

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาเกมออนไลน์สามมิติ

3D Game Online Development



นายจักรชัย  
นายชยากร  
นายจิตติกร

แช่อึ้ง  
อาจารย์ยะ  
เกียรติรัตน์

รฟ.  
ฉ ๒๓๒๗  
๒๕๔๙

เลขหมู่.....  
เลขทะเบียน..... 72968  
วัน,เดือน,ปี 26 ส.ย. 2550

b. 11๙๙๕๖๙๕  
i.....

ปฏิญานีพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การพัฒนาเกมออนไลน์สามมิติ

3D Game Online Development



นายฉัตรชัย  
นายชยากร  
นายฐิติกร

แช่อึ้ง  
อาจารย์ยะ  
เถียไพรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาเกมออนไลน์สามมิติ

3D GAME ONLINE DEVELOPMENT

ผู้จัดทำ

- |               |             |                       |
|---------------|-------------|-----------------------|
| 1. นายฉัตรชัย | แซ่อึ้ง     | รหัสนักศึกษา 46010135 |
| 2. นายชยากร   | อาจารย์ยะ   | รหัสนักศึกษา 46010147 |
| 3. นายฐิติกร  | เธิยไพรัตน์ | รหัสนักศึกษา 46010184 |



อาจารย์ที่ปรึกษา

(ดร.วรวัฒน์ ลิ้มโกคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การพัฒนาเกมออนไลน์สามมิติ

นายจักรชัย แซ่อึ้ง	46010135
นายชยากร อาจารย์ยะ	46010147
นายฐิติกร เรียบไพรัตน์	46010184
ดร. วรวัฒน์ ลิ้มโกคา	อาจารย์ที่ปรึกษา
ปีการศึกษา 2549	

### บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการสร้างเกมออนไลน์สามมิติ ซึ่งเป็นเกมผจญภัยในโลกแฟนตาซี (RPG) โดยมีจุดเด่นในเรื่องของการเพิ่มประสิทธิภาพให้กับ Server Side จากการนำ Network Load Balancing Technology มาประยุกต์ ซึ่งผลงานชิ้นนี้ถูกพัฒนาโดย C++, MySQL Server 4.0 และ DirectX 9.0 SDK ที่มี API เหมาะในการพัฒนา Game



# 3D GAME ONLINE DEVELOPMENT

Mr.Chatchai Saeoung 46010135

Mr.Chayakorn Arjariya 46010147

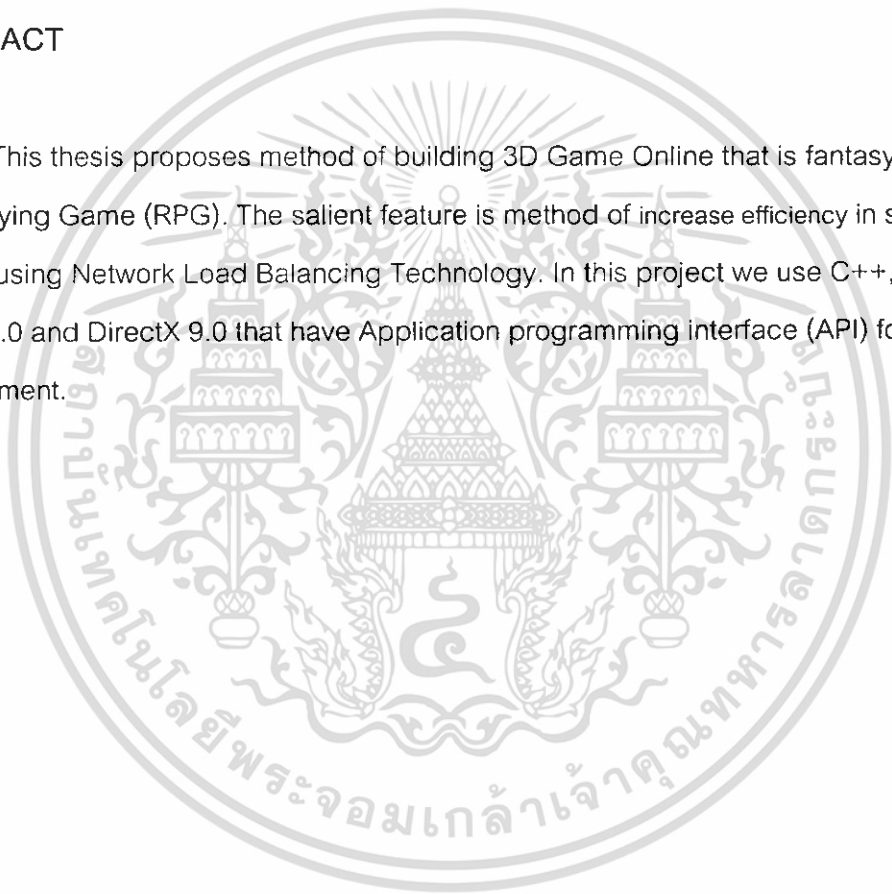
Mr.Thitikorn Terepirat 46010184

Dr.Voravat Limpoka Advisor

Academic Year 2006

## ABSTRACT

This thesis proposes method of building 3D Game Online that is fantasy and Role Playing Game (RPG). The salient feature is method of increase efficiency in server side by using Network Load Balancing Technology. In this project we use C++, MySQL Server 4.0 and DirectX 9.0 that have Application programming interface (API) for game development.



## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้เป็นอย่างดีด้วยคำแนะนำและคำปรึกษาจาก  
ดร.วรวัฒน์ ลิ้มโกศา ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญาบัตร ข้าพเจ้ารู้สึกซาบซึ้งในความ  
อนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับ  
ข้าพเจ้า

ขอขอบคุณเพื่อนๆ ทุกคนในห้อง Olala ที่คอยร่วมทุกข์ร่วมสุขอยู่ทำ Project กันมาทั้งปี  
รวมทั้งคอยให้คำแนะนำและคำปรึกษาเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็น  
กำลังใจ และให้การสนับสนุนในทุกๆ เรื่อง ทำให้ข้าพเจ้าสามารถทำปริญญาบัตรฉบับนี้สำเร็จ  
ลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากปริญญาบัตรฉบับนี้ ข้าพเจ้าขอมอบแต่ผู้มีพระคุณทุก  
ท่าน

นายฉัตรชัย แซ่ฮึง  
นายชยากร อาจารย์ยะ  
นายจตุติกร เขียวไพรัตน์

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ที่มาของโครงการ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของการศึกษา	1
1.4 ขั้นตอนการศึกษา	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 ส่วนประกอบของปริญญาานิพนธ์	3
บทที่ 2 ทฤษฎีพื้นฐาน	4
2.1 ความรู้เกี่ยวกับการสร้างภาพสามมิติ	4
2.2 Light	5
2.3 Shading	15
2.4 Texture	16
2.5 Camera	16
2.6 DirectX	17
2.7 X File	24
2.8 Data structure	26
2.9 Network Load Balancing และ Clustering	27
บทที่ 3 Graphic Tool	30
3.1 เหตุผลที่เลือกใช้	30
3.2 การสร้าง model สามมิติ	30

## สารบัญ(ต่อ)

	หน้า
บทที่ 4 การออกแบบและพัฒนา Software	43
4.1 เครื่องมือที่ใช้ในการพัฒนาโปรแกรม	43
4.2 ส่วนการทำงานของ Software	43
4.3 การทำงานของ Game (Client Programming)	44
4.4 การทำงานของ Program ควบคุมข้อมูลฝั่ง Database	52
4.5 การทำงานของ Server Load Balancing (Server Programming)	55
4.6 การทำงานของ Server (Server Programming)	55
บทที่ 5 Network Load Balancing กับเกมออนไลน์สามมิติ	56
5.1 Server Connection	57
5.2 Map Connection	58
5.3 Server & Map Connection	59
บทที่ 6 ผลการทดลอง	60
6.1 เนื้อหาของเกม	60
6.2 จุดเด่นของเกม	60
6.3 การดำเนินเกม	60
6.4 ตัวละคร	61
6.5 คู่ต่อสู้ภายในเกม	61
6.6 สถานที่ต่างๆ ภายในเกม	63
6.7 ค่าสถานะ (Status)	64
6.8 ค่าพื้นฐานและสูตรที่ใช้ในการคำนวณ	64
6.9 การต่อสู้ภายในเกม	65
6.10 การควบคุมภายในเกม	65
6.11 Interface ภายในเกม	66
6.12 กฎต่างๆ ภายในเกม	67
6.13 ค่าพื้นฐานของแต่ละตัวละคร	68
6.14 การทดลองการทำงานของ Game Server	69

## สารบัญ(ต่อ)

	หน้า
บทที่ 7 บทสรุป	71
7.1 ปัญหาและอุปสรรคที่เกิดจากการดำเนินงาน	71
7.2 แนวทางในการพัฒนาต่อ	72
บรรณานุกรม	73
ภาคผนวก	
ภาคผนวก ก คู่มือการติดตั้ง	74
การติดตั้ง My SQL Server 4.1	74
การติดตั้ง DirectX SDK 9.0	83
ภาคผนวก ข. การตั้งค่า Network Load Balancing (Window 2003 Server)	87



