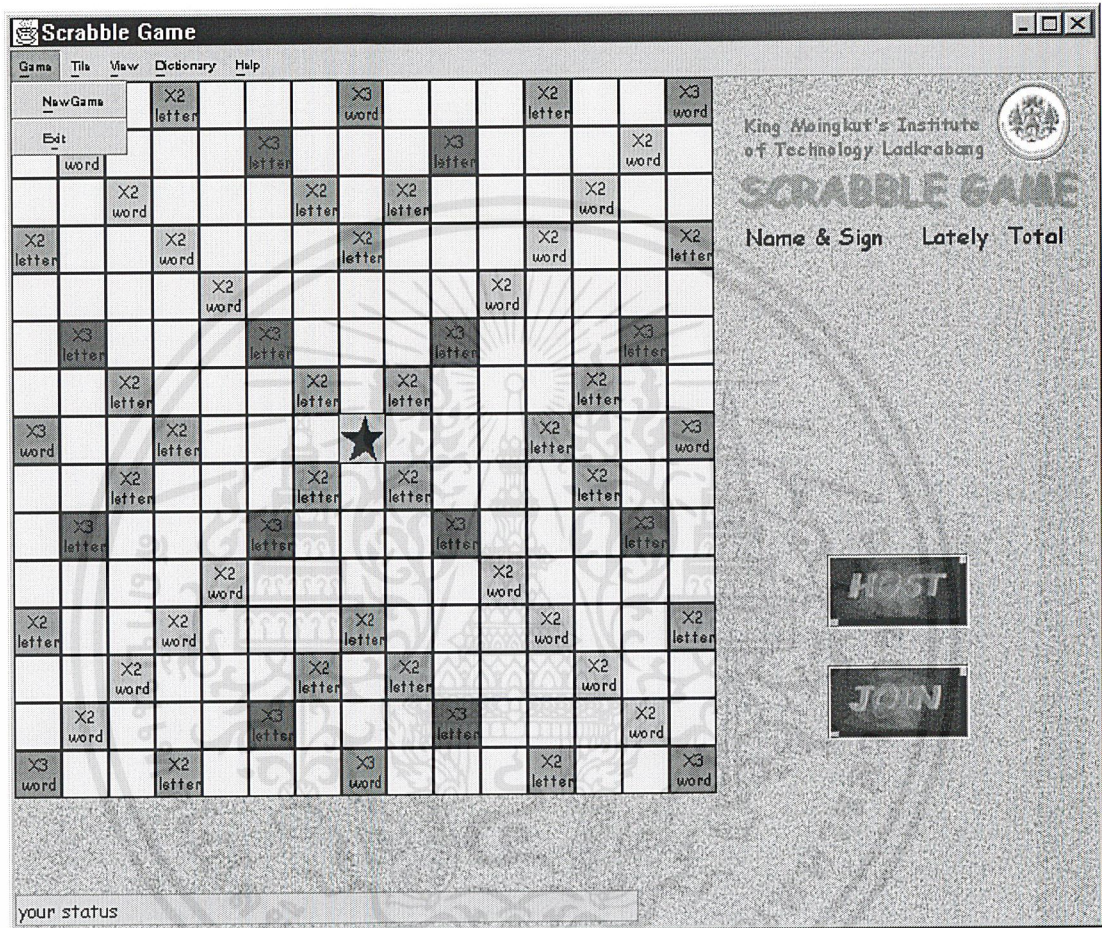
รูปที่ ก-6 แสดงจอภาพเมื่อทำการติดตั้งโปรแกรม Scrabble.exe สมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

คำอธิบายส่วนเมนูบาร์ของโปรแกรม

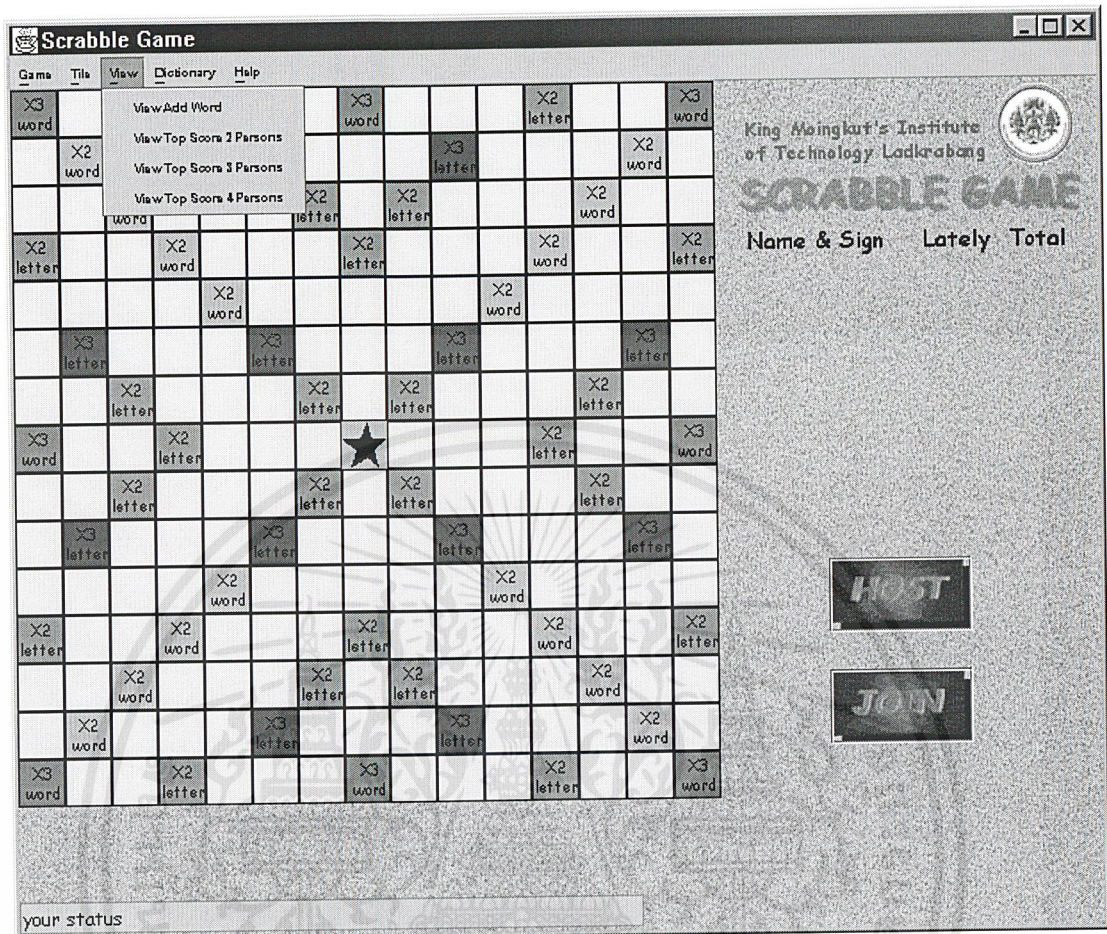
1) เมื่อผู้เล่นเลือกเมนู Game โปรแกรมจะแสดงจอภาพดังรูปที่ ข-1



