

การพัฒนาเกมผ่านเครือข่ายสองมิติ

2D Game Online Development

นายกฤษณะ ศรีไกรรส รหัส 44362523
นายธงชัย ภู่ประภากร รหัส 44362598

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 25 / พ.ค. 2553 /

เลขทะเบียน..... 1504 2395

เลขเรียกหนังสือ..... ๖281๓

มหาวิทยาลัยนเรศวร 2547

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2547

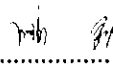


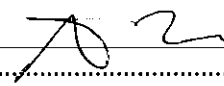
ใบรับรองโครงการวิศวกรรม

หัวข้อโครงการ การพัฒนาเกมผ่านเครือข่ายสองมิติ
ผู้ดำเนินโครงการ นายกฤษณะ ศรีไกรรอด รหัส 44362523
 นายธงชัย ภูประภากร รหัส 44362598
อาจารย์ที่ปรึกษา อาจารย์ พงศ์พันธ์ กิจสนาโยธิน
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2547

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์ อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะกรรมการสอบโครงการวิศวกรรม


.....ประธานกรรมการ
(อาจารย์พงศ์พันธ์ กิจสนาโยธิน)


.....กรรมการ
(ดร.พนมขวัญ รियะมงคล)


.....กรรมการ
(ดร.สมยศ เกียรติวนิชวิไล)

หัวข้อโครงการ	การพัฒนาเกมผ่านเครือข่ายสองมิติ		
ผู้ดำเนินโครงการ	นายกฤษณะ	ศรีไกรรส	รหัส 44362523
	นายธงชัย	ภูประภากร	รหัส 44362598
อาจารย์ที่ปรึกษา	อาจารย์พงศ์พันธ์ กิจสนาโยธิน		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2547		

บทคัดย่อ

เกมคอมพิวเตอร์ในปัจจุบันนั้นมีการพัฒนาในหลายรูปแบบ โดยมีทั้งเกมที่เล่นบนเครื่องคอมพิวเตอร์เพียงเครื่องเดียวและเกมออนไลน์ที่ผู้เล่นสามารถแข่งขันกับผู้เล่นคนอื่นๆผ่านเครือข่าย โดยโครงการนี้ได้ทำการศึกษาและประยุกต์ใช้โปรแกรม Macromedia Flash MX ในการพัฒนาพัฒนาเกมออนไลน์ที่สามารถรองรับผู้เล่นได้ 4 คนพร้อมกันบนเครือข่าย โดยใช้การติดต่อสื่อสารแบบซ็อกเก็ต(socket) ระหว่างแม่ข่าย(server) ที่เปิดบริการอยู่บนเครื่องคอมพิวเตอร์กับเครื่องลูกข่าย(client) โดยมีการส่งผ่านข้อมูลในรูปแบบ XML

ผลที่ได้จากการทำโครงการนี้ คือ เกมออนไลน์ที่สามารถทำงานได้บนเครือข่ายและเชื่อมต่อกันระหว่างแม่ข่ายกับเครื่องลูกข่าย สามารถรองรับการติดต่อสื่อสารกันแบบซ็อกเก็ตได้

Project Title	2D Game Online Development	
Name	Mr. Krishna Srikrairose	ID. 44362523
	Mr. Thongchai puprapakorn	ID. 44362598
Project Advisor	Mr. Phongphun Kijsanayothin	
Major	Computer Engineering	
Department	Electrical and Computer Engineering	
Academic Year	2004	

.....

ABSTRACT

Nowadays, computer games have been developed in many patterns. Players can play with computer or other players. The purpose of this project is to study and develop the application by Macromedia Flash MX. Four-player online game uses socket connection between Server and Client. This game uses XML as a communication protocol.

The result of this project is a online game, which can play on network and connect from a server to a personal computer. This application can also support socket connection.

กิตติกรรมประกาศ

โครงการวิศวกรรม การพัฒนาเกมผ่านเครือข่ายสองมิติ จะไม่สามารถสำเร็จได้ถ้าไม่ได้
ได้รับความช่วยเหลือจาก บุคคลจากหลายๆฝ่าย ซึ่งประกอบไปด้วยอาจารย์พงศ์พันธ์ กิจสนาโยธิน
อาจารย์ที่ปรึกษา ซึ่งคอยให้คำปรึกษาที่ดีเพื่อการพัฒนาโปรแกรมให้มีประสิทธิภาพที่ดีขึ้น บริษัท
Startup Design & Network ที่ให้คำปรึกษาและให้เวลาในการพัฒนาโปรแกรม คุณ พรพจน์ จิตแจ้ง
ที่คอยให้คำปรึกษาทางด้านกราฟฟิก รวมถึงจนถึงการรวบรวมข้อมูลต่างๆโดยให้ความเป็นกันเอง
และให้ความร่วมมือเป็นอย่างดี นอกจากนี้ยังมีกำลังใจและคำปรึกษาที่ได้รับจาก ครอบครัว และ
เพื่อนวิศวกรรมคอมพิวเตอร์ ซึ่งสิ่งต่างๆนี้คอยเป็นแรงผลักดันให้คณะผู้จัดทำมีกำลังใจในการทำ
ให้โครงการผ่านไปได้ด้วยดี

สุดท้ายนี้ต้องขอขอบคุณต่อผู้ที่มีส่วนช่วยเหลือคณะผู้จัดทำในทุกๆด้าน ซึ่งรวมไปถึง
บุคคลที่ไม่ได้กล่าวถึงมา ณ ที่นี้ด้วย

นายกฤษณะ

ศรีไกรรส

นายธงชัย

ภูประภากร



สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของ โครงการ	1
1.2 วัตถุประสงค์ของ โครงการ	1
1.3 ขอบเขตของ โครงการ	1
1.4 ขั้นตอนการดำเนินงาน	2
1.6 ผลที่คาดว่าจะได้รับ	2
1.7 งบประมาณที่ใช้ในการดำเนินงาน	2
บทที่ 2 หลักการและทฤษฎี	
2.1 โพรโตคอลที่ซีพี/ไอพี(TCP/IP).....	3
2.2 อินเทอร์เน็ต แอดเดรส (Internet Address).....	6
2.3 การเก็บข้อมูล และการส่งข้อมูล.....	7
2.4 หมายเลขประจำตัวของ โพรโตคอล(Port Number).....	7
2.5 มาโครมีเดีย แฟลชเอ็มเอ็ม (Macromedia FlashMX).....	8
2.6 หลักการสร้างระบบผู้ใช้หลายคน ด้วยโปรแกรม Macromedia FlashMX.....	9
2.7 ซ็อกเก็ต(Socket).....	11
2.8 XML(eXtensible Markup Language).....	13
2.9 XML Socket(XML : eXtensible Markup Language).....	14
2.10 Macromedia FlashMX และ XML Socket	14
2.11 จาวา(Java).....	17
2.12 การทำงานฝั่งเครื่องแม่ข่าย(Server).....	18

สารบัญ(ต่อ)

	หน้า
บทที่ 3 วิธีการดำเนินงาน	
3.1 การออกแบบเกมในเครื่องลูกข่าย.....	20
3.2 การทำงานของโปรแกรมประยุกต์ฝั่งเครื่องแม่ข่าย (Server).....	21
3.3 การทำงานของ โปรแกรมประยุกต์ฝั่งเครื่องลูกข่าย (Client).....	24
บทที่ 4 ผลการดำเนินงาน	
4.1 การทดสอบการทำงาน	27
4.2 วิธีการทดสอบการทำงาน	27
4.3 ผลการดำเนินงาน.....	29
บทที่ 5 บทสรุป	
5.1 บทสรุป	37
5.2 ข้อเสนอแนะ	37
เอกสารอ้างอิง	38
ภาคผนวก	39
ประวัติผู้เขียน	46

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงช่วงของ ไอพี แอดเดรสของแต่ละคลาส.....	6



สารบัญรูป

รูปที่	หน้า	
2.1	ชั้นที่ซีพี/ไอพี(TCP/ IP Layer).....	3
2.2	ชั้นของโพรโทคอลในชุดของ ทีซีพี/ไอพี.....	5
2.3	แสดงการส่งข้อมูลในโมเดลของทีซีพี/ไอพี.....	7
2.4	หน้าต่างโปรแกรม Macromedia FlashMX.....	8
2.5	การติดต่อระหว่างลูกข่ายและแม่ข่ายแบบผู้ใช้หลายคน.....	10
2.6	การติดต่อระหว่าง ลูกข่ายและแม่ข่ายในการเรียกข้อมูลเว็บผ่านทางบราวเซอร์.....	11
2.7	ติดต่อระหว่าง ลูกข่ายและแม่ข่ายผ่าน ซีอกเก็ต.....	12
2.8	การกระจายข้อมูลจาก แม่ข่าย ไปให้ลูกข่ายแต่ละตัว.....	12
2.9	ขั้นตอนการติดต่อระหว่างลูกข่ายและแม่ข่ายผ่าน XML Socket.....	14
2.10	การเขียนคำสั่งการเชื่อมต่อเครื่องแม่ข่าย โดยใช้ฟังก์ชัน XML Socket.....	17
3.1	แผนภาพแสดงหน้าต่างเกม KILLER.....	19
3.2	แผนภาพแสดงขั้นตอนการทำงานของเกมKILLER.....	20
3.3	แผนภาพแสดงขั้นตอนการทำงานของเครื่องแม่ข่าย.....	22
3.4	แสดงการเริ่มต้นของเกม.....	23
3.5	แผนภาพแสดงขั้นตอนการทำงานของเครื่องลูกข่าย.....	25
4.1	แสดงการเปิด โปรแกรมประยุกต์ฝั่งแม่ข่าย.....	27
4.2	แสดงหน้าต่างในการกรอกชื่อ.....	28
4.3	แสดงหน้าต่างในการรอผู้เล่น.....	29
4.4	แสดงหน้าต่างในการเล่นเกม.....	30
4.5	แสดงการเดินของผู้เล่น.....	31
4.6	แสดงการวางระเบิด.....	32
4.7	แสดงการระเบิด.....	33
4.8	แสดงไอเท็มที่ซ่อนอยู่.....	34
4.9	แสดงการวางระเบิดเมื่อเก็บไอเท็มระเบิดได้.....	35
4.10	แสดงการถูกไฟของตัวละคร.....	36

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ปัจจุบันการใช้งานออนไลน์เป็นแบบ ชุมชน (Community) หรือ ผู้ใช้หลายคน(Multi User) มากขึ้น เพราะทำให้ผู้ใช้มีส่วนร่วมบนอินเทอร์เน็ต(Internet) หรือเว็บไซต์(web site) กับอีกฝ่ายซึ่งเป็นคนจริงๆไม่ใช่เป็นเพียงเครื่องจักรเหมือนที่ผ่านมา จะเห็นว่ามีการพัฒนาการใช้งานเพื่อให้สามารถติดต่อกันได้ตั้งแต่ไปรษณีย์อิเล็กทรอนิกส์เรื่อยมาจนถึง โปรแกรมประเภท IRC (Internet Relay Chat) ซึ่งทำให้ลักษณะของการสื่อสารเป็นไปแบบรับและส่งข้อมูลในเวลาเดียวกัน (Real-Time System) และสำหรับเกมเองก็เช่นกัน ลักษณะของเกมจากที่มีผู้เล่นสองคนที่ต้องนั่งเล่นที่เครื่องคอมพิวเตอร์เดียวกัน ปัจจุบันจะมีเกมประเภทผู้เล่นหลายคน (Multi Player) ที่สามารถให้ผู้เล่นหลายๆคนที่อยู่ต่างสถานที่กันเข้ามาเล่นเกมด้วยกันผ่านระบบเครือข่ายได้

เกมผ่านเครือข่ายจึงเป็นอีกทางเลือกหนึ่งที่ผู้ใช้สามารถที่จะเข้ามาเล่นเกมด้วยกันได้ โดยผ่านทางระบบเครือข่าย เกมที่จะเล่นนั้นสามารถเล่นผ่านทาง โปรแกรมค้นดู (Browser) ได้เลย โดยที่เครื่องลูกข่าย (Client) ไม่ต้องติดตั้งเกมลงในเครื่อง ทำให้มีความสะดวก รวดเร็วในการเล่นผู้ใช้ บริการสามารถเลือกเพลงฟังขณะเล่นเกมได้ เพื่อเป็นการผ่อนคลาย และให้ความบันเทิงแก่ผู้ใช้ บริการได้มากยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงการ

1. พัฒนาโปรแกรมประยุกต์ทางด้านเครือข่ายของเกมผ่านเครือข่าย
2. นำเสนอแนวคิด โปรแกรมประยุกต์แบบเครือข่าย และหลักการติดต่อสื่อสารแบบซ็อกเก็ต (Socket) เพื่อเป็นแนวทางในการติดต่อสื่อสารแบบอื่นๆต่อไป

1.3 ขอบเขตของโครงการ

1. ทำงานในระบบปฏิบัติการวินโดวส์ (Windows)
2. ผู้เล่นสามารถเล่นเกมพร้อมกัน 4 คนได้ผ่านเครือข่ายได้
3. ผู้เล่นสามารถเล่นเกมพร้อมกัน 2 คนได้ในเครื่องเดียวกันได้
4. ผู้เล่นสามารถเลือกฟังเพลงขณะเล่นเกมได้
5. ผู้เล่นสามารถสนทนาผ่านเครือข่ายได้

1.4 ขั้นตอนการดำเนินงาน

กิจกรรม	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	
ศึกษาและรวบรวมข้อมูล	←			→								
ออกแบบโปรแกรม				←			→					
พัฒนาโปรแกรม								←		→		
พัฒนาโปรแกรม									←		→	
สรุปผล											←	→

1.5 ผลที่คาดว่าจะได้รับ

1. ได้เรียนรู้เกี่ยวกับการเขียน โปรแกรมภาษา JAVA และระบบเครือข่าย
2. ได้เรียนรู้ถึงการถ่ายภาพเคลื่อนไหวและการเขียนโปรแกรมต่างๆ ของโปรแกรมมาร์โครมีเดีย แฟลชเอ็มเอ็กซ์ (Macromedia FlashMX)

1.6 งบประมาณที่ต้องใช้

1. ค่าเอกสารและหนังสือ 800 บาท
 2. ค่าวัสดุอุปกรณ์ 1,000 บาท
 3. ค่าจัดทำรูปเล่มโครงการ 200 บาท
- รวมค่าใช้จ่าย 2,000 บาท

บทที่ 2

หลักการและทฤษฎี

ส่วนของหลักการและทฤษฎีที่เกี่ยวข้องในการพัฒนาเกมผ่านเครือข่ายโดยใช้โปรแกรม Macromedia FlashMX เป็นแบบจำลอง(Application) และใช้ ภาษา JAVA ในการสร้างเป็นตัวแม่ข่าย(Server) มีความจำเป็นที่ต้องศึกษาและทำความเข้าใจเกี่ยวกับกลไกการทำงานและคำสั่งของ Macromedia FlashMX และการทำงานของระบบเครือข่าย

2.1 โพรโทคอลที่ซีพี/ไอพี (TCP/IP)

TCP(Transmission Control Protocol) มีหน้าที่ในการตรวจสอบการรับส่งข้อมูลระหว่างคอมพิวเตอร์ผู้รับและผู้ส่งให้ได้รับข้อมูลถูกต้อง ส่วนไอพี มีหน้าที่ ในการเลือกเส้นทางที่ใช้รับส่งข้อมูล ผ่านระบบเครือข่าย และตรวจสอบที่ แอดเดรสของผู้รับซึ่งเรียกว่า ไอพีแอดเดรส (IP Address)

โพรโทคอล ที่ซีพี/ไอพี(TCP/IP) ได้รับการออกแบบให้เป็นอิสระ จากชนิดของคอมพิวเตอร์ ฮาร์ดแวร์และระบบปฏิบัติการ การทำงานของโพรโทคอลมีความเชื่อถือได้สูงและทำงานได้แม้ในบางสภาวะที่มีการสื่อสารมีความผิดปกติ รวมทั้งความสามารถในการเลือกเส้นทางในการส่งข้อมูลได้ โพรโทคอล ที่ซีพี/ไอพี ไม่ได้มีเพียงสองโพรโทคอลดังที่ได้กล่าวมา แต่ ทีซีพี/ไอพี เป็นกลุ่มของโพรโทคอลที่นำมาจัดรวมกันไว้ เรียกว่าเป็นชุดโพรโทคอลที่ซีพี/ไอพี

2.1.1 การแบ่งชั้น (Layering)

ทีซีพี/ไอพี (TCP/IP) เป็นชุดของโพรโทคอล ที่ประกอบด้วยโพรโทคอลย่อยหลายตัวแต่ละตัวก็ทำหน้าที่ ในแต่ละชั้น ซึ่งรับผิดชอบและแปลความหมายของข้อมูล ในแต่ละระดับของการสื่อสาร ซึ่งโดยภาพรวมแล้วทีซีพี/ไอพี แบ่งออกเป็น 4 ชั้น ดังนี้

Application
Transport
Network
Link

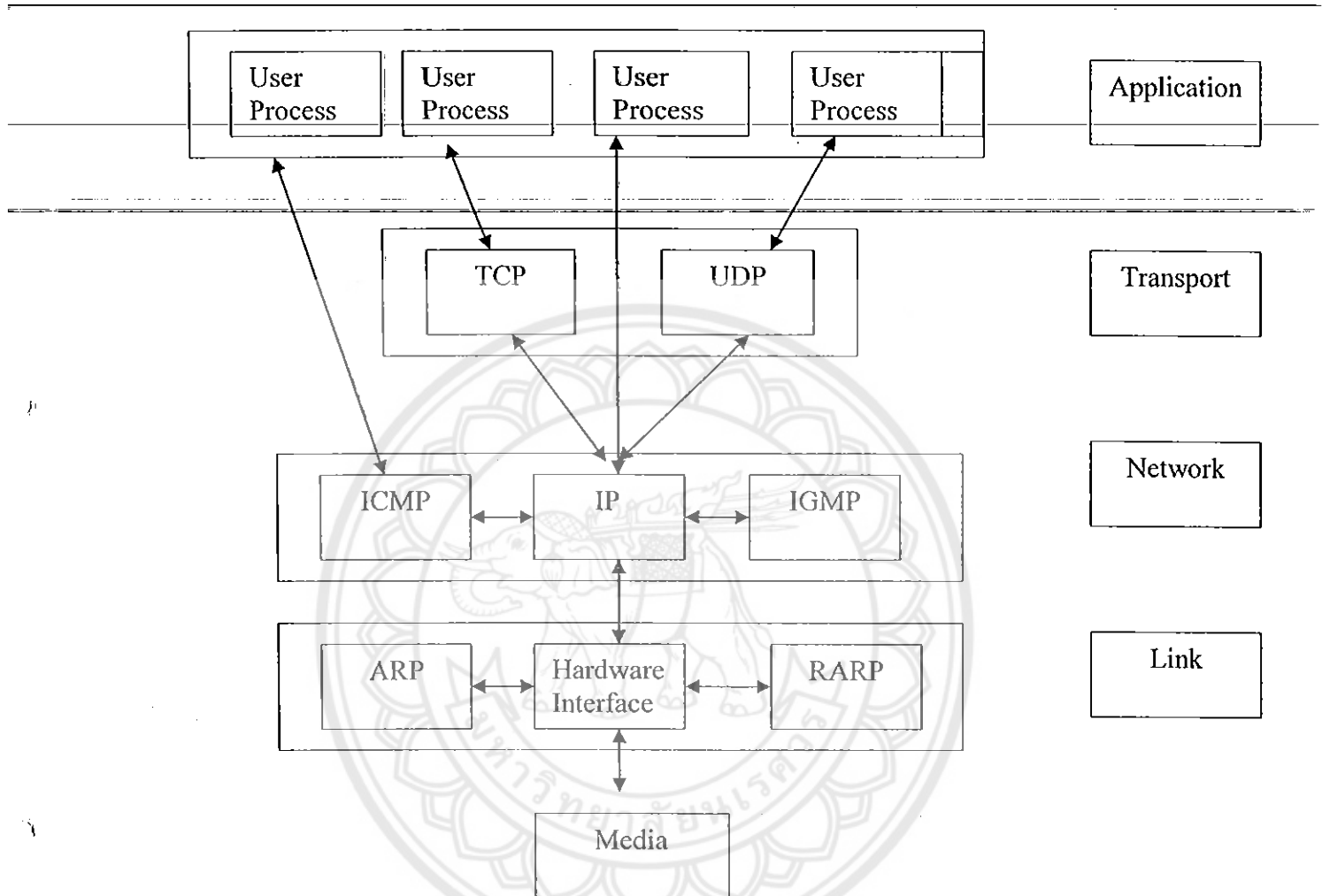
รูปที่ 2.1 ชั้นที่ซีพี/ไอพี(TCP/IP Layer)