รูปที่ ข-1 แสดงจอภาพเมื่อผู้เล่นเลือกที่เมนู Game

- NewGame : ผู้เล่นต้องการที่จะเริ่มเล่นเกมใหม่
- Exit : ผู้เล่นต้องการที่จะออกจากโปรแกรม

3) เมื่อผู้เล่นเลือกเมนู View โปรแกรมจะแสดงจอภาพดังรูปที่ ข-3

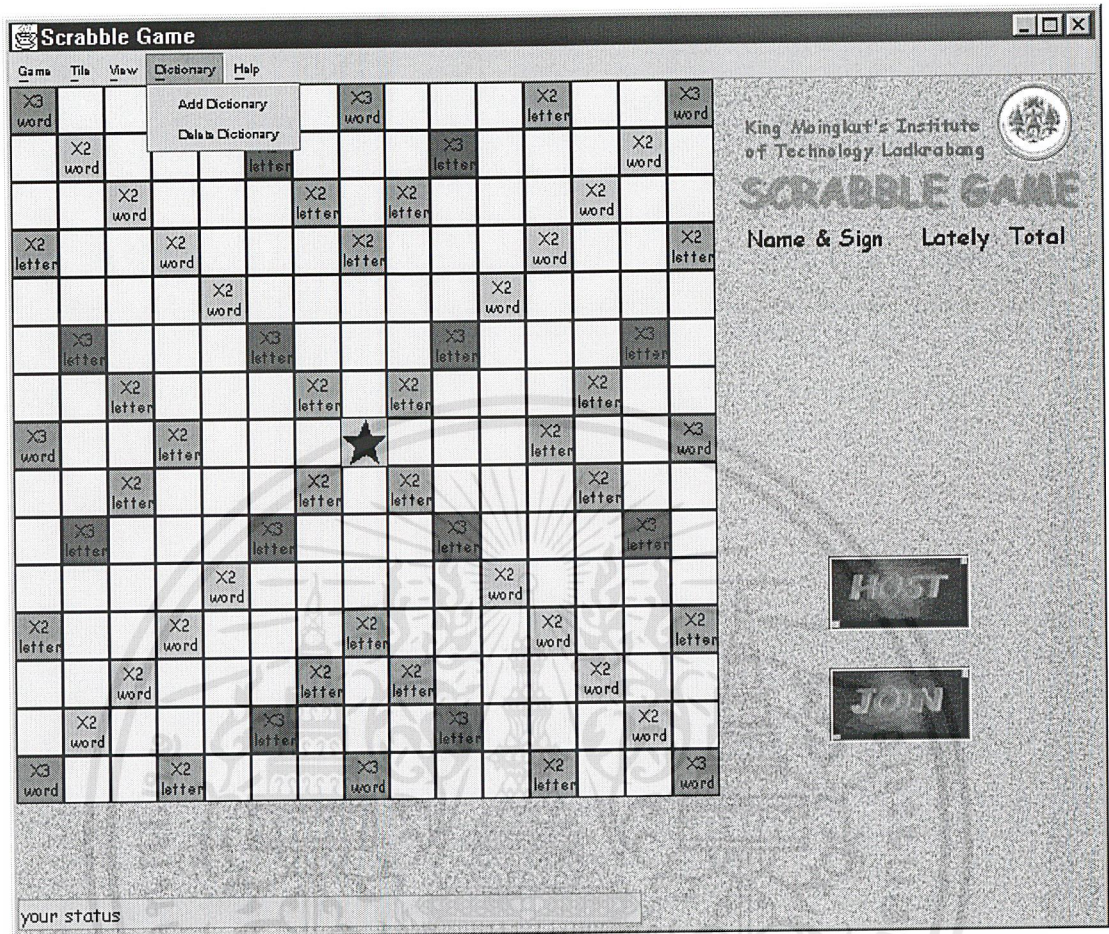


รูปที่ ข-3 แสดงจอภาพเมื่อผู้เล่นเลือกเมนู View

- View Add Word : ผู้เล่นเลือกเพื่อให้เห็นคำศัพท์ที่ผู้เล่นที่เป็น Host ได้ทำการเพิ่มนอกเหนือจากที่มีในพจนานุกรมฉบับมาตรฐาน โดยจะสามารถดูได้เฉพาะตอนก่อนเริ่มเล่นเกมนั้นๆ
- View Top Score 2 Persons : ผู้เล่นเลือกเพื่อให้เห็นคะแนนสูงสุดของผู้เล่นที่เล่นกัน 2 คน มีลำดับสูงสุด 5 อันดับ
- View Top Score 3 Persons : ผู้เล่นเลือกเพื่อให้เห็นคะแนนสูงสุดของผู้เล่นที่เล่นกัน 3 คน มีลำดับสูงสุด 5 อันดับ
- View Top Score 4 Persons : ผู้เล่นเลือกเพื่อให้เห็นคะแนนสูงสุดของผู้เล่นที่เล่นกัน 4 คน มีลำดับสูงสุด 5 อันดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

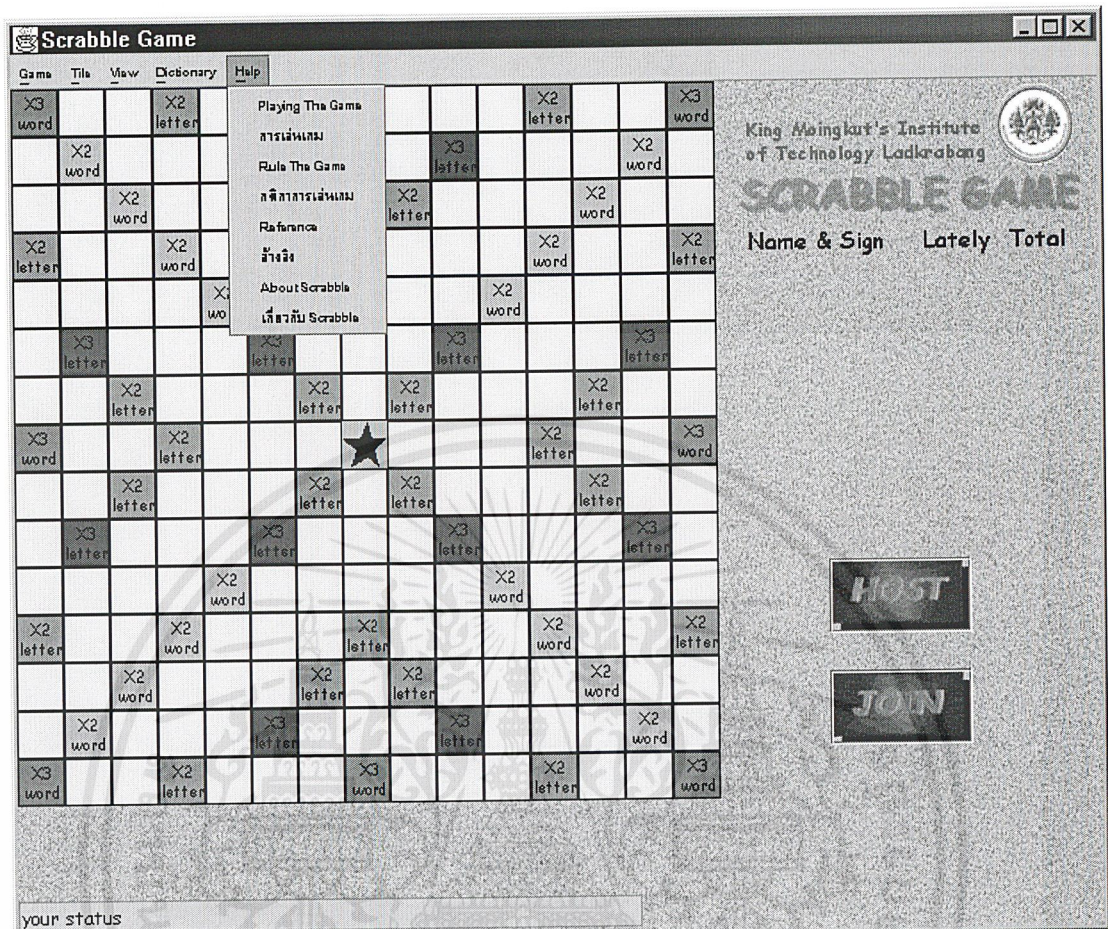
4) เมื่อผู้เล่นเลือกเมนู Dictionary โปรแกรมจะแสดงจอภาพดังรูปที่ ข-4



รูปที่ ข-4 แสดงจอภาพเมื่อผู้เล่นเลือกเมนู Dictionary

- Add Dictionary : ผู้เล่นสามารถเพิ่มคำศัพท์เข้าไปได้ โดยคำศัพท์ที่เพิ่มนั้นไม่ได้ถูกนำไปรวมกับคำศัพท์ที่มีอยู่ในพจนานุกรมฉบับมาตรฐาน แต่จะถูกสร้างขึ้นมาเพิ่มเติมจากพจนานุกรมฉบับมาตรฐาน โดย ผู้เล่นที่เป็น Host เท่านั้นที่สามารถทำการเพิ่มคำศัพท์เข้าไปได้
- Delete Dictionary : ผู้เล่นสามารถลบคำศัพท์ได้ โดยคำศัพท์ที่สามารถลบได้เป็นเฉพาะคำที่สร้างขึ้นมาเพิ่มเติมจากพจนานุกรมฉบับมาตรฐาน โดยผู้เล่นที่เป็น Host เท่านั้นที่สามารถทำการลบคำศัพท์ได้