2.1.2 หน้าที่ของแต่ละชั้นมี ดังนี้

- **ชั้นลิงก์ (Link Layer)** ในชั้นนี้จะเป็นดีไวซ์ไดรเวอร์(Device Driver) ที่ทำงานอยู่บนระบบปฏิบัติการแต่ละระบบ ทำหน้าที่รับผิดชอบในการรับส่งข้อมูลตั้งแต่ ระดับกายภาพ สัญญาณไฟฟ้า จนถึงการแปลความหมายจากระดับแรงดันสัญญาณ ไฟฟ้าเป็นข้อมูลของคอมพิวเตอร์ โพรโตคอลระดับนี้ เช่น อีเทอร์เน็ต(Ethernet)และเอสแอลไอพี(Serial Line Internet Protocol)
- **ชั้นเครือข่าย(Network Layer)** รับผิดชอบในการรับ-ส่งข้อมูลในเน็ตเวิร์คส่งต่อข้อมูลไปยัง จุดหมายปลายทาง โพรโตคอลระดับนี้ได้แก่ โพรโตคอลไอพี (ไอพี) ,โพรโตคอลไอซีเอ็มพี (ICMP)โพรโตคอลไอจีเอ็มพี (IGMP)
- **ชั้นทรานสปอร์ต(Transport Layer)** รับผิดชอบในการรับส่งข้อมูลระหว่างเครื่องหนึ่ง ไปยังอีกเครื่องหนึ่ง และจะส่งข้อมูลไปให้ชั้นของแอปพลิเคชัน(Application Layer) นำไปใช้งานต่อ โพรโตคอลระดับนี้ได้แก่ โพรโตคอลทีซีพี (TCP),โพรโตคอล (UDP)
- **ชั้นแอปพลิเคชัน(Application Layer)** เป็นชั้นที่แอปพลิเคชันเรียกโพรโตคอลระดับล่างๆ ลงไป เพื่อวัตถุประสงค์แตกต่างกัน เช่น โพรโตคอลเอฟทีพี(FTP :File Transfer Protocol)ใช้สำหรับรับส่งข้อมูลระหว่างคอมพิวเตอร์โพรโตคอลเอสเอ็มทีพี(SMTP :Simple Mail Transfer Protocol)ใช้สำหรับรับส่งจดหมายอิเล็กทรอนิกส์โพรโตคอลเทลเน็ต (Telnet)ใช้สำหรับควบคุมเครื่องระยะไกลโพรโตคอลเอชทีทีพี (HTTP :Hypertext Transfer Protocol)เป็นโพรโตคอลที่ใช้รับส่งข้อมูลเว็บเพจระหว่างบราวเซอร์และ เว็บเซิร์ฟเวอร์

2.1.3 ระดับชั้นที่สี่ (TCP Layer)

ระดับของ TCP Layer แบ่งออกเป็นระดับดังรูปที่



รูปที่ 2.2 ชั้นของโปรโตคอลในชุดของ ทีซีพี/ไอพี

2.1.3.1 โปรโตคอลทีซีพี(TCP) อยู่ในชั้นขนส่ง ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูลให้มีเสถียรภาพและเชื่อถือได้

2.1.3.2 โปรโตคอลยูดีพี(UDP) อยู่ในชั้นขนส่งทำหน้าที่จัดการและควบคุมการรับส่งข้อมูลเช่นเดียวกันแต่ไม่มีกลไกการรับส่งที่มีเสถียรภาพ และเชื่อถือได้ โดยปล่อยหน้าที่นี้ให้กับแอปพลิเคชันเป็นผู้ทำหน้าที่นี้แทน

2.1.3.3 โปรโตคอลไอพี(ไอพี) อยู่ในชั้นเครือข่าย เป็นโปรโตคอลหลักในการสื่อสารข้อมูลซึ่งกลไกสำคัญที่ทำให้ข้อมูลสามารถเคลื่อนที่ไปยังปลายทางได้ ก็คือโปรโตคอลไอพีนี้เอง

2.1.3.4 โปรโตคอลไอซีเอ็มพี(ICMP) อยู่ในชั้นเครือข่าย ทำหน้าที่เสริมการทำงานของไอพี

(Internet Control Message Protocol) ให้สมบูรณ์ โดยจะเป็นโปรโตคอลที่คอยส่งข่าวสารและแจ้งความผิดพลาดให้แก่ไอพี

2.1.3.5 โพรโตคอลไอจีเอ็มพี(IGMP) อยู่ในชั้นเครือข่าย ทำหน้าที่ในการส่ง(Inter Group Management Protocol) ยูดีพีดาต้าแกรม(UDP Datagram)ไปยังกลุ่มของคอมพิวเตอร์ในเครือข่ายหลายๆตัวพร้อมกัน

2.1.3.6 โพรโตคอลเออาร์พี(ARP) อยู่ในลิงค์ชั้น ทำหน้าที่เปลี่ยน ไอพี แอดเดรสที่ใช้ (Address Reservatio no Protocol) โดยให้เป็นแอดเดรสของเน็ตเวิร์ค อินเทอร์เฟซ (NetworkInterface)

2.1.3.7 โพรโตคอลอาร์เออาร์พี(RARP) อยู่ในชั้นเชื่อมต่อ ทำหน้าที่กลับกันกับเออาร์พี (Reverse ARP) คือ เปลี่ยนระหว่างแอดเดรสของตัวเชื่อมเครือข่าย(Network Interface) ให้เป็นแอดเดรสที่ใช้ โดยอินเทอร์เน็ตแอดเดรส(IP)

2.2 อินเทอร์เน็ต แอดเดรส (Internet Address)

ทุกหน้าที่ต่ออยู่บนอินเทอร์เน็ต จะต้องมีหมายเลขประจำตัวเพื่อใช้ในการสื่อสารข้อมูล เรียกว่าอินเทอร์เน็ต แอดเดรส (Internet Address) หรือเรียกย่อๆว่า ไอพี แอดเดรส (IP Address) โดยค่า ไอพี แอดเดรสนี้ จะเป็นหมายเลขจำนวน 32 บิตแต่แทนที่จะกำหนดให้เลขทั้ง 32 บิตนั้น ถูกนับต่อเนื่องกันไป วิธีการแบ่งหมายเลขดังกล่าวออกเป็นกลุ่มของเลขขนาด 8 บิต จำนวน 4 ชุด และค้นแต่ละชุดด้วยจุด เช่น 192.168.68.20

นอกจากนี้ ในไอพี แอดเดรส ยังถูกแบ่งออกเป็น 2 ส่วนคือ ส่วนที่เป็นแอดเดรสของเน็ตเวิร์ค (Network ID) และส่วนที่เป็นแอดเดรสของโฮสต์ (Host ID) ซึ่งข้อมูลดังกล่าวจะถูกใช้สำหรับค้นหาเส้นทางของไอพีในการส่งข้อมูลจากต้นทางไปจนถึงปลายทางได้อย่างถูกต้อง ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงช่วงของ ไอพี แอดเดรสของแต่ละคลาส

Class	Range
A	0.0.0.0-127.255.255.255
B	128.0.0.0-191.255.255.255
C	192.0.0.0-223.255.255.255
D	224.0.0.0-239.255.255.255
E	240.0.0.0-255.255.255.255

2.3 การเก็บข้อมูล และการส่งข้อมูล

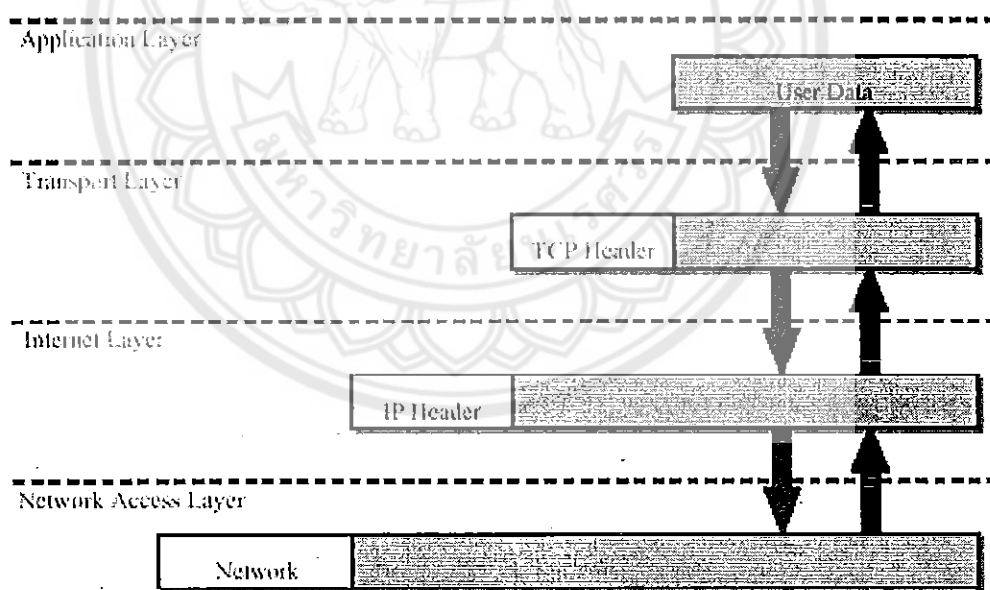
ในการรับส่งข้อมูลนั้น ข้อมูลที่รับส่งกันจริงๆ บนเน็ตเวิร์กนั้นจะประกอบด้วย 2 ส่วนคือ ข้อมูลจริงกับข้อมูลของโพรโตคอล ข้อมูลของโพรโตคอลเรียกว่า เฮดเดอร์(Header) เรียกส่วนที่มีข้อมูลจริงกับข้อมูลของโพรโตคอลว่าแพ็คเกจ เปรียบเสมือนการส่งจดหมายซึ่งจะต้องประกอบไปด้วยเนื้อความของจดหมายและซองจดหมายที่เขียนชื่อที่อยู่ ติดแสตมป์ จะเปรียบเหมือนข้อมูลที่ใช้ในการรับส่งข้อมูลของโพรโตคอลนั้น โพรโตคอลก็จะใส่ 1 ซอง ถ้าข้อมูลต้องส่งผ่านหลายชั้นจำนวนซองก็จะถูกใส่เพิ่มหลายชั้นตามลำดับ ดังรูปที่ 2.4 ดังนั้นถ้าส่งข้อมูลผ่านโพรโตคอลที่ซีพีข้อมูลก็จะมีการทำงานตามลำดับดังนี้

ลำดับที่ 1 ซองของทีซีพี(TCP)

ลำดับที่ 2 ซองไอพี(ไอพี)

ลำดับที่ 3 ซองอีเทอร์เน็ต(Ethernet)

และฝ่ายที่รับข้อมูลก็ต้องแกะซองออกตามลำดับ โดยจะต้องแกะซองของอีเทอร์เน็ต(Ethernet) ก่อน แล้วจะเจอซองของไอพี(ไอพี)แกะซองของไอพี แล้วก็เจอซองของทีซีพี(TCP) และในลำดับสุดท้ายก็เจอข้อมูลที่ต้องการตามลำดับ



รูปที่ 2.3 แสดงการส่งข้อมูลในโมเดลของทีซีพี/ไอพี

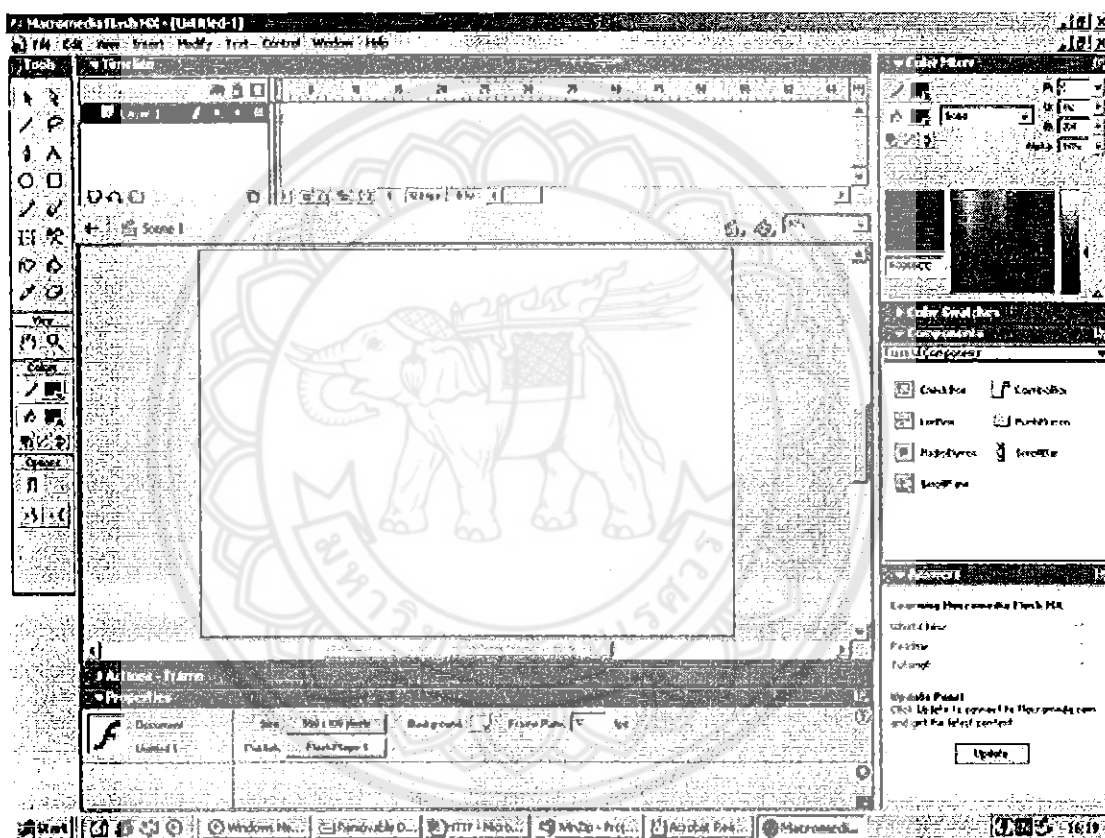
2.4 หมายเลขประจำตัวของโพรโตคอล(Port Number)

ในโพรโตคอลที่ซีพี/ไอพีจะมีการกำหนดพอร์ตอยู่ในเฮดเดอร์ของแพ็คเกจ เพื่อระบุว่าข้อมูลส่วนนี้เป็นของแอปพลิเคชันอะไร เช่น พอร์ต 20,21 เป็นของ เอฟทีพี(FTP),23 เป็นของ เทลเน็ต (Telnet) ,พอร์ต 80 เป็นของเอชทีทีพี (HTTP) เป็นต้น

ในชุดโปรโตคอลที่ซีพี/ไอพี มีโปรโตคอลหลักที่ขอกว่าถึง 5 โปรโตคอลได้แก่โปรโตคอลที่ซีพี โปรโตคอลยูดีพี โปรโตคอลไอซีเอ็มพี โปรโตคอลไอพี และโปรโตคอลเออาร์พีซึ่งการทำงานของแต่ละโปรโตคอลมีรายละเอียด ดังนี้

2.5 มาร์โครมีเดีย แฟลชเอ็มเอ็กซ์ (Macromedia FlashMX)

เมื่อเปิดใช้งานโปรแกรม Macromedia FlashMX จะมีหน้าต่างการทำงานของโปรแกรม ดังรูปที่ 2.1



รูปที่ 2.4 หน้าต่างโปรแกรม Macromedia FlashMX

Macromedia FlashMX เป็นโปรแกรมที่เน้นการทำภาพเคลื่อนไหว(Animation)และเชิงโต้ตอบ(Interactive)เพื่อใช้งานบนเว็บเพจ(Webpage)แต่ก็สามารถทำเป็นโปรแกรมเพื่อทำงานแบบมาตรฐานเดียว(Standard alone)ในเครื่อง PC(Personal Computer) ได้ด้วย ยกตัวอย่าง เช่น ภาพยนตร์ หนึ่งเรื่อง การเคลื่อนไหวในภาพยนตร์ที่เราได้เห็นนั้นมืองค์ประกอบหลักสำคัญมาจากการนำภาพนิ่งแต่ละภาพมาทำเป็นเฟรม(Frame)หนึ่งเฟรมแล้วเอาเฟรมเหล่านั้นมาเรียงต่อกัน จากนั้นก็นำมาแสดงผลโดยการฉายอย่างรวดเร็วจากเฟรมที่หนึ่งไปเฟรมที่สองและจากเฟรมที่สองต่อไปยังเฟรมที่สามต่อไป (โดยปกติสำหรับภาพยนตร์จะใช้ยี่สิบสี่เฟรมต่อวินาที)

2.5.1 ประโยชน์ของโปรแกรม Marcromedia FlashMX

2.5.1.1. **มีขนาดไฟล์เล็กกว่าไฟล์ทั่วไป** สาเหตุสำคัญที่ทำให้ไฟล์มัลติมีเดียที่สร้างโดยโปรแกรม Marcromedia FlashMX มีขนาดเล็กมากเมื่อเทียบกับไฟล์มัลติมีเดีย(Multimedia)ประเภทอื่น นั้นเป็นเพราะรูปแบบการเก็บรายละเอียดของรูปเป็นแบบลายเส้นนั่นคือเป็นไฟล์แบบ เวกเตอร์ (Vector) ในขณะที่โปรแกรมอื่นๆจะบันทึกรายละเอียดของรูปทุกจุดภาพ(Pixel) ซึ่งลักษณะภาพที่เหมาะสมสำหรับใช้โปรแกรม Marcromedia FlashMX ควรเป็นภาพแนวการ์ตูน(Cartoon)ที่เน้นลายเส้นและใช้สีที่ค่อนข้างสม่ำเสมอ แต่ภาพในลักษณะอื่นก็สามารถใช้ได้เช่นกัน

2.5.1.2 **มีความสามารถในการโต้ตอบกับผู้ใช้** เป็นลักษณะเฉพาะของโปรแกรม Marcromedia FlashMX ที่ผู้เล่นสามารถที่จะมีการโต้ตอบกับ Marcromedia FlashMX ที่เราได้สร้างขึ้นมา เช่น เมื่อทำการคลิก(Click)ที่ภาพ ภาพนั้นมีการแสดงโต้ตอบกลับมาหาเรา

2.5.2 หลักการทำงานของโปรแกรม Marcromedia FlashMX

เราสามารถแบ่งขั้นตอนการทำงานเป็น 4 ส่วนด้วยกัน

2.5.2.1 เป็นการนำภาพที่สร้างขึ้น หรือที่เราได้มาจากที่อื่น(Import) นำมาวางบนระยะ (Stage)

2.5.2.2 ทำการแปลงวัตถุนั้นเป็นสัญลักษณ์(Symbol) ซึ่งเป็นลักษณะที่ Marcromedia FlashMX สามารถนำไปทำภาพเคลื่อนไหวได้

2.5.2.3 เป็นขั้นตอนที่เราจะมาทำสัญลักษณ์ให้เป็นภาพเคลื่อนไหวซึ่งในขั้นตอนนี้เราจะสามารถใส่เสียงเพื่อเพิ่มความน่าสนใจ ทั้งยังสามารถที่จะใส่คำสั่งการทำงาน(Action Script)เพิ่มเติม เพื่อเพิ่มความสามารถอื่นได้อีกด้วย

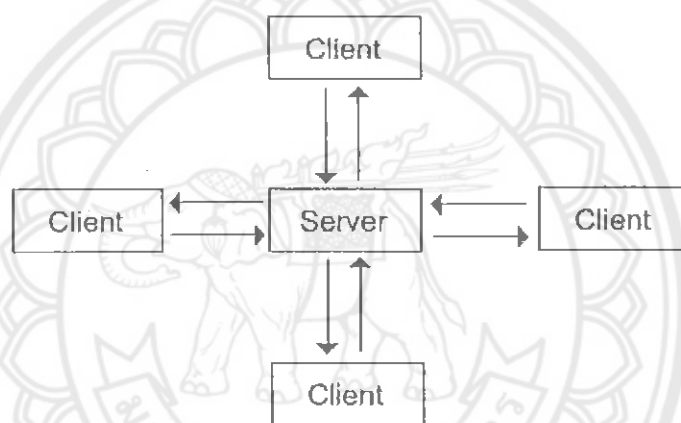
2.5.2.4 เมื่อเราได้สร้างชิ้นงานเสร็จแล้ว เราก็ทำการแปลงชิ้นงานของเราออกมา

2.6 หลักการสร้างระบบผู้ใช้หลายคน ด้วยโปรแกรม Marcromedia FlashMX

การใช้โปรแกรม Marcromedia FlashMX ในการสร้างเกมผ่านเครือข่ายแบบผู้เล่นหลายคน (Multi Player) โดยอาศัยความสามารถทางการสื่อสารกับเครื่องแม่ข่ายที่มีมาให้กับ โปรแกรม Marcromedia FlashMX ซึ่งจะมีวิธีที่จะสร้างระบบผู้ใช้หลายคนนี้ด้วยกัน 2 วิธี คือ การส่งข้อมูลหลายมิติ(HTTP – Based Multi User; HTTP :Hyper Text Transport Protocol)และ XML Socket (XML :Extensible Markup Language)

2.6.1 การส่งข้อมูลหลายมิติ (HTTP – Based Multi User)

ลักษณะของการใช้ Marcromedia FlashMX ติดต่อกับเครื่องแม่ข่ายในแบบ HTTP-Based Multi User นี้จะเหมือนกับการใช้รูปแบบ HTML(Hyper Text Markup Language) ติดต่อกับการทำงานฝั่งเครื่องแม่ข่าย(Server Side Script) เช่นเดียวกับการสร้างรูปแบบ Marcromedia FlashMX ซึ่งในลักษณะการทำงานแบบผู้ใช้หลายคนนี้ ฝั่งลูกข่ายจะมีด้วยกันหลายเครื่อง ซึ่งจะต้องติดต่อกับเครื่องแม่ข่าย ตัวเดียวกันในเวลาพร้อมๆกัน การทำงานระหว่างลูกข่ายแต่ละตัวกับเครื่องแม่ข่ายก็จะมี การขอข้อมูลและตอบรับข้อมูลไปกลับตลอดทุกๆวินาทีตลอดเวลา ดังรูปที่ 2.6 โดยอาศัยการส่งงาน ให้มีการรับและส่งบน Marcromedia FlashMX ไปติดต่อกับโปรแกรมประเภท CGI (Common Gateway Interface) บนฝั่งเครื่องแม่ข่าย



รูปที่ 2.5 การติดต่อระหว่างลูกข่ายและแม่ข่ายแบบผู้ใช้หลายคน

หลักการทำงานในลักษณะนี้นับ Marcromedia FlashMX ก็คือการใช้คำสั่ง loadVariables ในการรับและส่งข้อมูลผ่านทาง วิธีการรับส่งข้อมูลแบบ HTTP(HTTP GET/POST protocol)นั่นเอง โดยวิธีการก็ยังเป็น ไปในลักษณะเช่นเดียวกับการ ใช้รูปแบบแฟลชฟอร์ม (Flash Form)คือการส่งค่าตัวแปรบน Marcromedia FlashMX ไปให้กับการทำงานฝั่งเครื่องแม่ข่ายโดยคำสั่งเหล่านี้ก็จะมองเห็นค่าดั่งแปรในตัวแปรชื่อเดียวกัน ในขณะที่เดียวกันคำสั่งเหล่านั้นก็จะกลับมาในรูปแบบ

PHP : echo "variable1=value1&variable2=value2&variable3=value3";

ASP : Response.write (" "variable1=value1 &variable2=value2&variable3=value3")

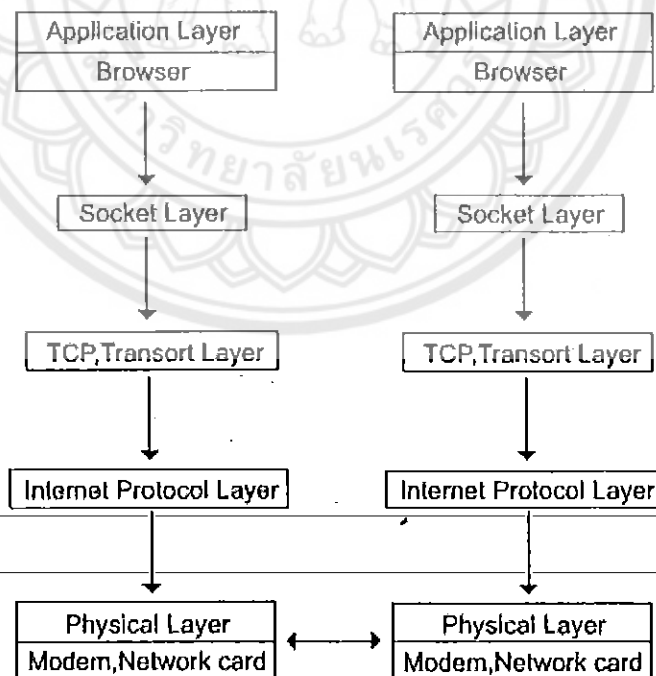
ฝั่งแม่ข่ายก็ต้องทำหน้าที่คอยจัดการกับผู้ใช้แต่ละคนที่ส่งข้อมูลเข้ามา แต่การทำงานในลักษณะนี้ จะมีความเร็วต่ำ เพราะลูกข่ายหรือผู้ใช้แต่ละคนก็จะทำหน้าที่ในการรับส่งข้อมูลกับแม่ข่ายแยกกัน ไป การทำงานด้วยวิธีนี้จึงไม่เป็น ไปในลักษณะแบบการรับและส่งข้อมูลในเวลาเดียวกัน(Realtime)

ดังนั้นหากเราต้องการสร้างเกมหรือระบบที่ต้องการให้มีการแลกเปลี่ยนข้อมูลที่เป็นไปแบบการรับและส่งข้อมูลในเวลาเดียวกันก็ต้องใช้วิธีการที่ Marcromedia FlashMX ได้เตรียมไว้คือ XML Socket ซึ่งเป็นฟังก์ชันหนึ่งของโปรแกรม Marcromedia FlashMX

ข้อดีของวิธีการสร้างระบบผู้ใช้หลายคนแบบ HTTP Based นี้ก็คือ เราสามารถเขียนการทำงานฝั่งแม่ข่ายด้วยคำสั่ง เช่น PHP หรือ ASP ซึ่งการทำงานจะไม่ซับซ้อนไปกว่าวิธีการแบบ XML Socket และยังสามารถหาแม่ข่ายที่จะให้ทดลองได้ง่ายกว่า แต่ก็มีขีดความสามารถอยู่ระดับหนึ่งเหมาะที่จะไปพัฒนาให้กับระบบที่ไม่ต้องการการทำงานแบบการรับและส่งข้อมูลในเวลาเดียวกันมากกว่า ในส่วนต่อไปเราก็จะมาทำความเข้าใจถึงการทำงานของ Socket และการใช้ XML Socket สำหรับสร้างระบบ ผู้ใช้หลายคนบน Marcromedia FlashMX ขึ้นมา

2.7 ซ็อกเก็ต(Socket)

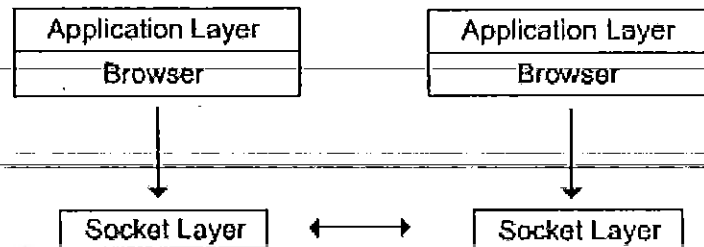
ซ็อกเก็ตเป็นช่องทางหนึ่งที่ทำให้เกิดการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์สองเครื่องได้ การเรียกดูเว็บผ่านทางบราวเซอร์โดยปกติมีการติดต่อระหว่างลูกข่ายและแม่ข่ายก็จริงแต่ไม่ได้ใช้ ซ็อกเก็ตในการติดต่อถึงกัน โดยตรงการทำงานโดยปกติในลักษณะนี้จะเป็นไปตามรูปที่ 2.7



รูปที่ 2.6 การติดต่อระหว่าง ลูกข่ายและแม่ข่ายในการเรียกข้อมูลเว็บผ่านทางบราวเซอร์

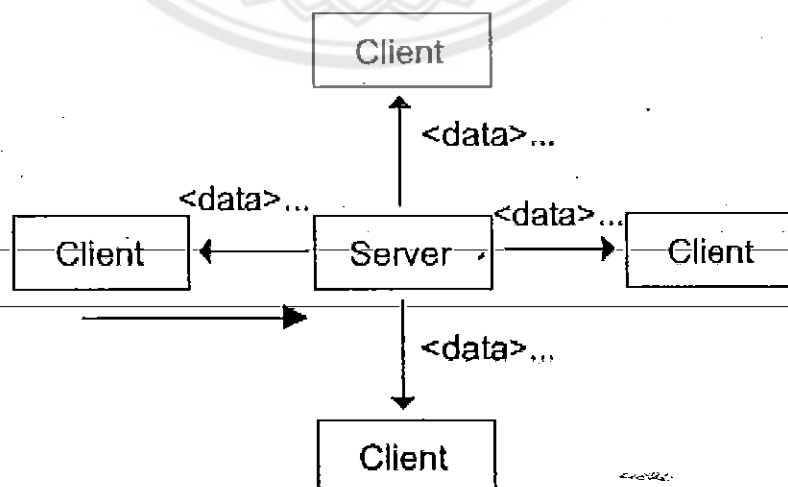
แต่การทำงานของ XML Socket บน Marcromedia FlashMX จะทำให้ลูกข่ายและแม่ข่ายติดต่อกันผ่านทางซ็อกเก็ตเลเยอร์(Socket layer) โดยตรง โดยรูปแบบที่ออกแบบไว้ XML Socket

บน Macromedia FlashMX ได้เตรียมไว้ให้แล้ว ซึ่งจะทำให้การทำงานเป็นไปได้เร็วกว่าแบบ HTTP Based ลักษณะของโปรแกรมติดต่อสื่อสารอื่นๆ เช่น IRC หรือพวเกมหลายผู้เล่น (Multi Player Game) ก็ใช้การทำงานแบบซ็อกเก็ตนี้เช่นกัน ดังรูปที่ 2.8



รูปที่ 2.7 การติดต่อระหว่าง ลูกข่ายและแม่ข่ายผ่าน ซ็อกเก็ต

แต่การทำงานในลักษณะนี้ก็ยังคงประกอบด้วย โปรแกรมฝั่งลูกข่ายซึ่งก็คือคำสั่งการทำงาน และ โปรแกรมฝั่งแม่ข่ายซึ่งก็มีอยู่หลายภาษาที่มีความสามารถในการทำงานกับ ซ็อกเก็ตเช่น Java , Visual Basic เป็นต้น การทำงานในลักษณะนี้ โปรแกรมฝั่งแม่ข่ายจะถูกสั่งให้ทำงานอยู่ตลอดเวลา อยู่แล้ว เพื่อตรวจสอบการขอติดต่อจากผู้ใช้และการปิดการติดต่อ ซึ่งจะต่างจากแบบ HTTP Based ที่การทำงานจะเกิดขึ้นก็ต่อเมื่อมีการเรียกใช้ชื่อไฟล์คำสั่ง (Script) ตัวนั้นเป็นครั้งๆ ไป และการส่ง ข้อมูลจากแม่ข่ายกลับไปให้ลูกข่ายแต่ละเครื่องนั้นก็ทำงานผ่านทางซ็อกเก็ต ที่ตัวแม่ข่ายจะทำการกระจายข้อมูล (Broadcast) ไปยังลูกข่ายทุกเครื่องพร้อมๆ กันเมื่อมีข้อมูลใหม่เกิดขึ้นมา ซึ่งต่าง กับวิธี HTTP Based ที่คำสั่งการทำงานจะต้องคอยเรียกคำสั่งทุกๆ วินาทีเพื่อการขอข้อมูลใหม่ดังรูป ที่ 2.9



รูปที่ 2.8 การกระจายข้อมูลจาก แม่ข่ายไปให้ลูกข่ายแต่ละตัว

2.8 XML(eXtensible Markup Language)

XML นั้นย่อมาจากคำว่า eXtensible Markup Language ซึ่งเป็นภาษาที่ใช้กำหนดภาษา หรือที่เรียกว่า Metadata ก็คือข้อมูลที่บอกรายละเอียดของข้อมูล ซึ่งจะใช้สำหรับกำหนดรูปแบบของลำดับคำสั่งต่างๆ XML นั้นได้ถูกพัฒนามาจาก SGML (Standard Generalized Markup Language) ซึ่งเป็นมาตรฐานสำหรับการกำหนดการแสดงผลเอกสาร สำหรับการแสดงผลบนอุปกรณ์ชนิดต่างๆ ซึ่ง HTML นั้นก็เป็นส่วนหนึ่งของ SGML และคำสั่งหรือ ชุดข้อมูล (Tag) ที่ใช้ในเอกสาร จะถูกกำหนดมาตรฐาน DTD (Document Type Definition) ซึ่งเป็นภาษาที่ใช้สำหรับกำหนดคุณสมบัติต่างๆ ของ Tag ได้ เช่น พารามิเตอร์ของ Tag เป็นต้น การใช้งาน XML นั้น จำเป็นต้องใช้ร่วมกับ สไตลชีท (Style Sheet) หรือมาตรฐานอื่นๆ เพราะ XML เพียงแต่กำหนดรูปแบบของ Tag แต่ไม่ได้กำหนดว่า Tag ใดจะแสดงผลแบบใด เพราะเมื่อเอาข้อมูลในรูปแบบ XML ไปแสดงผลในอุปกรณ์ชนิดใด ก็จะต้องใช้วิธีแสดงผลของอุปกรณ์นั้น เช่น ใช้มาตรฐาน SMIL (Synchronized MultiMedia Integration Language) สำหรับข้อมูลมัลติมีเดีย หรือใช้ XSL (eXtensible Stylesheet Language) สำหรับการแสดงผลในโปรแกรมค้นดู (Browser) นอกจากนี้ XML ยังสนับสนุนตัวอักษรภาษานานาชาติ โดยใช้มาตรฐาน ISO 10646 โดยจุดมุ่งหมายของภาษา XML นั้นก็คือภาษาเรียบง่าย

XML เราสามารถที่จะกำหนดชื่อให้กับ Tag เป็นชื่ออะไรก็ได้ ซึ่งจะใช้สำหรับการเก็บข้อมูล เพื่อให้ชื่อ Tag บ่งบอกถึงข้อมูลที่อยู่ภายใน Tag นั้นๆ ว่าเป็นค่าอะไร เช่น

```
<USERNAME>XSAINT</USERNAME>
```

จะเป็นการเขียน XML เพื่อเก็บค่า USERNAME โดยใช้ Tag ชื่อ USERNAME ที่เก็บค่า XSAINT เอาไว้ เป็นต้น หรือ

```
<USERNAME>XSAINT</USERNAME>
```

```
<MESSAGE>Hello XML Socket</MESSAGE>
```