5) เมื่อผู้เล่นเลือกเมนู Help โปรแกรมจะแสดงจอภาพดังรูปที่ ๗-5

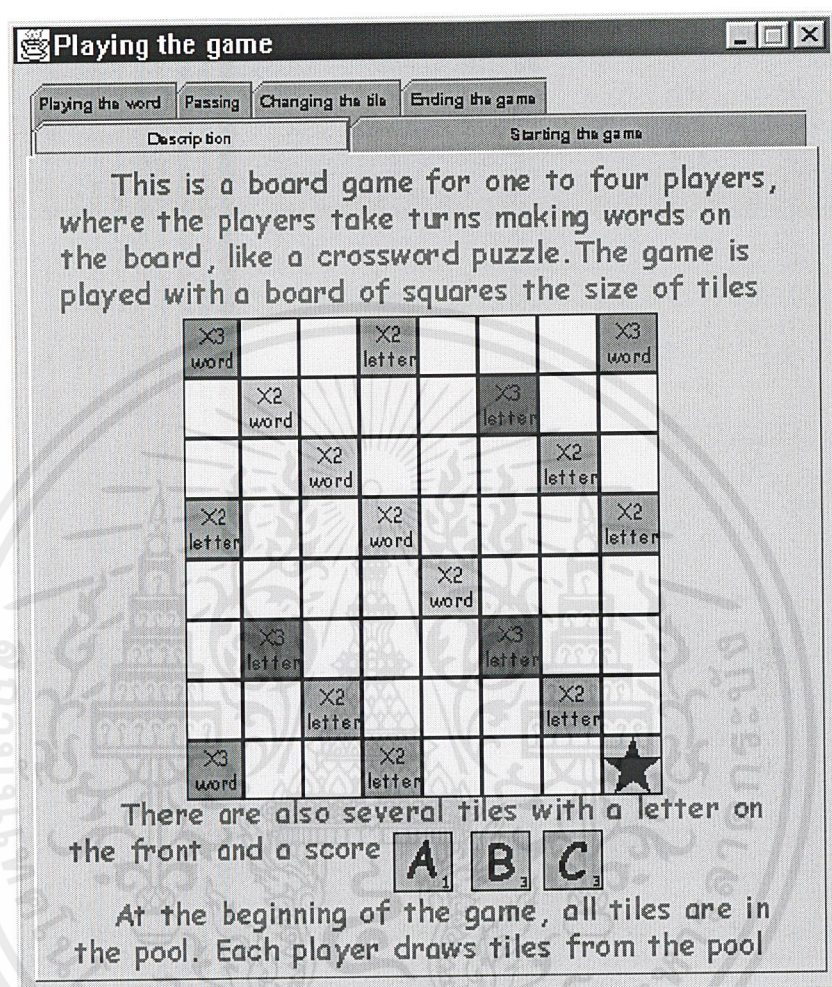


รูปที่ ๗-5 แสดงจอภาพเมื่อผู้เล่นเลือกเมนู Help

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1) เมื่อผู้เล่นเลือกเมนู Playing The Game โปรแกรมจะแสดงจอภาพต่างๆ ดังต่อไปนี้

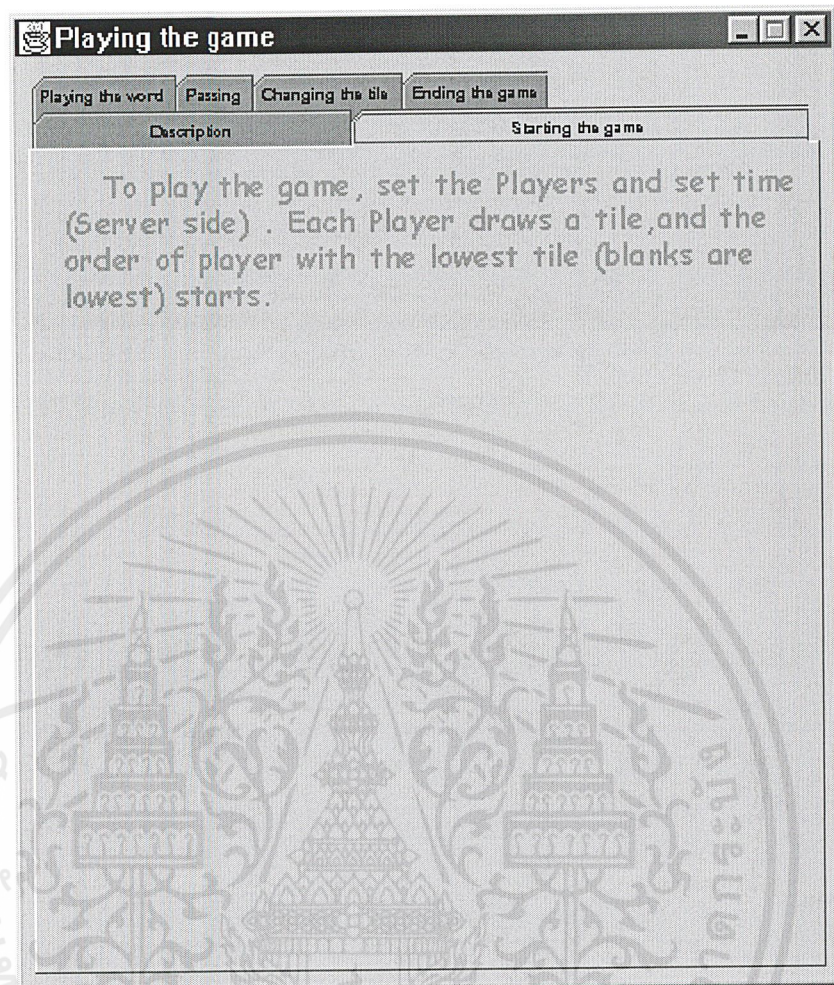
- Description : แสดงรายละเอียดของเกมดังรูปที่ ๗-6