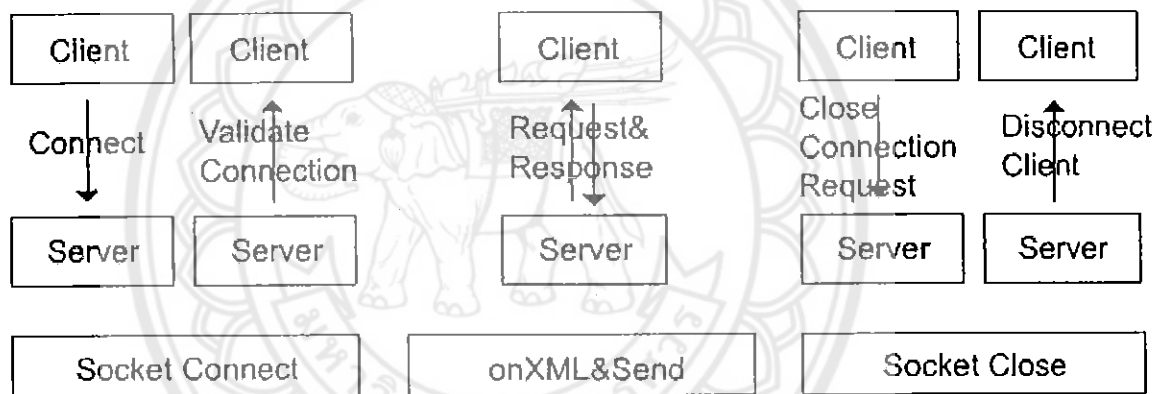
ก็จะเพิ่มชุดข้อความ (Tag Message) เข้ามาเพื่อเก็บค่าข้อความที่ USER ชื่อ XSAINT ได้ส่งออกไป สำหรับการใช้งาน XML Socket บน Macromedia FlashMX ก็คือเราจะส่งข้อมูลไปยัง โปรแกรมฝั่งแม่ข่าย โดยมีรูปแบบของข้อมูลอยู่ในรูปแบบ XML และเมื่อข้อมูลนี้ไปยังแม่ข่ายแล้ว แม่ข่ายก็จะกระจายข้อมูลที่อยู่ในรูป XML ไปให้กับลูกข่ายแต่ละตัว การทำงานที่ลูกข่ายแต่ละตัวก็จะนำ ออบเจ็กต์ XML มาถอดข้อมูลที่อยู่ในรูป XML นี้ว่า ค่าภายในมีค่าเป็นอะไรเพื่อเก็บลงตัวแปรและนำไปใช้งานต่อไป

2.9 XML Socket(XML : eXtensible Markup Language)

XML Socket เป็นออบเจ็กต์(object)มาตรฐานที่เตรียมมาให้ใน Macromedia Flash ตั้งแต่เวอร์ชัน 5 ขึ้นไปเพื่อให้ผู้พัฒนาสามารถใช้ Macromedia Flash ในการติดต่อระหว่างลูกข่ายและแม่ข่าย โดยผ่านซ็อกเก็ต โดยมีรูปแบบของข้อมูลอยู่ในรูป XML ได้

2.10 Macromedia FlashMX และ XML Socket

การทำงานระหว่างลูกข่ายกับแม่ข่ายแต่ละคู่ผ่านทาง ซ็อกเก็ตโดยอาศัยออบเจ็กต์ XML ซ็อกเก็ตลักษณะเหมือนกันคือ เริ่มจากลูกข่ายขอติดต่อกับแม่ข่าย เมื่อการติดต่อ(connect)ผ่านทางซ็อกเก็ตสำเร็จก็จะอยู่ในขั้นตอนที่มีการรับส่งข้อมูลกันและเมื่อเครื่องลูกข่ายเลิกใช้ก็จะเป็ขั้นตอน การปิด ซ็อกเก็ตสำหรับติดต่อกัน ดังรูปที่ 2.6



รูปที่ 2.9 ขั้นตอนการติดต่อกันระหว่างลูกข่ายและแม่ข่ายผ่าน XML Socket

ต่อจากนี้ไปจะอธิบายการทำงานของออบเจ็กต์ XML Socket ที่อยู่ในแต่ละขั้นตอน เพื่อใช้สำหรับเตรียมสร้างเกมหลายผู้เล่นที่จะอาศัยการติดต่อผ่านทาง XML Socket ก่อนอื่นต้องสร้างออบเจ็กต์ของคลาส(Class) XML Socket สำหรับใช้งานขึ้นมาก่อน โดยผ่านทาง คอนสตรัคเตอร์ (Constructor) XML Socket ซึ่งจะอยู่ในรูปแบบ

```
mySocket = new XMLSocket ();
```

ซึ่งจะได้ mySocket เป็นชื่อของออบเจ็กต์สำหรับใช้งาน คลาส XML Socket ต่อไป

2.10.1 ขั้นตอนการติดต่อกับแม่ข่าย (Connection)

การขอติดต่อกับแม่ข่ายจะใช้เมธอด `connect` ซึ่งจะมรูปแบบคือ

```
mySocket . connect(host , port)
```

โฮสต์(host) เป็น IP(Internet Protocol) หรือชื่อแม่ข่าย(Host Name) ของแม่ข่าย ที่ต้องการขอติดต่อไป โดยหากระบุเป็นค่าว่าง(null) หมายถึง host เป็นตัวเดียวกันกับที่ไฟล์ swf อยู่

พอร์ต(port) เป็นเลข TCP(Transmission Control Protocol) พอร์ต ที่ระบุเป็นช่องทางการติดต่อระหว่างลูกข่ายและแม่ข่ายซึ่งจะต้องมีค่าตรงกันระหว่างภายในคำสั่งการทำงานและฝั่งแม่ข่ายโดยจะต้องมีค่าตั้งแต่ 1024 ขึ้นไป

เมธอด `connect` นี้จะส่งค่ากลับมาเป็น true หากการขอติดต่อสำเร็จ และจะส่งค่ากลับมาเป็น false หากการขอติดต่อไม่สำเร็จ โดยเมธอดเปิดการติดต่อจะมาทำหน้าที่รับช่วงต่อว่าจะให้ผลเป็นอย่างไร

```
mySocket . onConnect(success) ;
```

`success` จะเป็นค่าที่ถูกส่งกลับมาจากเมธอด `connect` การนำเมธอดนี้ไปใช้งานจะต้องกำหนดฟังก์ชันใดๆสำหรับตอบรับกับค่า `success` นี้ขึ้นมา โดยการประกาศ

```
mySocket . onConnect = handleConnect ต่อไป
```

2.10.2 ขั้นตอนการรับ – ส่งข้อมูลระหว่างลูกข่ายและแม่ข่าย

ออบเจ็กต์ `XMLSocket` จะมีเมธอดสำหรับรับข้อมูลจากแม่ข่ายคือ `onXML` โดยมีรูปแบบ

```
mySocket . onXML (XMLObject);
```

`XMLObject` เป็นออบเจ็กต์ที่มีการเก็บข้อมูลในรูปแบบ XML ซึ่งส่งมาจากแม่ข่าย โดยการทำงานจะเป็นไปในลักษณะเดียวกับ `onConnect` คือจะต้องสร้างฟังก์ชันขึ้นมาควบคู่ไปกับเมธอดนี้ด้วย โดยกำหนดให้ส่วนต้นเสียก่อนเป็น

```
mySocket . onXML = handleIncoming;
```


แล้วกำหนดการจัดการกับ XMLObject ภายในฟังก์ชันชื่อ handleIncoming ต่อไป โดยการทำงานภายในฟังก์ชันนี้จะเป็นการส่งเกี่ยวกับออบเจ็กต์ XML ของ Macromedia FlashMX ซึ่งจะมีเมธอดต่างๆอีกมากมาย การส่งข้อมูลเราใช้เมธอด send โดยเรียกใช้เมธอดนี้เมื่อต้องการที่จะส่งข้อมูลไปให้เครื่องแม่ข่าย ซึ่งจะต่างจากการรับซึ่งจะมีการรอ handle อยู่ตลอดเวลา รูปแบบของการใช้เมธอด send คือ

```
mySocket . send(XMLObject);
```

โดยที่ XMLObject ก็จะเป็นออบเจ็กต์ XML ซึ่งถูกสร้างขึ้นมาจากตัวสร้าง XML (XML Constructor) และกำหนดรูปแบบ Tag เก็บค่าข้อมูลในรูปแบบ XML เพื่อเตรียมส่งไปให้กับแม่ข่าย

2.10.3 การยกเลิกการติดต่อกับแม่ข่าย (Close connection)

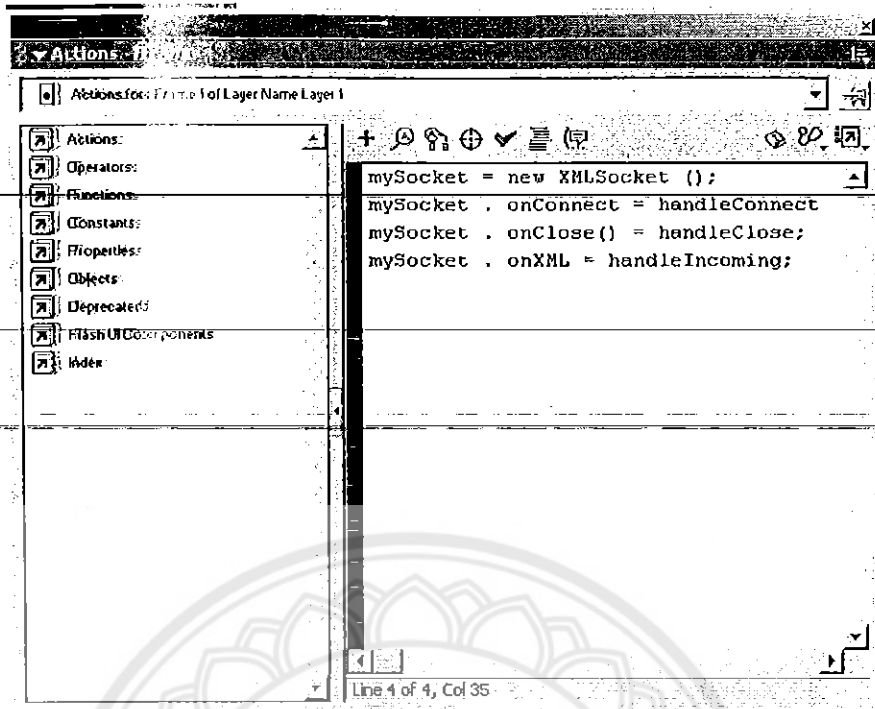
ในการปิดการทำงานหรือต้องการออกจากระบบ (Log out) เราจะใช้เมธอด close สำหรับการปิดการติดต่อผ่านทาง Socket กับแม่ข่าย ซึ่งเรียกใช้ในรูปแบบ

```
mySocket . close();
```

และใช้เมธอด onClose ในการทำงานต่อจากการปิดการติดต่อจะให้ทำงานต่อไปเช่นไร โดยกำหนดฟังก์ชันมา handle เช่นเดียวกับ onConnect

```
mySocket . onClose() = handleClose;
```

แล้วไประบุการทำงานเมื่อปิดการติดต่อเสร็จสิ้นภายในฟังก์ชันชื่อ handleClose การเขียนคำสั่งเพื่อติดต่อกับเครื่องแม่ข่ายนั้นจะเขียนที่เฟรมแรกดังรูปที่ 2.10



รูปที่ 2.10 การเขียนคำสั่งการเชื่อมต่อเครื่องแม่ข่ายโดยใช้ฟังก์ชัน XML Socket

2.11 จาวา(Java)

JAVA เป็นโปรแกรมคอมพิวเตอร์ภาษาหนึ่งพัฒนาโดยบริษัท Sun Microsystem ที่มีลักษณะคล้ายกันกับโปรแกรมภาษา C และ C++(Object Orinted) แต่แตกต่างกันตรงที่ภาษาจาวา ได้ถูกออกแบบมาให้ใช้งานบนระบบปฏิบัติการใด ๆ ก็ได้ ไม่ว่าจะเป็น วินโดวส์,ยูนิก (Windows,UNIX)หรือแม้กระทั่งเครื่องระบบเมนเฟรม(Mainframe)โดยการเขียนโปรแกรมเพียงครั้งเดียวช่วยลดเวลาการทำงานของผู้เขียนโปรแกรมที่ต้องการเขียนโปรแกรมสำหรับใช้งานบนระบบปฏิบัติการหลาย ๆ ระบบ นอกจากนี้ภาษาจาวายังมีความเด่นในการสร้างโปรแกรมมัลติมีเดียได้อย่างง่ายดาย

โปรแกรมที่พัฒนาจากภาษา JAVA มี 2 ลักษณะ คือ JAVA Application และ JAVA Applet โดย JAVA Application คือโปรแกรม application ที่รันได้เหมือนโปรแกรมทั่วไปใน Windows และ OS(Operating System) อื่นๆ แต่มีข้อแม้ว่าในระบบจะต้องมีสิ่งที่เรียกว่า JVM (JAVA Virtual Machine) ก่อนจึงจะเรียกใช้งาน JAVA Application นี้ได้ สำหรับ JAVA Applet เป็นโปรแกรมเล็กๆ ที่จะถูกฝังอยู่ในเว็บเพจ แล้วจะถูกบรรจุลง(download) ตัวโปรแกรมฐานสอง(binary program) ซึ่งก็คือตัวแอฟเพลท(applet) นั้นเอง มาพร้อมๆกัน แล้วนำมาประมวลผล, ทำงาน และแสดงผลอยู่ใน ตัว Browser นั้นเอง ซึ่งก็คล้ายๆกัน Browser ที่จะใช้งาน applet ได้ ก็จะต้องมีสิ่งที่เรียกว่า JVM อยู่ในตัว Browser เองด้วยเช่นกัน เพื่อทำการประมวลผลตัวแอฟเพลท

ปัจจุบันมีโปรแกรมอีกลักษณะหนึ่งที่เรียกว่า JAVA Servlet จะเป็นโปรแกรมที่ทำงานที่ฝั่งเครื่องแม่ข่ายที่ถูกพัฒนาขึ้น โดยใช้ภาษา JAVA

2.12 การทำงานฝั่งเครื่องแม่ข่าย(Server)

การทำงานของฝั่งเครื่องแม่ข่ายจะเป็นหน้าที่ของ โปรแกรม Java ซึ่งจะคอยเปิดซ็อกเก็ตเตรียมรอรับการขอการเชื่อมต่อจากเครื่องลูกข่ายอยู่แล้ว

โปรแกรมของฝั่งเครื่องแม่ข่ายนี้จะประกอบด้วย Java คลาสสองชุดซึ่งจะทำงานควบคู่กัน คือ ฝั่งเครื่องแม่ข่ายและฝั่งเครื่องลูกข่าย โดย เครื่องแม่ข่ายจะเป็นคลาสหลักในการทำงานเมื่อมีการเชื่อมต่อมาจากฝั่งเครื่องลูกข่ายเพื่อจะเข้าสู่เกม ฝั่งเครื่องแม่ข่ายก็จะสร้างออบเจกต์เครื่องลูกข่ายขึ้นมา

เมื่อเริ่มต้นเครื่องแม่ข่ายจะสร้าง ซ็อกเก็ตแล้ววนลูปเพื่อการติดต่อจากเครื่องลูกข่ายและทันทีที่มาติดต่อเข้ามาเครื่องแม่ข่ายจะสร้างออบเจกต์เครื่องลูกข่ายแล้วรับค่าตัวแปร userID ที่ส่งมาจากเครื่องลูกข่ายเพื่อตรวจสอบว่าชื่อนั้น ซ้ำกันกับชื่อผู้เล่นเดิมที่มีอยู่แล้วหรือไม่ ถ้าซ้ำกัน ก็ส่งข้อมูลเป็น

```
<VALIDATE>0</VALIDATE>
```

ซึ่งค่าที่เป็น 0 จะเป็นตัวบอกว่าชื่อที่ใส่มาครั้งแรก ให้เปลี่ยนชื่อใหม่แล้วใส่อีกครั้ง กรณีที่ชื่อไม่ซ้ำกัน จะส่งข้อมูล

```
VALIDATE>1</VALIDATE>
```

ค่าที่ส่งไปจะเป็น 1 จากนั้นจะส่งจำนวนผู้ใช้ที่ติดต่อกอยู่ในขณะนั้นด้วย

```
<NUMCLIENTS>จำนวนผู้ใช้ </NUMCLIENTS>
```

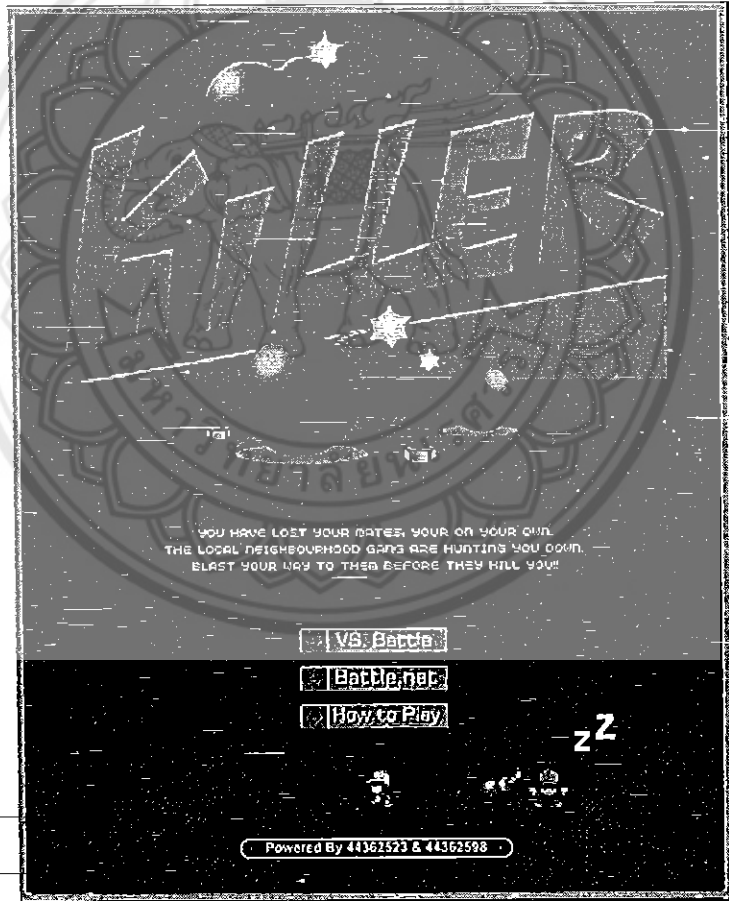
แล้วเริ่มทำงานของออบเจกต์เครื่องลูกข่ายต่อไป

การทำงานของออบเจกต์เครื่องลูกข่ายจะรับส่งข้อมูลที่มาจาก Macromedia FlashMX แล้วส่งไปให้ออบเจกต์ เครื่องแม่ข่ายเพื่อให้ทำการกระจายข้อมูลนี้ โดยส่งให้ผู้เล่นทุกคนที่ติดต่อเข้ามาในเครื่องแม่ข่ายเพื่อให้ในขณะนั้นตลอดการทำงาน จนกว่าผู้ใช้งานจะยกเลิกการติดต่อและลบเครื่องลูกข่ายนั้นออกไป