รูปที่ ๗-6 แสดงจอภาพเมื่อผู้เล่นเลือกแท็บ Description

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

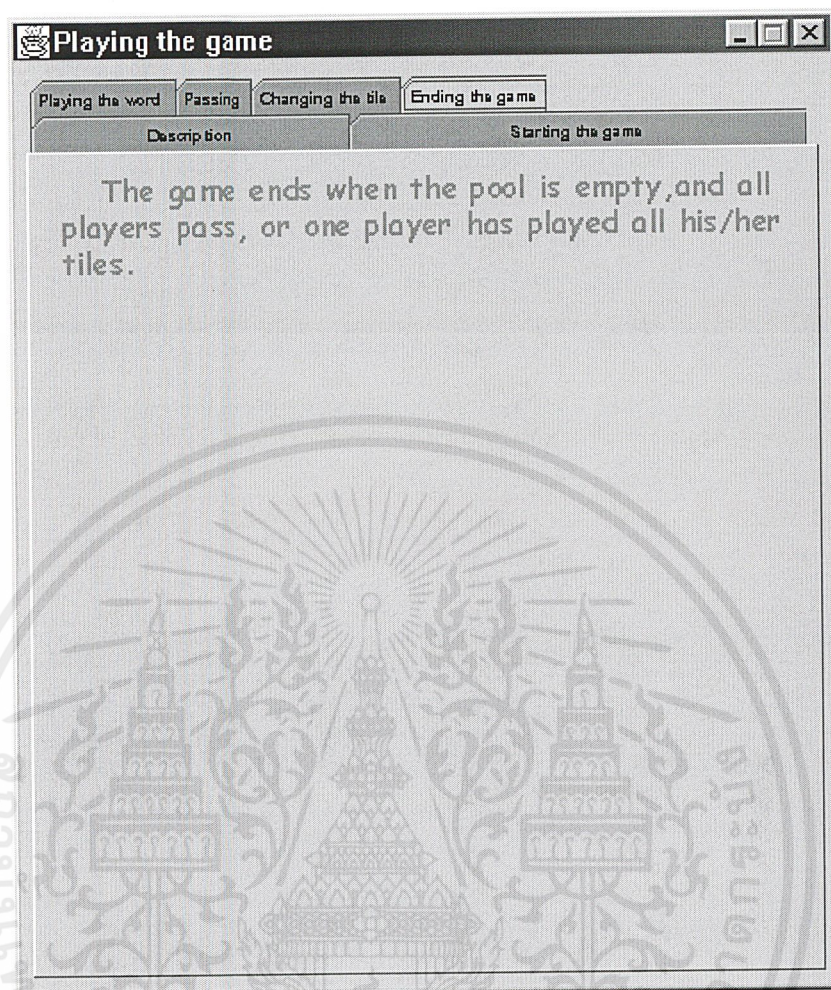
- Starting the game : แสดงวิธีเริ่มเล่นเกมดังรูปที่ ข-7



รูปที่ ข-7 แสดงจอภาพเมื่อผู้เล่นเลือกแท็บ Starting the game

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

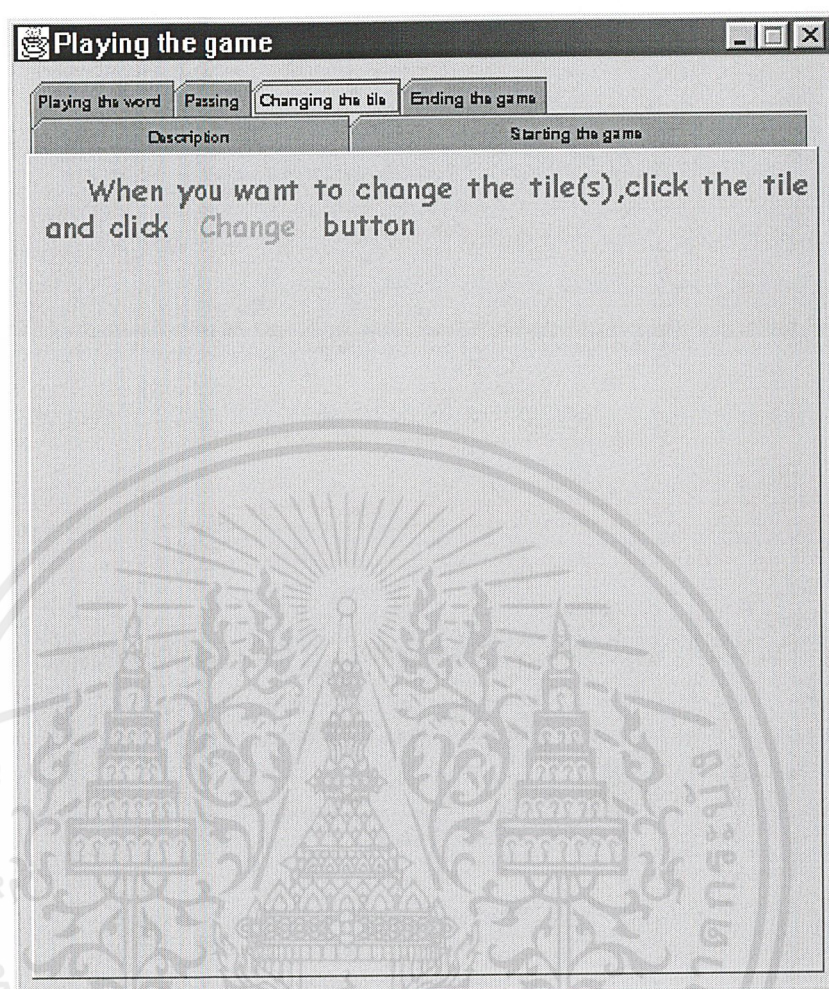
- Ending the game : อธิบายการจบเกมดังรูปที่ ข-8



รูปที่ ข-8 แสดงจอภาพเมื่อผู้เล่นเลือกแท็บ Ending the game

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

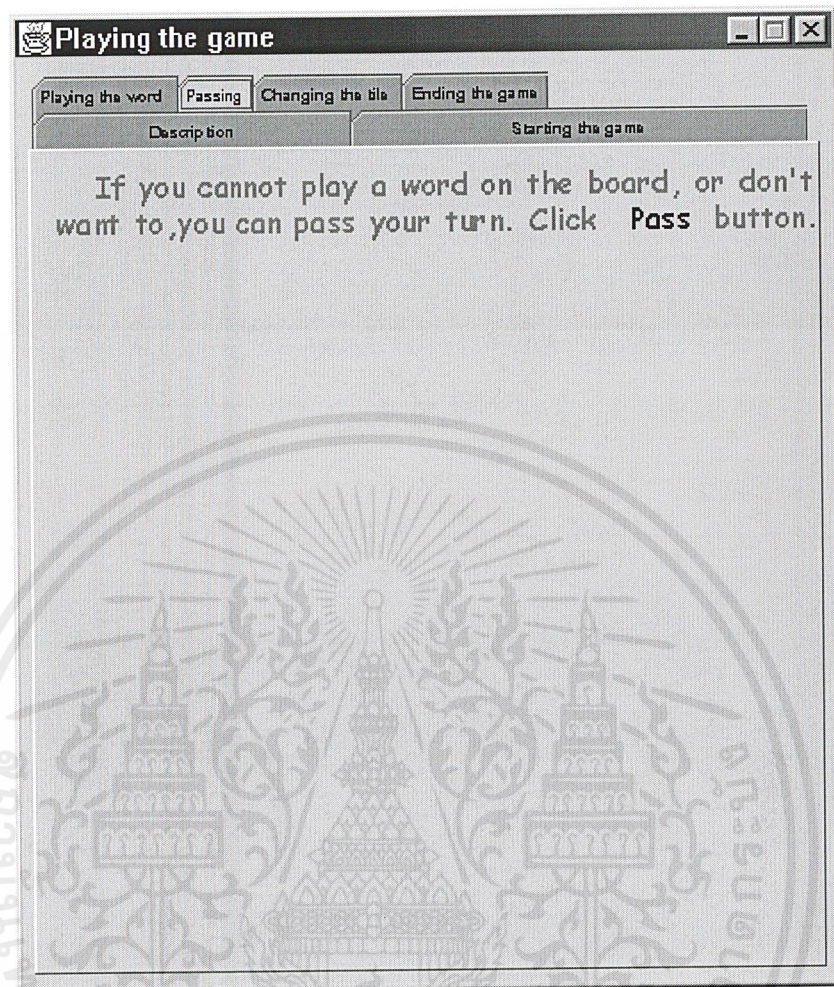
- Changing the tile : อธิบายวิธีการเปลี่ยนตัวอักษรของผู้เล่นดังรูปที่ ข-9



รูปที่ ข-9 แสดงจอภาพเมื่อผู้เล่นเลือกแท็บ Changing the tile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