บทที่ 3

วิธีการดำเนินงาน

ในการสร้างเกมออนไลน์ KILLER แบ่งออกเป็น 2 ส่วน คือ VS Battle และ VS Battle.net ในส่วนของ VS Battle ก็จะสามารถเล่นเกมพร้อมกัน 2 คน บนเครื่องคอมพิวเตอร์เครื่องเดียวกัน โดยใช้คีย์บอร์ด อันเดียวกัน ส่วน VS Battle.net ก็ สามารถเล่นเกมพร้อมกันได้ 4 คน ผ่านเครือข่าย โดยใช้คอมพิวเตอร์ในการเล่นคนละเครื่อง คอมพิวเตอร์จะสามารถรับส่งข้อมูลกันได้โดยผ่านเครื่องแม่ข่าย (Server) ที่เป็นตัวกลางอยู่บนเครื่องคอมพิวเตอร์ โดยใช้วิธีการติดต่อกันแบบซ็อกเก็ต (Socket)



รูปที่ 3.1 แผนภาพแสดงหน้าต่างเกม KILLER

เปิดบริการฝั่งแม่ข่ายเพื่อรอรับการติดต่อ

เครื่องลูกข่ายที่ 1, 2, 3 และ 4 เชื่อมต่อเข้าสู่ระบบ

เครื่องลูกข่ายรับค่าจากเครื่องแม่ข่ายเพื่อนำมาสร้างออปเจ็ค

เครื่องแม่ข่ายส่งค่าเริ่มต้นของเกมไปให้เครื่องลูกข่ายทั้งสอง







เครื่องลูกข่ายรับค่าต่างๆแล้วส่งไปที่เครื่องแม่ข่าย


เครื่องแม่ข่ายส่งค่าไปให้เครื่องลูกข่ายทุกเครื่อง


รูปที่ 3.2 แผนภาพแสดงขั้นตอนการทำงานของเกมKILLER

3.1 การออกแบบเกมในเครื่องลูกข่าย

เกม KILLER เป็นเกมที่มีลักษณะเป็นช่อง (Block) ทำให้การเขียนโปรแกรม ในแผนที่จะมีขนาด 15x15 ช่องแต่ละช่องมีขนาด 32x32 พิกเซลประกอบด้วย ส่วน

- Floor  เป็นส่วนที่ตัวละครสามารถเดินผ่านได้
- Block  เป็นส่วนที่ตัวละครเดินผ่านไม่ได้แต่เมื่อส่วนนี้ถูกระเบิดจะเปลี่ยนเป็นFloor
- Wall  เป็นส่วนที่ตัวละครเดินผ่านไม่ได้ เมื่อถูกระเบิดจะไม่มีเปลี่ยนแปลง
- Bomb  เป็นส่วนที่ตัวละครจะปล่อยออกมาเพื่อทำลายฝ่ายตรงข้าม
- Effect  เป็นส่วนที่เกิดการระเบิดจาก Bombเพื่อทำลายฝ่ายตรงข้าม
- Player  เป็นส่วนที่ผู้เล่นให้บังคับ

Itembomb  เป็นส่วนที่เมื่อผู้เล่นเก็บได้ผู้เล่นสามารถวางระเบิดได้เพิ่มอีกหนึ่งลูก

Itemspeed  เป็นส่วนที่เมื่อผู้เล่นเก็บได้ผู้เล่นสามารถวิ่งได้เร็วขึ้น

การเดินของตัวละคร จะเกิดขึ้นเมื่อลูกข่ายมีการกดปุ่มบนคีย์บอร์ด ซึ่งการเดินของตัวละคร ทั้งสองสามารถเดินซ้อนทับตำแหน่งกันได้ ซึ่งแต่ละปุ่มจะทำหน้าที่แตกต่างกันออกไป ดังนี้

เมื่อกดปุ่ม	↑	เดินขึ้นบน
เมื่อกดปุ่ม	←	เดินทางซ้าย
เมื่อกดปุ่ม	→	เดินทางขวา
เมื่อกดปุ่ม	↓	เดินลงล่าง
เมื่อกดปุ่ม	;	วางระเบิด

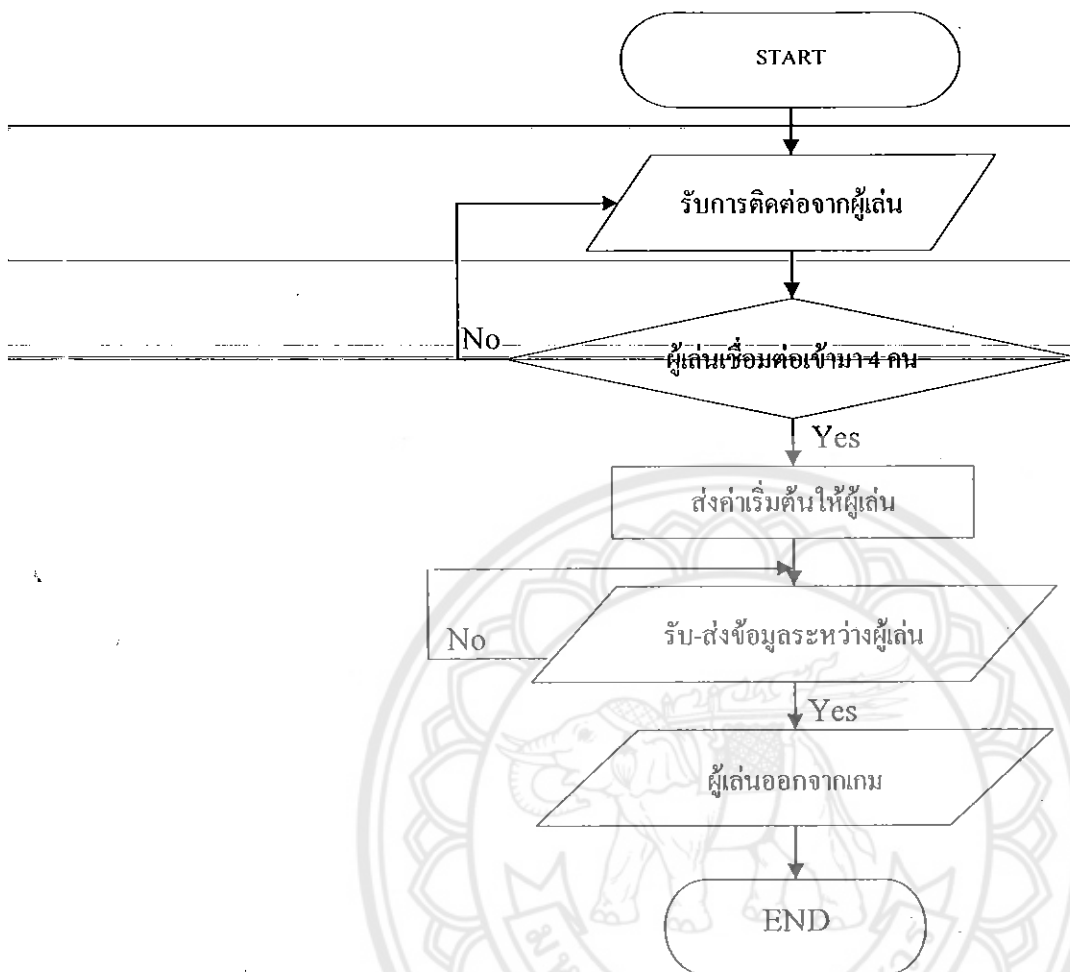
สำหรับการเล่น VS Battle ผู้เล่นคนที่ 2 ปุ่มการเดินของตัวละครคือ

เมื่อกดปุ่ม	W	เดินขึ้นบน
เมื่อกดปุ่ม	D	เดินทางซ้าย
เมื่อกดปุ่ม	A	เดินทางขวา
เมื่อกดปุ่ม	S	เดินลงล่าง
เมื่อกดปุ่ม	I	วางระเบิด

ซึ่งการวางระเบิดนี้เริ่มแรกจะวางได้คนละลูก การระเบิดของลูกระเบิด จะเกิดขึ้นเมื่อการวางระเบิดลูกนั้นผ่านไปประมาณ 3 วินาที การระเบิดนี้ทำให้เกิดไฟในตำแหน่งรอบๆ ของตำแหน่งที่วางระเบิดการจบเกม หรือการตายของตัวละคร จะเกิดขึ้นเมื่อตัวละครฝ่ายใดฝ่ายหนึ่งอยู่ในตำแหน่งเดียวกับไฟที่เกิดจากการระเบิด ก็คือตัวละครเดิน ไปถูกไฟหรือไฟที่เกิดจากการระเบิดถูกตัวละคร ซึ่งในทุกๆวินาทีเครื่องลูกข่ายจะส่งข้อมูลของตัวเองไปให้เครื่องแม่ข่ายไม่ว่าผู้เล่นยังไม่ได้กดปุ่มใดๆเลยเพื่อให้ตำแหน่งของผู้เล่นทุกเครื่องลูกข่ายเหมือนกัน

3.2 การทำงานของโปรแกรมประยุกต์ฝั่งเครื่องแม่ข่าย (Server)

รูปแบบของ โปรแกรมประยุกต์ซึ่งเป็นเกมออนไลน์ที่เขียนด้วย Macromedia FlashMX ใช้การติดต่อสื่อสารกันแบบซ็อกเก็ต และข้อมูลที่ส่งผ่านอยู่ในรูป XML โดยที่โปรแกรมประยุกต์นี้แบ่งออกเป็น 2 ส่วน คือ ส่วนของแม่ข่าย (Server.java) และลูกข่าย(Client.java)



รูปที่ 3.3 แผนภาพแสดงขั้นตอนการทำงานของเครื่องแม่ข่าย

เกม KILLER จะเริ่มจากเครื่องแม่ข่ายเปิดบริการไว้ก่อน แล้วมีเครื่องลูกข่ายที่เป็นผู้เล่นเชื่อมต่อเข้ามา โดยเริ่มแรกผู้เล่นจะต้องเข้าเล่นเกม KILLER ในเมนู VS Battle.net แล้วกรอกชื่อเพื่อเข้าเล่นเกมหลังจากที่ผู้เล่นครบ 4 คนเกมจะเริ่มโดยอัตโนมัติ เมื่อเข้าเล่นเกมเครื่องแม่ข่ายจะส่งค่าเริ่มต้นมาให้เครื่องลูกข่าย ค่าที่ส่งมานั้นจะเป็นลักษณะ สตริง แล้วเครื่องลูกข่ายจะนำไปทำให้อยู่ในรูป อาร์เรย์ ขนาด 15x15 ซึ่งค่าในอาร์เรย์จะแตกต่างกันออกไป

คือ 1, 2 และ 3 เพื่อสร้างอุปเจ็คในตารางทางฝั่งผู้เล่น ค่าต่างๆ เหล่านี้จะถูกเก็บไว้ในตัวแปร $map[i][j]$ การกำหนดเริ่มต้นที่เป็นตาราง มีวิธีการกำหนดค่าดังนี้

- ค่า $map[i][j] = 3$ แสดงรูป Wall เป็นส่วนที่แสดงขึ้นตายตัวทุกๆ เกม เครื่องแม่ข่ายจะกำหนดตำแหน่งที่

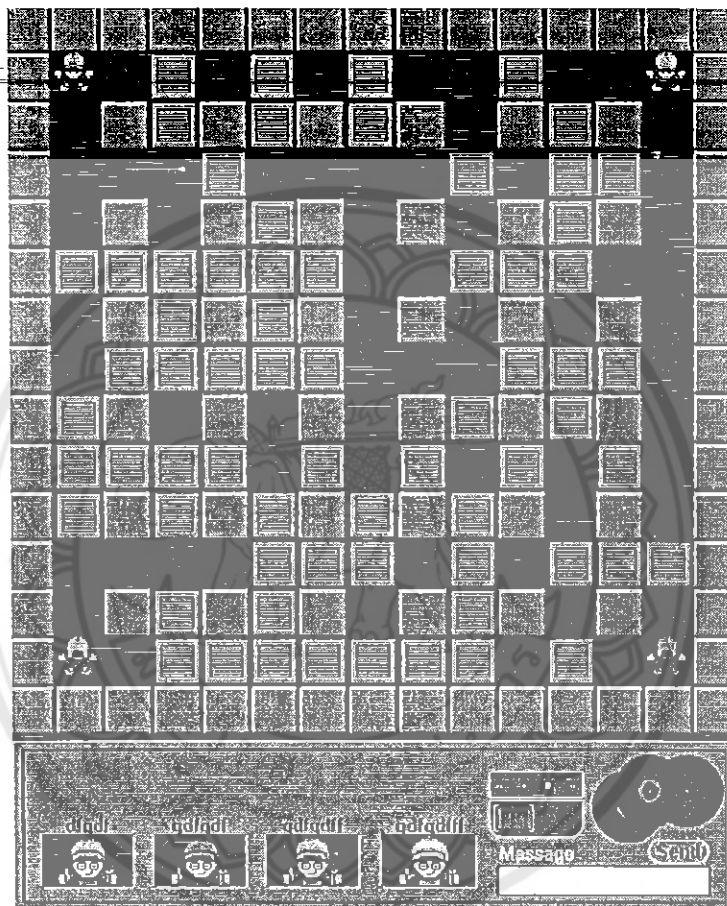
(2,2),(2,4),(2,6),(2,8),(2,10),(2,12),(4,2),(4,4),(4,6),(4,8),(4,10),(4,12),(6,2),(6,4),(6,6),(6,8),(6,10),(6,12),(8,2),(8,4),(8,6),(8,8),(8,10),(8,12),(10,2),(10,4),(10,6),(10,8),(10,10),(10,12),(12,2),(12,4),(12,6),(12,8),(12,10),(12,12) มีค่าเท่ากับ 3

- ค่า $map[i][j] = 1$ แสดงรูป Floor และ $map[i][j] = 2$ แสดงรูป Block ค่าทั้งสองจะเกิดจากการสุ่ม (Random) ขึ้นมา และเก็บอยู่ในตำแหน่งที่นอกเหนือจากส่วนที่เป็น Wall

- ค่า item ที่เครื่องแม่ข่ายส่งมานั้นจะทำการสุ่มตาม Block ที่มีอยู่ซึ่งแบ่งเป็น itemomb12

อื่นและ itemspeed 4 อื่น

ส่วนตัวผู้เล่นนั้นเครื่องลูกข่ายแต่ละเครื่องจะสร้างด้วยตัวเองโดยแบ่งแยกตามลำดับผู้เล่น



รูปที่ 3.4 แสดงการเริ่มต้นของเกม

การเดินทางของตัวละครนั้นเมื่อผู้เล่นกดปุ่มเดินเครื่องลูกข่ายนั้นจะส่งค่าตำแหน่ง $x-y$ ของตัวผู้เล่นของตัวเองไปให้ผู้เล่นคนอื่นโดยผ่านเครื่องแม่ข่าย เครื่องแม่ข่ายจะส่งไปให้เครื่องลูกข่ายทุกเครื่องเพื่อบอกตำแหน่งของผู้เล่นนั้น

การวางระเบิด การทำงานและการรับ-ส่งค่าต่างๆ จะคล้ายกับการเดินทางของตัวละคร เมื่อมีผู้เล่นฝ่ายใดฝ่ายหนึ่งกดปุ่ม ; เพื่อวางระเบิด เครื่องลูกข่ายนั้นก็ส่งข้อมูลตำแหน่งของระเบิดให้เครื่องแม่ข่ายแล้วกระจายไปให้เครื่องลูกข่ายทั้งหมดก่อนที่ระเบิดจะระเบิดเป็นไฟจะมีการหน่วงเวลาไว้ประมาณ 3 วินาที

เครื่องแม่ข่ายจะเก็บตำแหน่งของระเบิดไว้ เมื่อเวลาผ่านไปประมาณ 3 วินาที ระเบิดจะเกิดการระเบิดและเปลี่ยนเป็นรูปไฟแทนตำแหน่งของระเบิด รวมทั้งในตำแหน่งรอบๆ (บน, ล่าง, ซ้าย และขวา) ของระเบิด ถ้าตำแหน่งนั้นเป็นเดิมเป็นรูป Floor หรือ Block

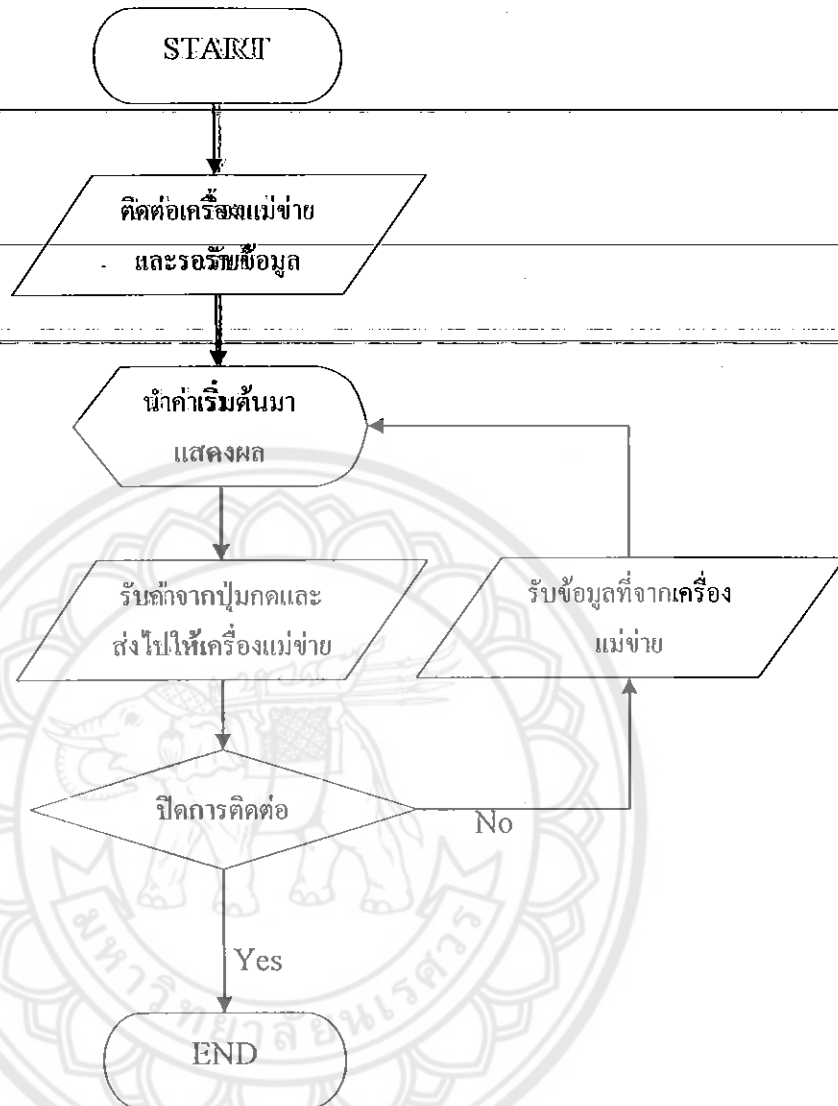
- ถ้าเป็นรูป Floor จะเป็นเหมือนเดิม
- ถ้าเป็นรูป Block จะเปลี่ยนเป็นรูป Floor