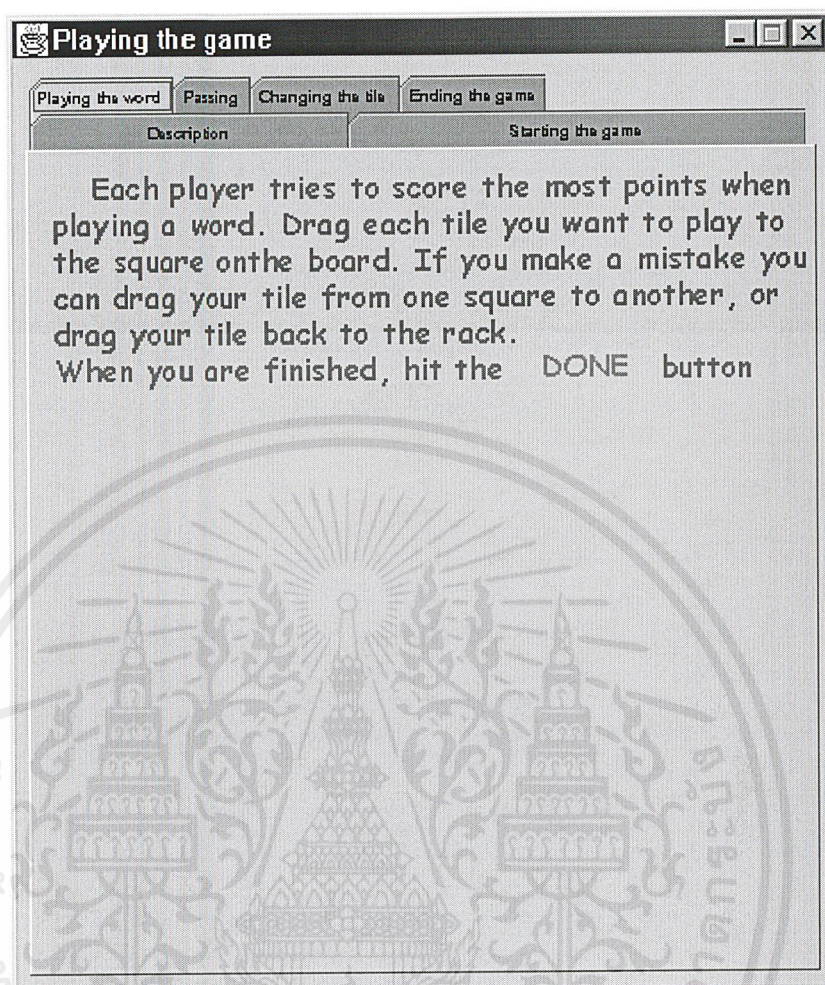
- Passing : อธิบายวิธีการผ่านตาเล่นนั้นไปดังรูปที่ ข-10



รูปที่ ข-10 แสดงจอภาพเมื่อผู้เล่นเลือกแท็บ Passing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Playing the word : อธิบายวิธีการลงคำศัพท์ในตาเล่นนั้นดังรูปที่ ข-11



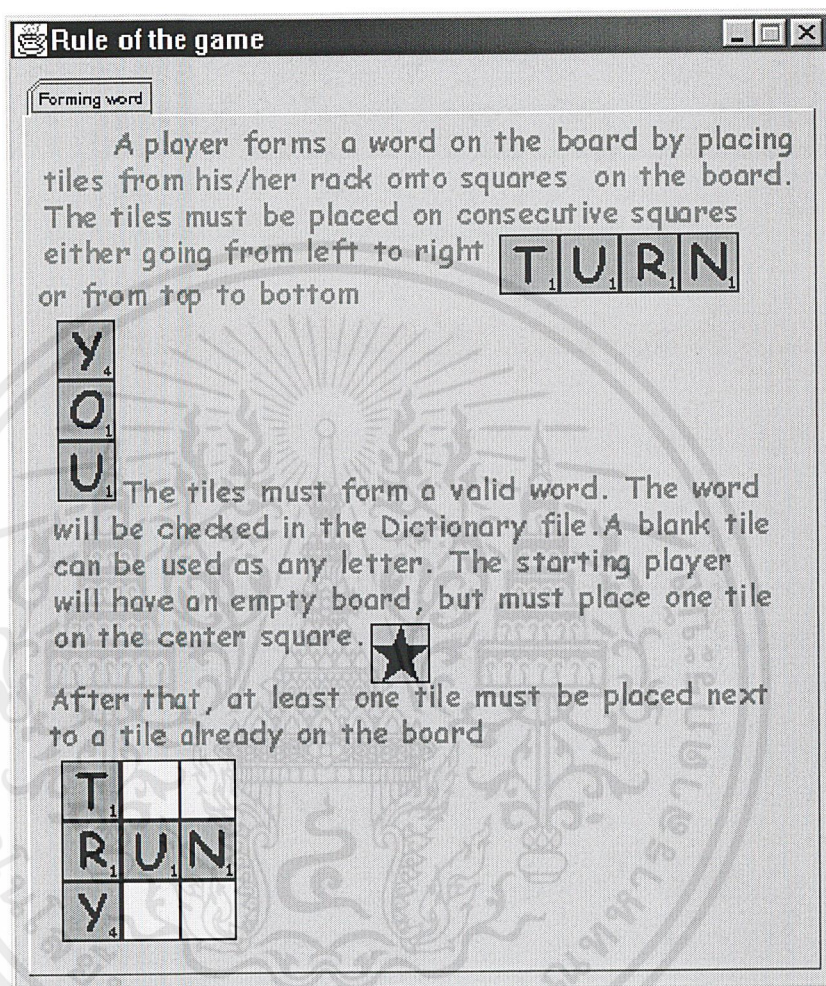
รูปที่ ข-11 แสดงจอภาพเมื่อผู้เล่นเลือกแท็บ Playing the word

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2) เมื่อผู้เล่นเลือกเมนู Rule The Game โปรแกรมจะแสดงจอภาพดังรูป