การจมน้ำหรือการตายของผู้เล่น จะเกิดขึ้นเมื่อผู้เล่นเดินเข้าไปในตำแหน่งที่มีไฟอยู่หรือเมื่อเกิดระเบิดแล้วไฟที่เกิดขึ้น เกิดในตำแหน่งเดียวกับที่ผู้เล่นยืนอยู่

3.3 การทำงานของโปรแกรมประยุกต์ฝั่งเครื่องลูกข่าย (Client)

การทำงานของโปรแกรมประยุกต์ทางฝั่งเครื่องลูกข่ายที่เป็น Macromedia FlashMX เมื่อเชื่อมต่อกับเครื่องแม่ข่ายเข้าไปอยู่ในระบบออนไลน์แล้ว จะมีหน้าที่รับค่าตัวแปรต่างๆ จากเครื่องแม่ข่ายแล้วนำค่าเหล่านั้นมาทำการสร้างออปเจ็คและรับค่าจากการคดปุ่มที่ใช้ในการเดินของตัวละคร





รูปที่ 3.5 แผนภาพแสดงขั้นตอนการทำงานของเครื่องลูกข่าย

โปรแกรมประยุกต์ทางฝั่งเครื่องลูกข่ายจะมีการสร้างช่องรับข้อความที่เป็น TextField ใช้รับชื่อของผู้เล่นก่อนเริ่มเกม

ในการสร้างออปเจ็ทของเครื่องลูกข่าย-ค่าที่รับมาจากเครื่องแม่ข่ายจะเป็นสตริงซึ่งเครื่องลูกข่ายจะนำมาทำการแปลงเป็นข้อมูลอาร์เรย์ที่มีขนาด 15×15 คือ มีค่าตั้งแต่ $(0, 0)$ ถึง $(14, 14)$ ซึ่งออปเจ็ทบางตัวจะใส่สคริปไว้เพื่อให้สามารถประมวลผลได้เลยและแต่ละออปเจ็ทมีขนาด 32×32 พิกเซลและในฝั่งเครื่องลูกข่ายนี้จะประกอบฟังก์ชันในการรับส่งข้อมูลคือฟังก์ชันส่งข้อมูล `sendMessage` และ ฟังก์ชันรับข้อมูล `handleIncoming`

ฟังก์ชัน sendmessage จะแบ่งออกเป็นหลายโหมคซึ่งประกอบด้วย

- getname โหมค รับชื่อผู้เล่นอื่นๆเมื่อเข้าเกม
- chat โหมค เมื่อผู้เล่นมีการส่งข้อความหากัน
- move โหมค เมื่อผู้เล่นมีการกดปุ่มเพื่อเดิน
- bom โหมค เมื่อผู้เล่นมีการวางระเบิด
- item โหมค เมื่อผู้เล่นเก็บไอเทมได้
- die โหมค เมื่อผู้เล่นนั้นตาย

ฟังก์ชัน handleIncoming จะแบ่งออกตาม โหมคที่เครื่องแม่ข่ายส่งมาว่าอยู่โหมค โหมคแล้วจะนำข้อมูลนั้นมาประมวลผลตามโหมคนั้นๆ



บทที่ 4

ผลการดำเนินงาน

การทำงานของเกม **KILLER** เริ่มต้นด้วยการเปิดโปรแกรมประยุกต์ทางฝั่งแม่ข่าย (Server) ไว้ที่คอมพิวเตอร์เครื่องหนึ่ง จากนั้นก็เปิดโปรแกรมประยุกต์ของฝั่งเครื่องลูกข่าย (Client) ที่เขียนด้วยโปรแกรม **Marcromedia FlashMX** แล้วเข้าไปติดต่อกับเครื่องแม่ข่าย เมื่อเครื่องลูกข่ายครบ 4 เครื่องแล้วก็จะทำการเริ่มเกม

4.1 การทดสอบการทำงาน

การทดสอบการทำงานของเกม **KILLER** ที่สร้างขึ้นนี้ เป็นการตรวจสอบและแก้ไขข้อผิดพลาดต่างๆ ที่เกิดขึ้นจากสาเหตุเขียน โปรแกรมทั้งทางฝั่งแม่ข่ายและลูกข่าย เพื่อให้โปรแกรมประยุกต์มีความถูกต้องตามที่ออกแบบไว้มากที่สุด

4.2 วิธีการทดสอบการทำงาน

ในการทดสอบได้เปิดโปรแกรมประยุกต์ฝั่งแม่ข่ายไว้ในเครื่องคอมพิวเตอร์ที่มี IP Address ตาม IP ของเครื่องที่เปิด และช่องส่งข้อมูล (Port) 8000 เครื่องแม่ข่ายจะเริ่มให้บริการโดยรอการติดต่อเข้ามาจากผู้เล่นทั้ง 4 จากนั้น เมื่อผู้เล่นติดต่อเข้าไปเพื่อร้องขอการเล่นเกม โดยต้องกรอกชื่อ ก่อนแล้วจอภาพของผู้เล่นคนที่ 1 จะเปลี่ยนเป็นหน้าจอซึ่งสามารถพิมพ์ข้อความสนทนาได้จะต้องรอให้มีผู้เล่นเชื่อมต่อเข้ามาให้ครบ เมื่อมีผู้เล่นครบ 4 คนแล้ว เครื่องแม่ข่ายจะสร้างเกมขึ้นมาแล้วส่งค่าต่างๆ ที่ใช้ในการเล่น ไปให้ผู้เล่นทั้ง 4 คน



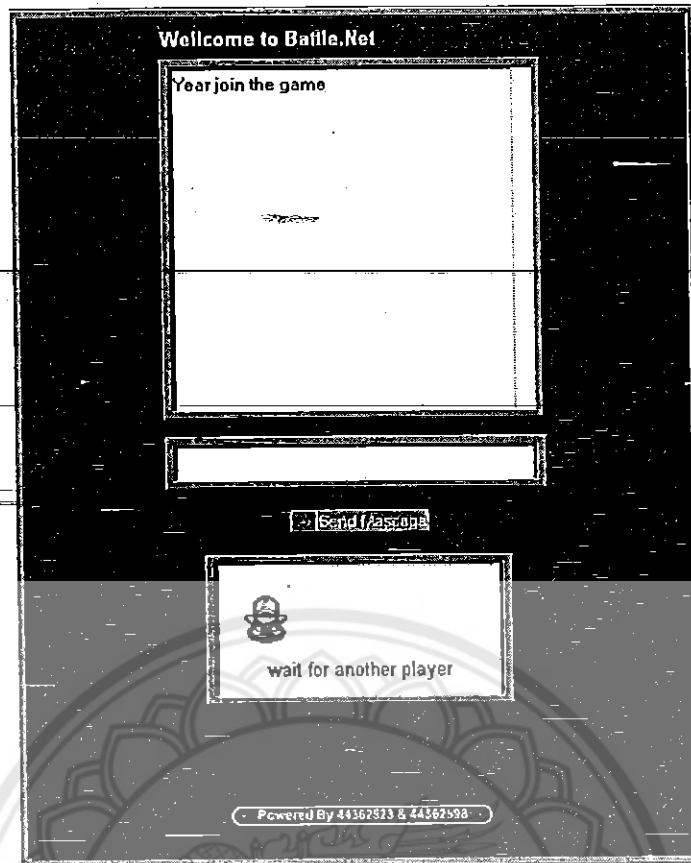
```
C:\WINDOWS\System32\cmd.exe
C:\Documents and Settings\Administrator\Desktop>cd
C:\>cd j2sdk1.4.2_07
C:\j2sdk1.4.2_07>cd bin
C:\j2sdk1.4.2_07\bin>java Server 8000
[4/11/2546 13:56:54] Attempting to Start Server
[4/11/2546 13:56:54] Server Started on Port: 8000
```

รูปที่ 4.1 แสดงการเปิดโปรแกรมประยุกต์ฝั่งแม่ข่าย

เมื่อใส่ชื่อเรียบร้อยแล้ว ผู้เล่นสามารถเริ่มเล่นเกมได้โดยเลือกที่ปุ่ม Connect ในการรอเริ่มต้นของเกม เครื่องแม่ข่ายจะรอให้มีผู้เล่นเชื่อมต่อเข้ามา 4 ผู้เล่นก่อนจึงจะส่งค่าเริ่มต้นของเกมไปให้ผู้เล่น



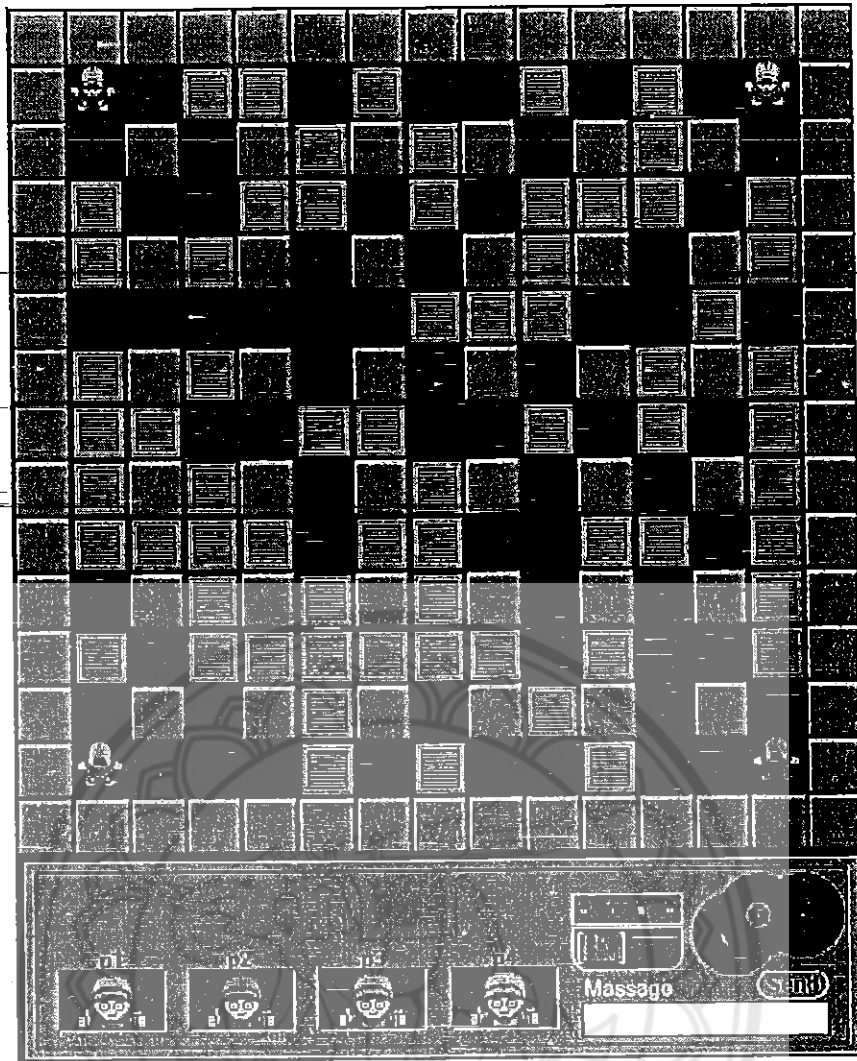
รูปที่ 4.2 แสดงหน้าต่างในการกรอกชื่อ



รูปที่ 4.3 แสดงหน้าต่างในการรอผู้เล่น

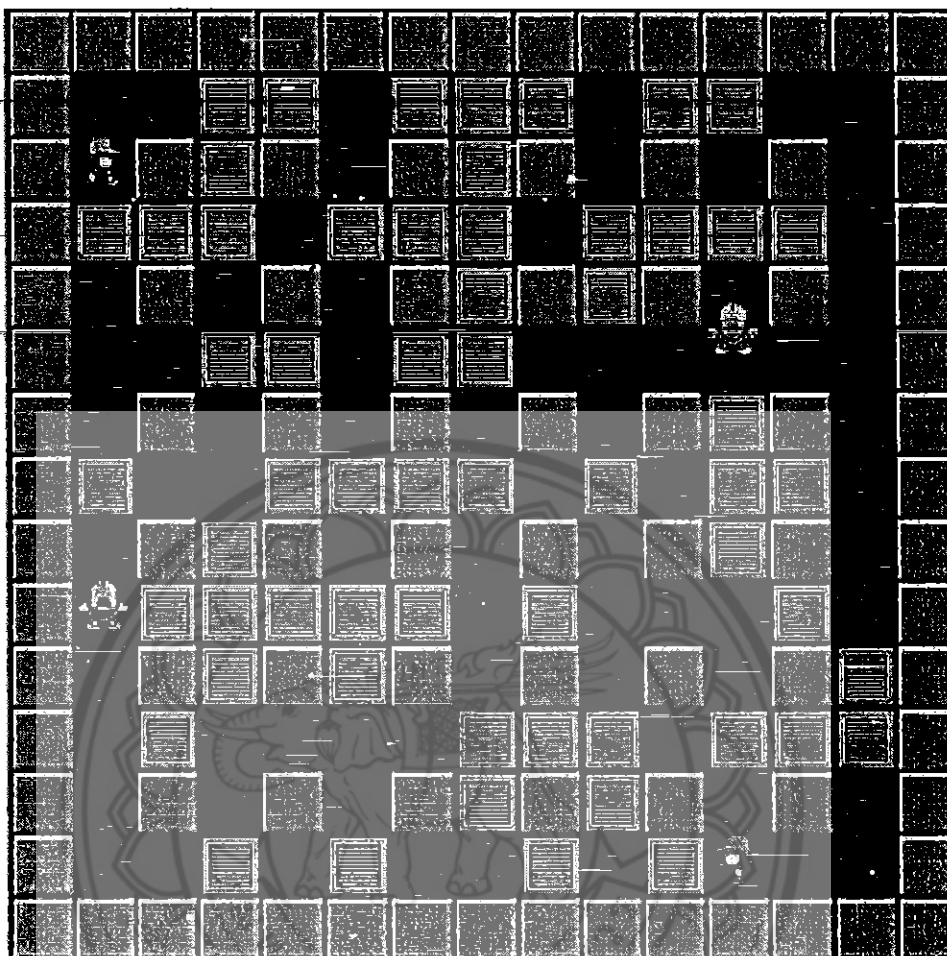
4.3 ผลการดำเนินงาน

หลังจากที่เครื่องลูกข่ายเข้ามาครบ 4 คนแล้วก็จะทำการเริ่มเกม โดยจะแบ่งเป็น 4 ผู้เล่น มุมซ้ายบน, มุมขวาล่าง, มุมซ้ายล่าง, มุมขวาบน คือ player1, player2, player3, player4 ตามลำดับชื่อที่เป็นสีน้ำเงินคือตัวที่เราบังคับอยู่ส่วนสีแดงคือศัตรู ในส่วนที่เป็นหน้าต่างแผ่นเสียงเราสามารถฟังเพลงพร้อมกับเล่นเกมได้ และกล่องข้อความสามารถพิมพ์ข้อความเพื่อสนทนาระหว่างการเล่นเกมที่



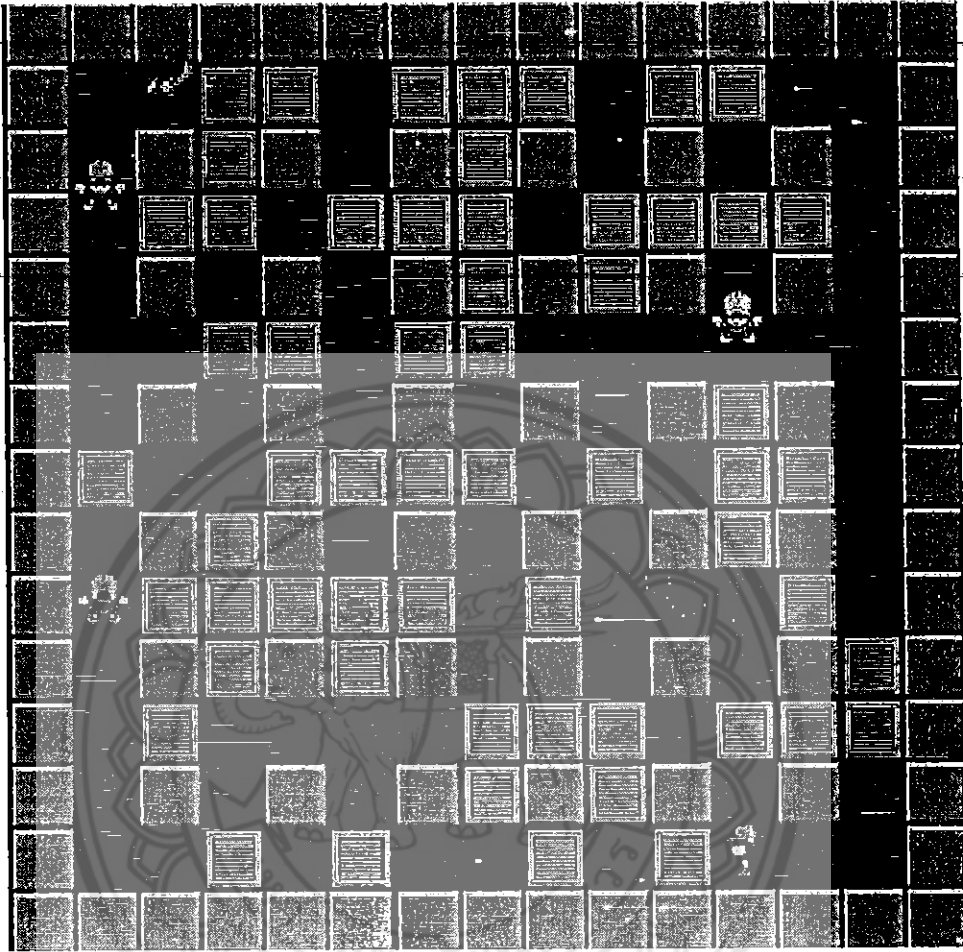
รูปที่ 4.4 แสดงหน้าต่างในการเล่นเกม

การแสดงการเดินของผู้เล่นซึ่งจะแสดงผลที่เครื่องถูกข่ายทั้งสองเหมือนกันแต่ภาพอาจจะ
กระตุกเล็กน้อย คังรูปที่ 4.5

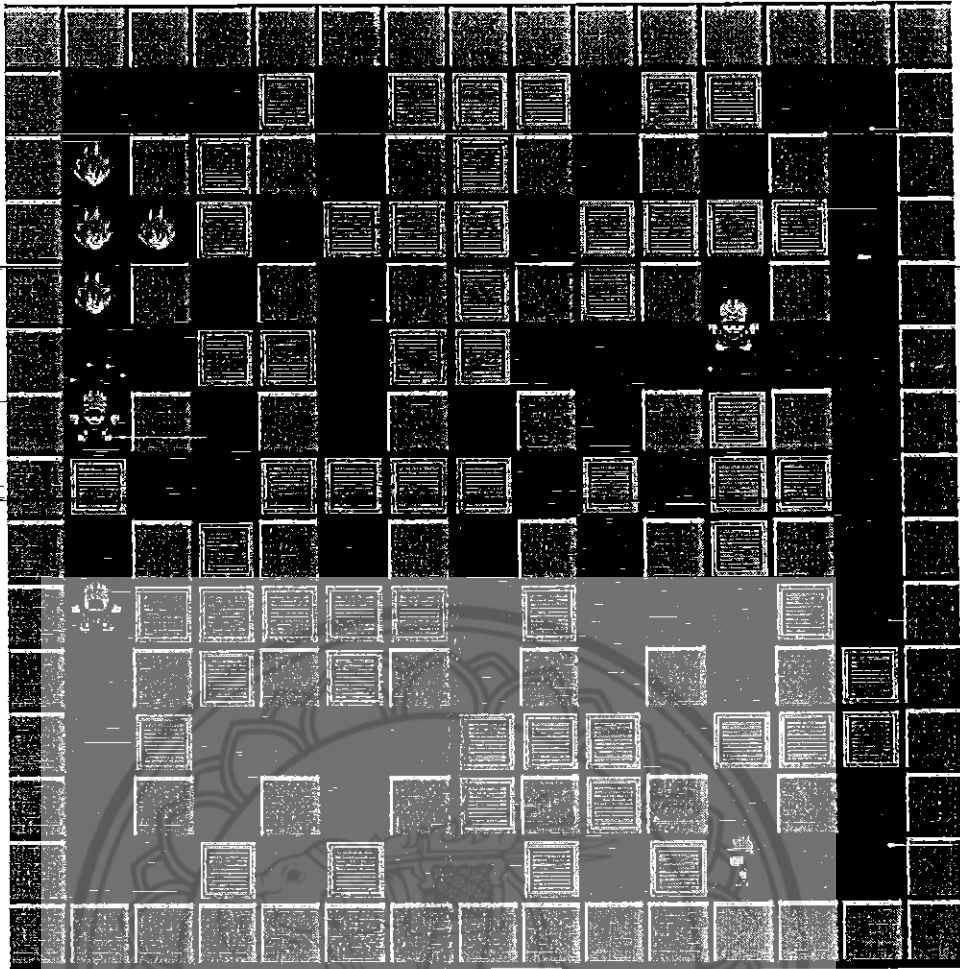


รูปที่ 4.5 แสดงการเดินของผู้เล่น

ลักษณะการวางระเบิดของผู้เล่นเมื่อผู้เล่นทำการกดปุ่ม ; ก็คือจะทำการวางลูกระเบิดซึ่งลูกระเบิดอยู่ในตำแหน่งเดียวกับตัวละคร ลูกระเบิดนี้ผู้เล่นไม่สามารถเดินผ่านได้ ดังรูปที่ 4.6 และจะระเบิดภายในระยะเวลาประมาณ 3 วินาที เกิดเป็นรูปไฟออกมา ดังรูปที่ 4.7

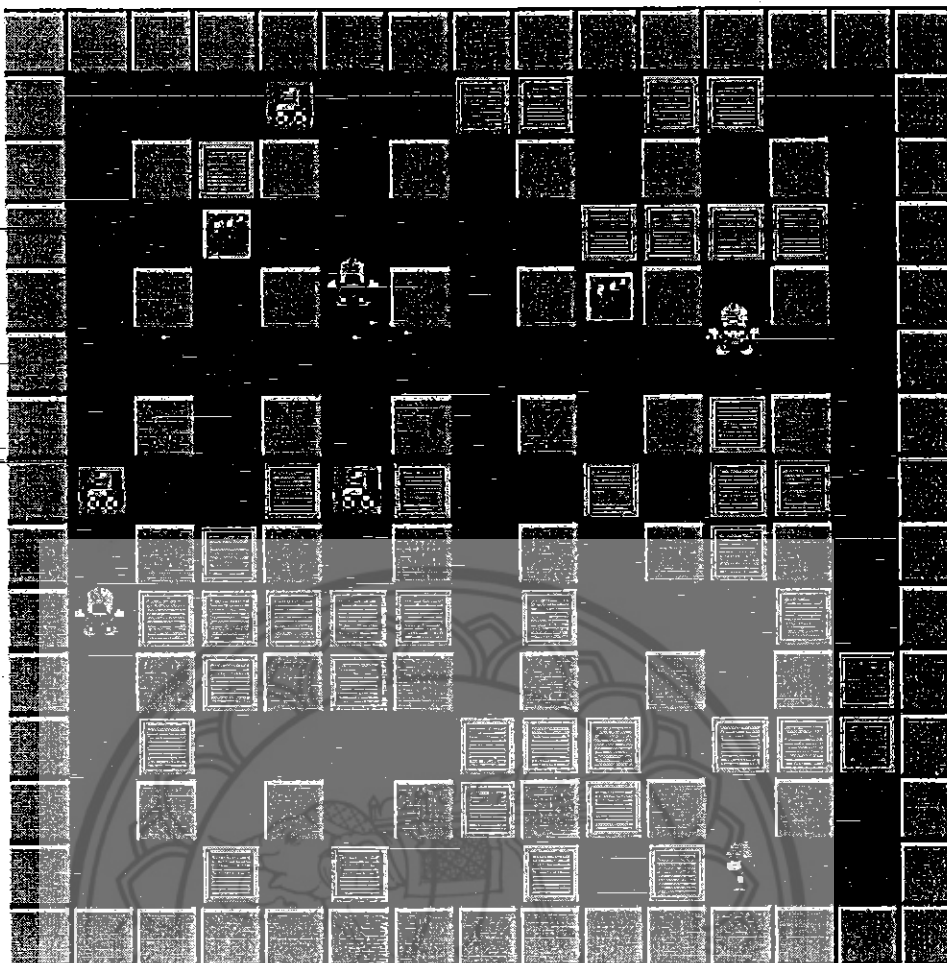


รูปที่ 4.6 แสดงการวางระเบิด



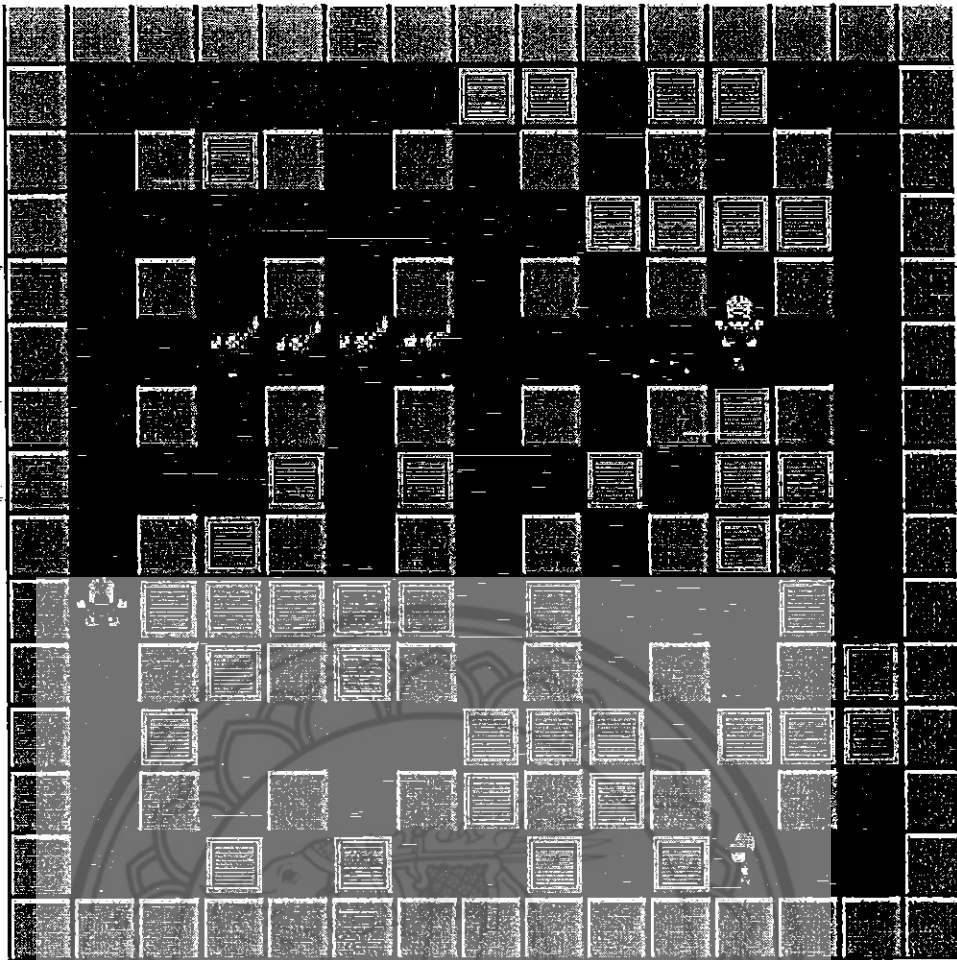
รูปที่ 4.7 แสดงการระเบิด

เมื่อมีการเก็บไอเทมก็จะได้ความสามารถเพิ่มขึ้นถ้าเก็บไอเทมระเบิดก็จะวางระเบิดได้เพิ่ม
 อีกหนึ่งลูก ถ้าเก็บรองเท้าสเก็ตก็จะวิ่งเร็วขึ้น 2 เท่าตัว



รูปที่ 4.8 แสดงไอเทมที่ซ่อนอยู่

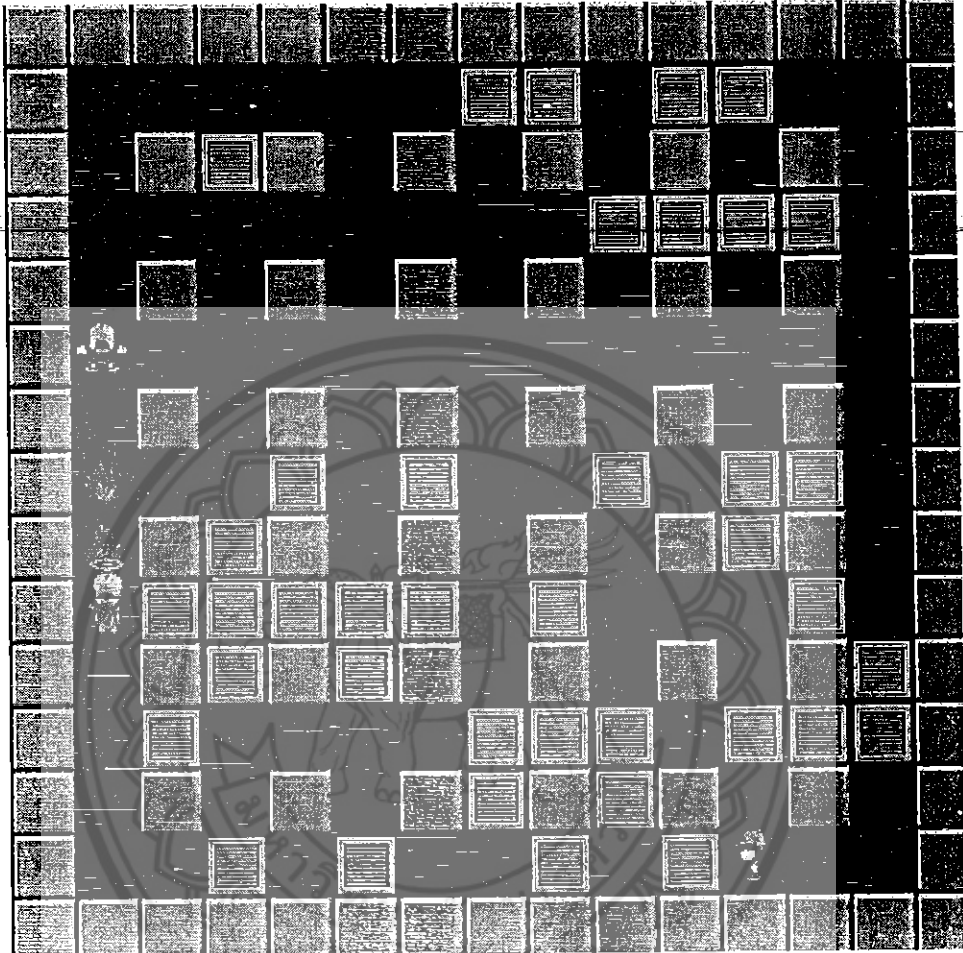




รูปที่ 4.9 แสดงการวางระเบียบเมื่อเก็บไอเทมระเบียบได้



เมื่อมีผู้เล่นฝ่ายใดที่โดนการระเบิดไม่ว่าถูกระเบิดจะเป็นผู้เล่นของฝ่ายใดก็ตามก็จะมีรูปไฟ
โดนตัวผู้เล่นนั้นและทำให้ผู้เล่นฝ่ายที่โดนระเบิดตายไปจนกระทั่งเหลือผู้เล่นคนสุดท้าย คือคนที่
ชนะ



รูปที่ 4.10 แสดงการถูกไฟของตัวละคร

บทที่ 5

บทสรุป

5.1 บทสรุป

จากการที่ได้ศึกษาเกี่ยวกับการพัฒนาเกมออนไลน์ด้วยโปรแกรม Macromedia FlashMX ด้วยการติดต่อสื่อสารข้อมูลทางซ็อกเก็ต จึงได้ข้อสรุปดังนี้

โครงการนี้ได้พัฒนาให้เกมนั้นสามารถที่จะรองรับผู้เล่นได้ถึง 4 คนพร้อมกัน และยังสามารถที่จะพูดคุยกันระหว่างเล่นเกมผ่านทางระบบ chat โดยที่เราได้ทดสอบในครั้งนี้ได้ใช้ซ็อกเก็ต XML ทำการติดต่อระหว่างเครื่องแม่ข่ายที่พัฒนาจากภาษา จาวา เราจะต้องนำไฟล์ของเครื่องแม่ข่ายและไฟล์ของเครื่องลูกข่ายไปไว้ที่เครื่องแม่ข่ายซึ่งจำเป็นจะต้องมี J2SDK เป็นโปรแกรมรองรับการทำงานที่เครื่องแม่ข่ายด้วย ในส่วนของเครื่องลูกข่ายนั้นจะต้องมีตัวเกมที่เป็นไฟล์ exe ผลที่ได้คือเกมดำเนินได้ราบรื่น แต่โปรแกรม Macromedia FlashMX นี้ ยังมีข้อจำกัดอย่างหนึ่งในการติดต่อคือ ข้อมูลของเกมส่วนใหญ่จะประมวลผลอยู่ที่เครื่องลูกข่ายซึ่งไม่ได้ประมวลผลที่เครื่องแม่ข่ายทำให้เมื่อการส่งผ่านมีความล่าช้าจะทำให้เกิดการดีเลย์บนเครื่องลูกข่ายซึ่งไม่ได้เป็น Real-Time System จริงๆ

5.2 ข้อเสนอแนะ

- Macromedia FlashMX สามารถทำงานบนเว็บเบราว์เซอร์ได้
- การนำไปพัฒนาในรูปแบบเกมออนไลน์ประเภทเล่นเป็นเทิร์นจะมีประสิทธิภาพ

มากกว่า

เอกสารอ้างอิง

- [1] กำพล สืลาภรณ์ปิย. *Advanced FLASH ActionScript*. กรุงเทพมหานคร : โปรวิชั่น.2544
-
- [2] วีระศักดิ์ ชิงदार. *Java Programming Volume I*. กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน). 2543.
-
- [3] วีระศักดิ์ ชิงदार. *Java Programming Volume II*. กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน). 2546.
-
- [4] วีระศักดิ์ ชิงदार. *Java Programming Volume III*. กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน). 2547.

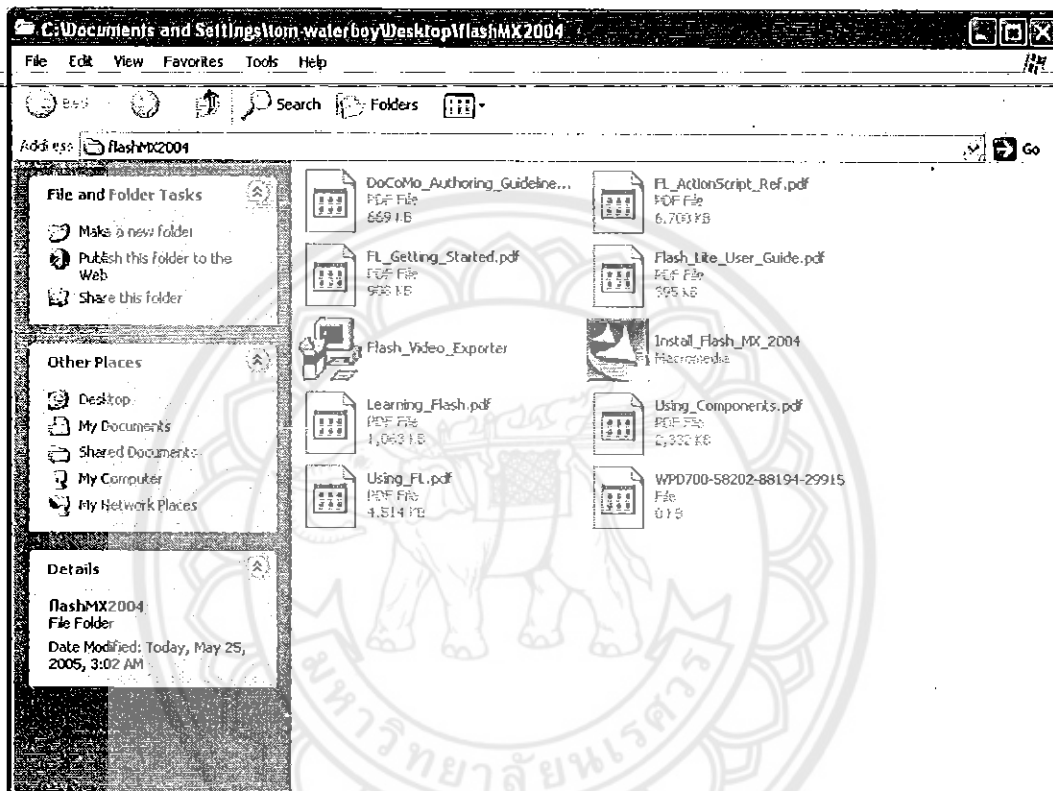




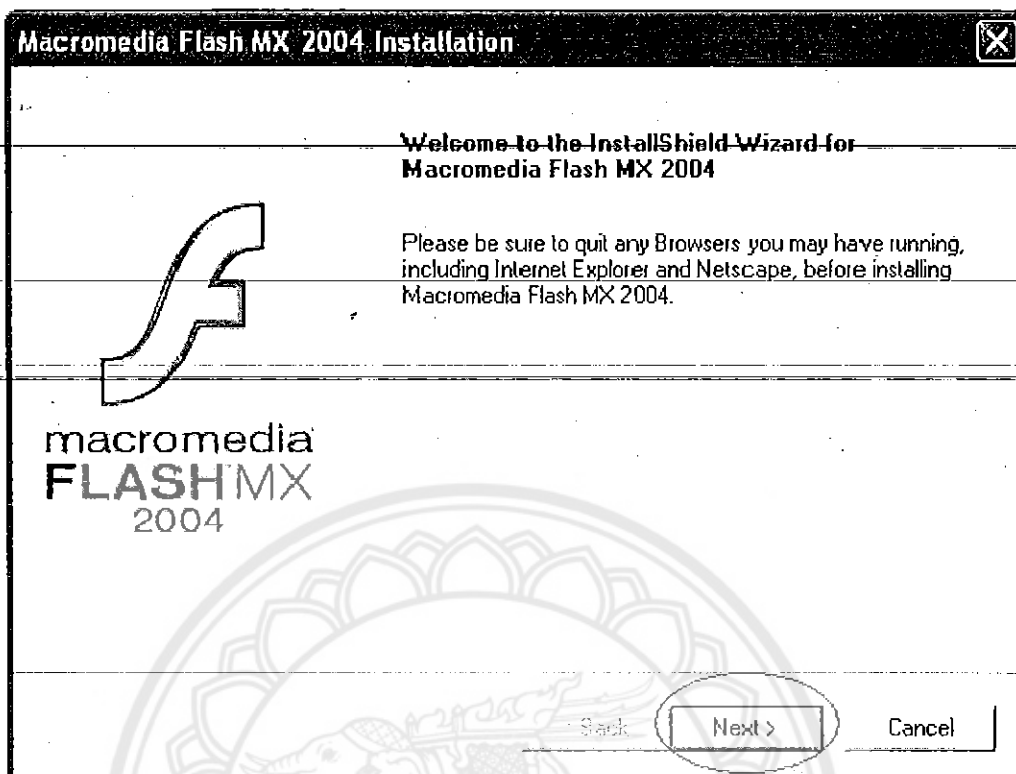
การติดตั้งโปรแกรม Macromedia Flash MX

ในการพัฒนาเกมออนไลน์นั้นจำเป็นต้องทดสอบ โปรแกรมที่เขียน โดยใช้โปรแกรม Macromedia Flash MX2004 ซึ่งมีวิธีการติดตั้ง โปรแกรม ดังนี้