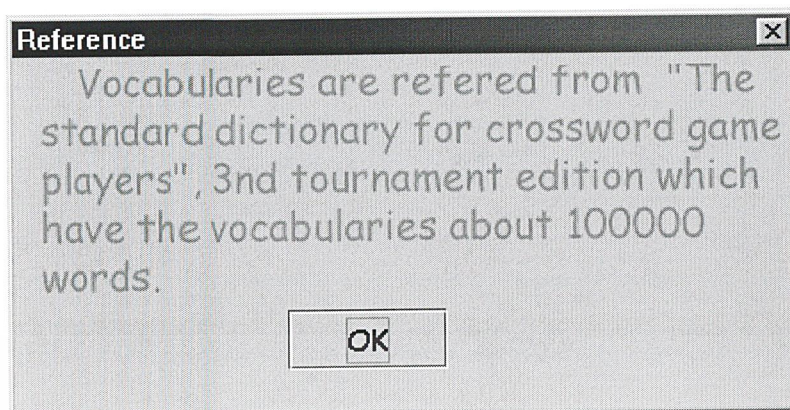
- Forming word : อธิบายรูปแบบของการวางตัวอักษรที่เราสามารถวางได้

ดังรูปที่ ข-12



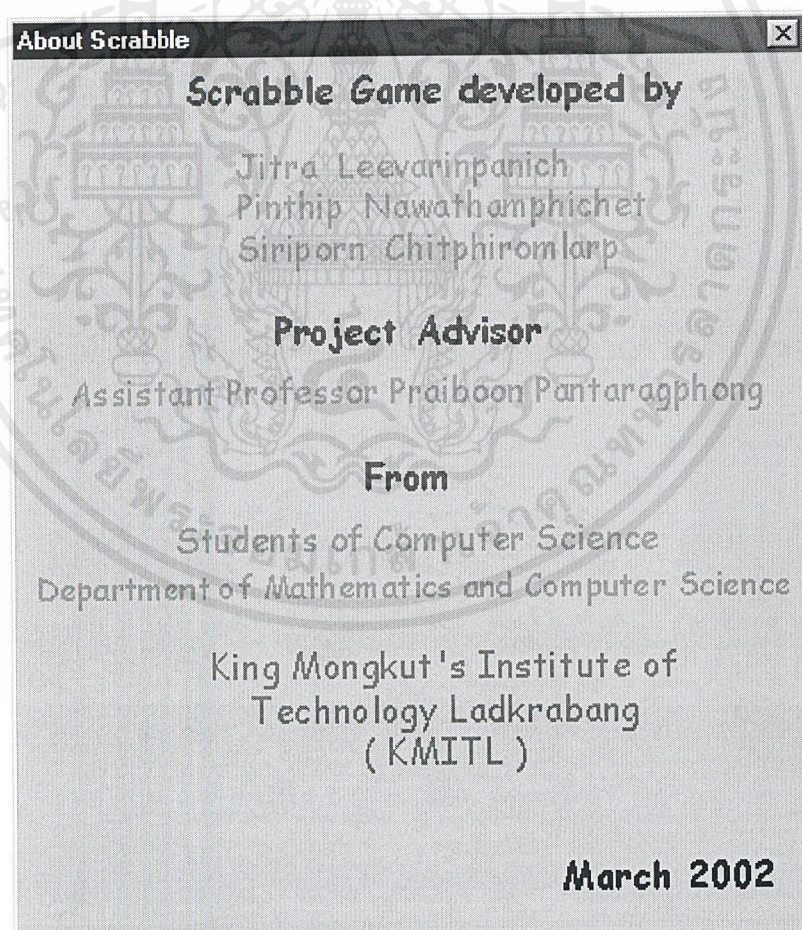
รูปที่ ข-12 แสดงจอภาพเมื่อผู้เล่นเลือกแท็บ Forming word

5.3) เมื่อผู้เล่นเลือกเมนู Reference โปรแกรมจะแสดงจอภาพดังรูปที่ ข-13



รูปที่ ข-13 แสดงจอภาพเมื่อผู้เล่นเลือกเมนู Reference

5.4) เมื่อผู้เล่นเลือกเมนู About Scrabble โปรแกรมจะแสดงจอภาพดังรูปที่ ข-14



รูปที่ ข-14 แสดงจอภาพเมื่อผู้เล่นเลือกเมนู About Scrabble

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ

เมนู “การเล่นเกม” อธิบายเหมือนเมนู Playing The Game แต่เป็นภาษาไทย
 เมนู “กติกาการเล่นเกม” อธิบายเหมือนเมนู Rule The Game แต่เป็นภาษาไทย
 เมนู “อ้างอิง” อธิบายเหมือนเมนู Reference แต่เป็นภาษาไทย
 เมนู “เกี่ยวกับ Scrabble” เหมือนเมนู About Scrabble แต่เป็นภาษาไทย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- กิตติ ภัคดีวัฒนกุล. 2544. Java ฉบับพื้นฐาน. KTP COMP&CONSULT.
- กิตติ ภัคดีวัฒนกุล. 2544. Java ฉบับโปรแกรมเมอร์. KTP COMP&CONSULT.
- สุวัฒน์ ปุณณรัชยะและคณะ. เปิดโลกของ TCP/IP และ โพรโตคอลของอินเทอร์เน็ต. Provision.
- อาจารย์อุดม จินประดับ และ ผศ.ดร.สมคิด เรืองธนะสกุลไทย. 2540. ทฤษฎีและตัวอย่าง โจทย์โครงสร้างข้อมูล. แมคกรอ-ฮิลอินเทอร์เน็ตเนชันแนลเอ็นเตอร์ไพรส์อิงค์.
- J.W. Spear & Sons LTD. Rules For Playing Scrabble. Enfield , Middlese : England



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้