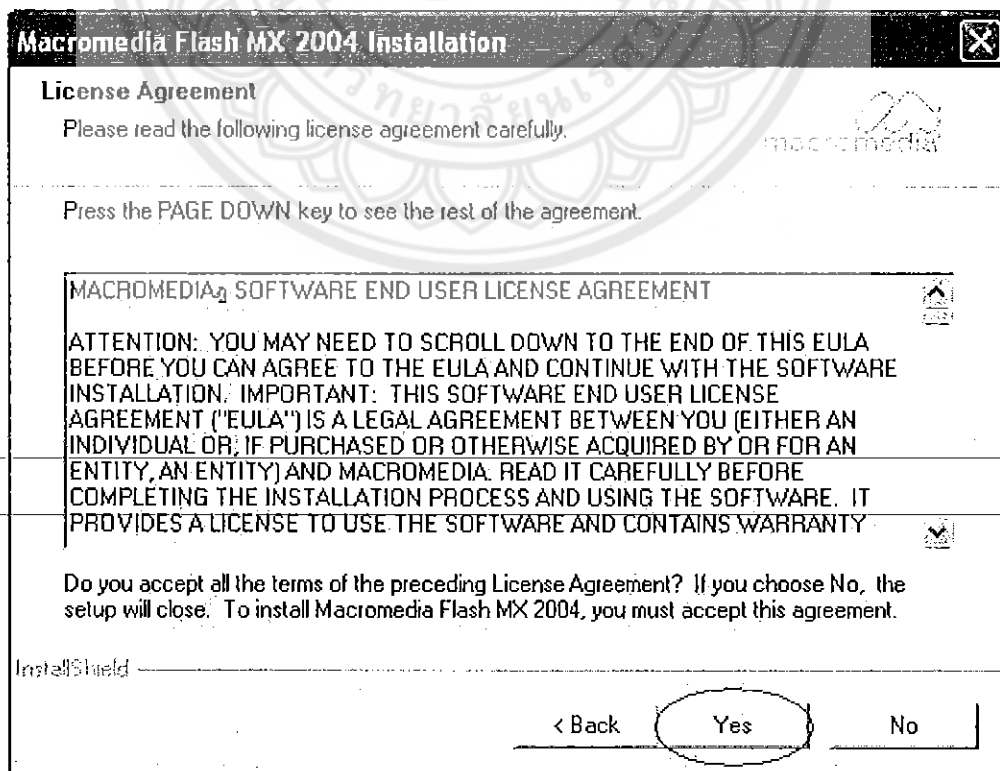
1. ดับเบิลคลิกไฟล์ที่มีชื่อว่า Install_Flash_MX_2004



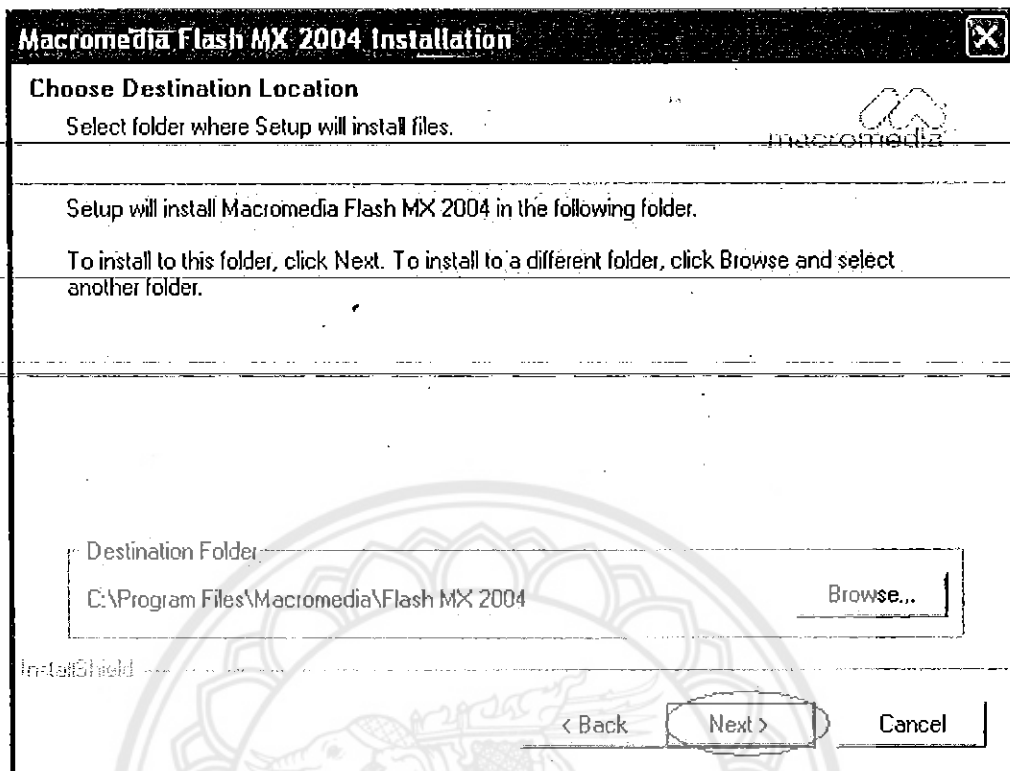
2.



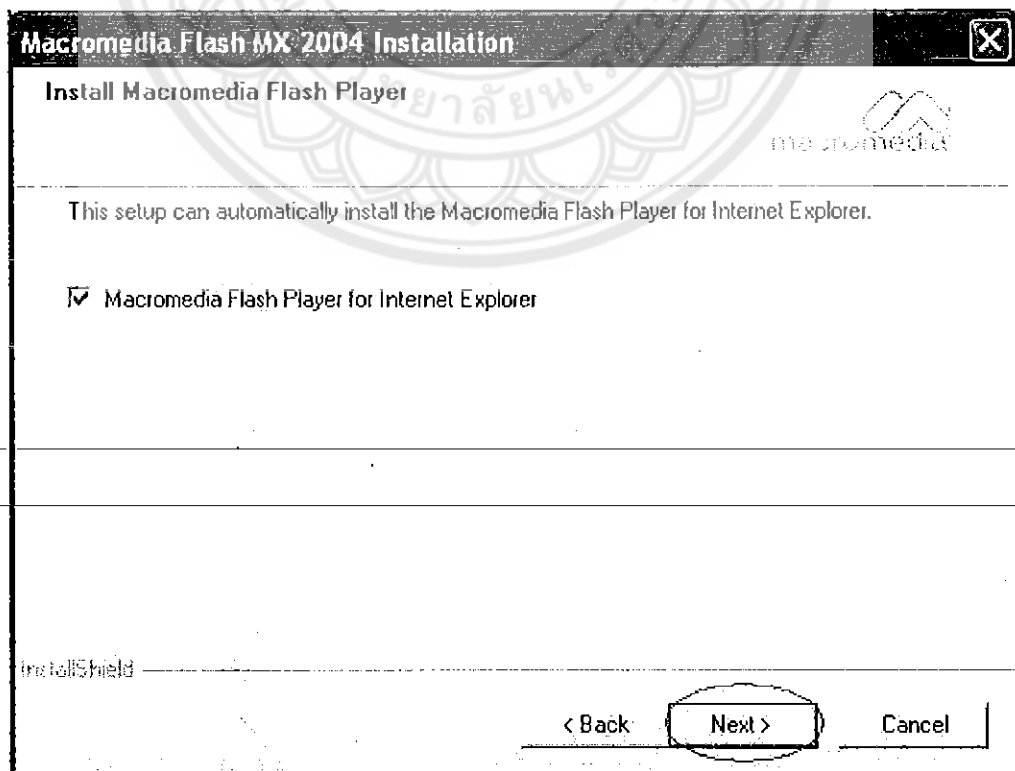
3.



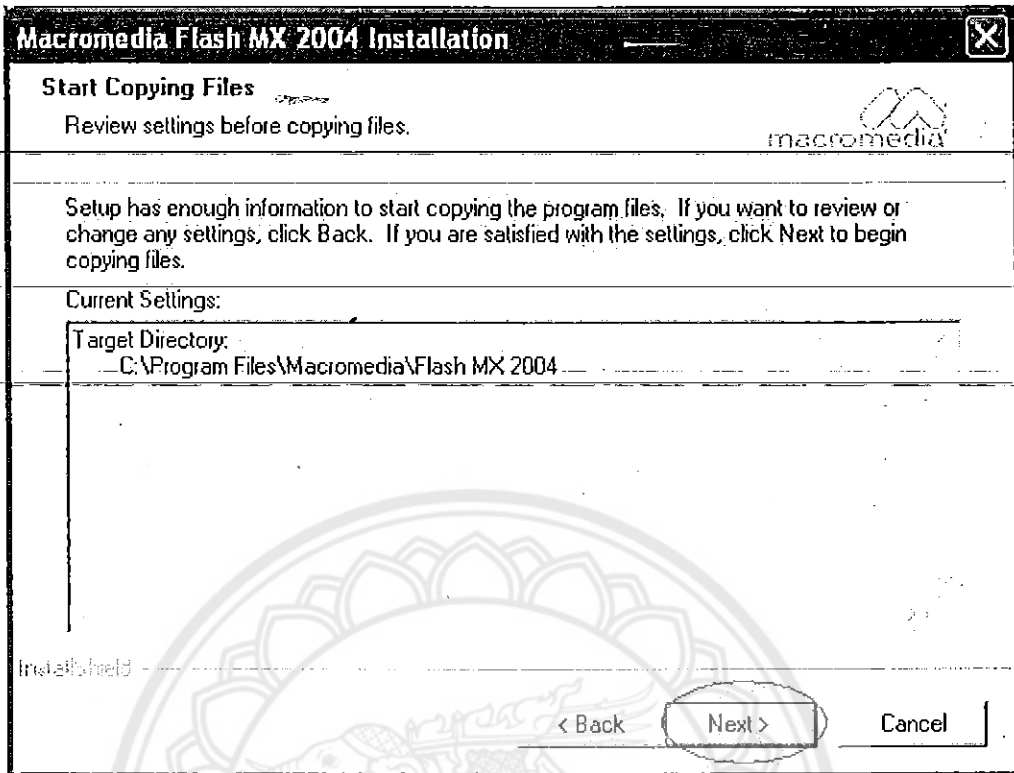
4.



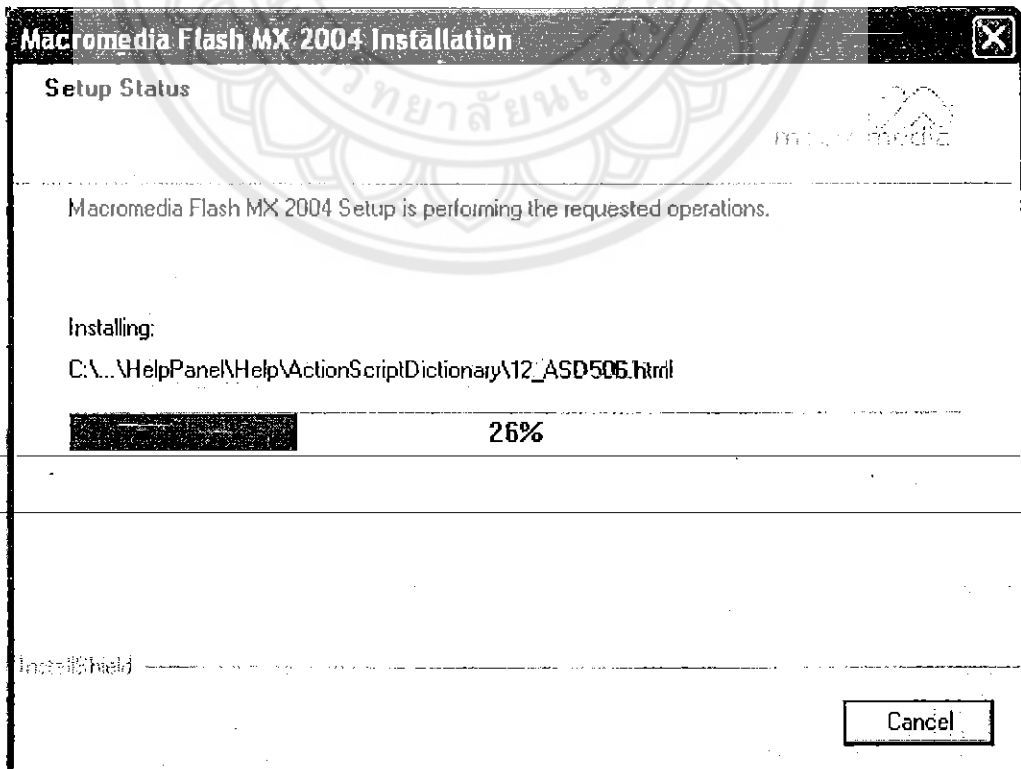
5.



6.



7.




8.

Macromedia Flash MX 2004 Installation

InstallShield Wizard Complete


Setup has finished installing Macromedia Flash MX 2004 on your computer.



Yes, view the Readme file now.

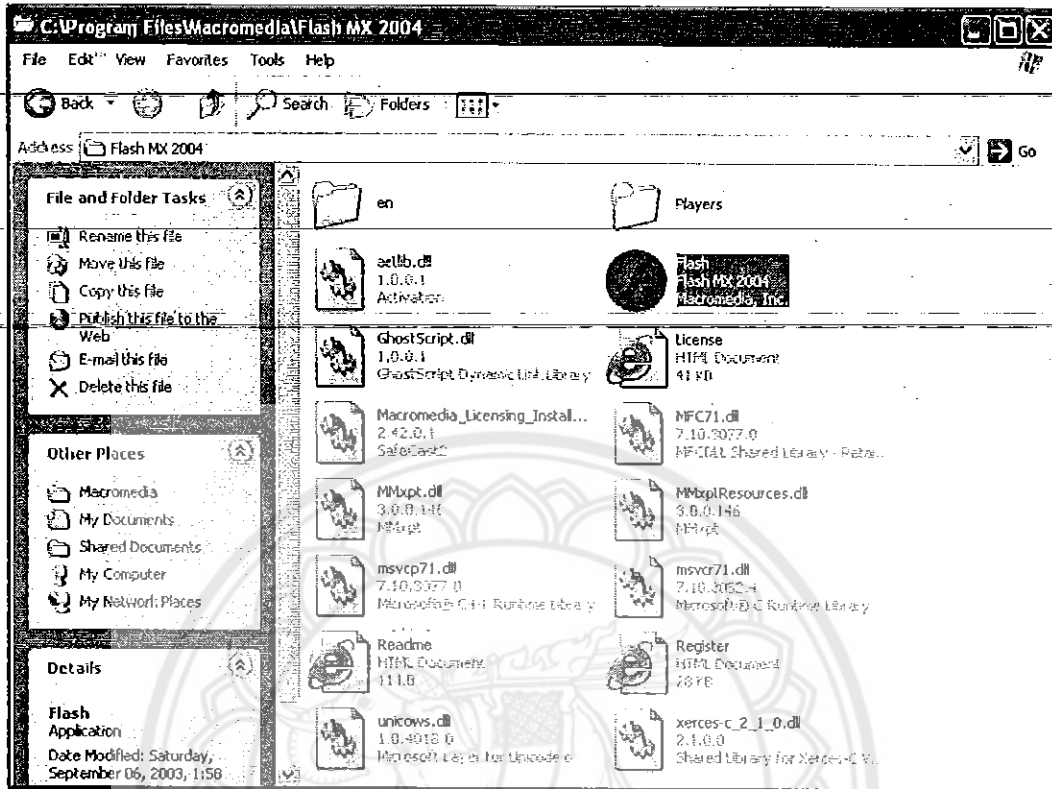
macromedia
FLASHMX
2004

Back Finish Cancel



เมื่อเสร็จขั้นตอนการติดตั้งสามารถเปิดโปรแกรม Macromedia Flash MX2004 ได้จาก

C:\Program Files\Macromedia\Flash MX 2004 ที่มีชื่อว่า Flash



ประวัติผู้เขียน



ชื่อ นาย กฤษณะ ศรีไกรรต
 ภูมิลำเนา 34/1 หมู่ 1 ต.เหล่าขาง อ.เวียงแก่น
 จ. เชียงราย 57310

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนจุฬารัตนาวิทยาลัย
 เชียงราย

- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4
 สาขาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
 คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

Email : tom_waterboy@hotmail.com



ชื่อ นาย ทรงชัย กูประภากร
 ภูมิลำเนา 352 หมู่ 2 ต.ต้า อ.ขุนตาล จ.เชียงราย 57340
 ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนจุฬารัตนาวิทยาลัย
 เชียงราย

- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4
 สาขาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
 คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

Email : yearpcc@hotmail.com