## สารบัญรูป

รูปที่	หน้า
1 แสดง Face ที่เกิดจาก Vertex สามจุด	7
2 แสดงการฉายแสงของ Spot light	9
3 แสดงการเรียกใช้งานของ DirectX	19
4 แสดงการติดต่อกับ hardware โดยผ่าน DirectX	22
5 แสดง Data type ที่ใช้ใน X File	24
6 แสดงโครงสร้าง Clustering	27
7 แสดงหลักการทำงานของ Network Load Balancing	28
8 แสดงการทำงานของ Network Diagram NLB	29
9 แสดง Interface ของโปรแกรม Milk shapes	30
10 แสดงการใช้งานเบื้องต้น Milk shapes	31
11 แสดงการใช้งานเบื้องต้น Milk shapes	32
12 แสดง Interface การขึ้นรูปเบื้องต้น	33
13 แสดง Interface การขึ้นรูปตัวละคร	33
14 แสดงการสร้างภาพเคลื่อนไหว 1	34
15 แสดงการสร้างภาพเคลื่อนไหว 2	34
16 แสดงการสร้างภาพเคลื่อนไหว 3	35
17 แสดงการปะ Texture 1	36
18 แสดงการ Export File เป็น .x เพื่อใช้งาน	37
19 แสดงการ Export File เป็น .x เพื่อใช้งาน 2	39
20 แสดงภาพการประยุกต์ภาพ Texture	41
21 แสดงการเปรียบเทียบภาพก่อนและหลังฉายแสง	42
22 แสดงโครงสร้างของระบบ	43
23 State Diagram ของ Software Client	46
24 Flow Chart แสดงถึงการทำงานหลักๆของ Program Client	47
25 Class Diagram ข้อมูลที่จะใช้ในการแสดงผล	49
26 แสดง Class Diagram การจัดการเสียงและแสดงภาพ 2 มิติ	50
27 แสดง State Diagram ของ Client Application	51
28 State Diagram ตัวศัตรูฝั่ง Database	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป(ต่อ)

รูปที่	หน้า
29 Flow Chart แสดงการทำงาน ที่ใช้ในการควบคุมตัวศัตรู	54
30 หลักการทำงานของ Network Load Balancing	56
31 แสดงโครงสร้าง Server Connection	57
32 แสดงโครงสร้าง Map Connection	58
33 แสดงโครงสร้าง Server & Map Connection	59
34 แสดงความสัมพันธ์ระหว่างจำนวน Connection กับ Response Time	69
35 แสดงความสัมพันธ์ระหว่างจำนวน Connection กับ RAM	69
36 แสดงความสัมพันธ์ระหว่างจำนวน Connection กับ CPU Usage	70
37 แสดงความสัมพันธ์ปริมาณข้อมูลเข้า-ออก Server	70
ก.1.1 แสดงหน้าเริ่มต้นการลงโปรแกรม	73
ก.1.2 แสดง Setup Type	73
ก.1.3 แสดง Ready to Install the Program	74
ก.1.4 แสดง Installing MySQL Server 4.1	74
ก.1.5 แสดง MySQL.com Sign-up	75
ก.1.6 แสดง Wizard Completed	75
ก.1.7 แสดง Welcome to the....	76
ก.1.8 แสดง MySQL Server Instance Configuration 1	76
ก.1.9 แสดง MySQL Server Instance Configuration 2	77
ก.1.10 แสดง MySQL Server Instance Configuration 3	77
ก.1.11 แสดง MySQL Server Instance Configuration 4	78
ก.1.12 แสดง MySQL Server Instance Configuration 5	78
ก.1.13 แสดง MySQL Server Instance Configuration 6	79
ก.1.14 แสดง MySQL Server Instance Configuration 7	79
ก.1.15 แสดง MySQL Server Instance Configuration 8	80
ก.1.16 แสดง MySQL Server Instance Configuration 9	80
ก.1.17 แสดง MySQL Server Instance Configuration 10	81
ก.1.18 แสดง MySQL Server Instance Configuration 11	81
ก.2.1 หน้าต่างเงื่อนไขในการติดตั้งDirectx	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญญรูป(ต่อ)

รูปที่	หน้า
ก.2.2 หน้าต่างต้องรับก่อนการติดตั้งDirectx	82
ก.2.3 หน้าต่างให้เลือก Function ที่ต้องการทำกับDirectx	83
ก.2.4 หน้าต่างเลือก Feature ที่ต้องการติดตั้งของDirectx	83
ก.2.5 หน้าต่างแสดงการติดตั้งDirectx	84
ก.2.6 หน้าต่างเงื่อนไขในการติดตั้งDirectx	84
ข.1 แสดง Local Area Connection Properties	86
ข.2 แสดง Network Load Balancing Properties	87
ข.3 แสดง Network Load Balancing Properties 2	88
ข.4 แสดง NetWork Load Balancing Properties 3	89



# บทที่ 1

## บทนำ

### 1.1 ที่มาของโครงการ

เนื่องจากในปัจจุบัน เกมสามมิติออนไลน์เป็นเกมอีกประเภทหนึ่งที่เป็นที่นิยมแพร่หลาย เป็นอย่างมาก ซึ่งตัวเกมนั้นมีอยู่หลากหลายแนวด้วยและรองรับผู้เล่นทุกเพศทุกวัย แต่เกมออนไลน์ประเภทที่โดดเด่นมากที่สุดคือ เกมแนวผจญภัย (RPG) ออนไลน์ ซึ่งผู้เล่นสามารถต่อสู้ ผจญภัยในโลกแฟนตาซีไปพร้อมกับผู้เล่นคนอื่นๆ ได้ในเวลาเดียวกัน ทางผู้พัฒนาเกมจึงได้มีแนวคิดและทำการออกแบบเกมสามมิติออนไลน์ขึ้น โดยผู้พัฒนาเกมพยายามหาจุดเด่นและนำระบบใหม่ๆ เข้ามาใช้ในเกมเพื่อเพิ่มความสนุกสนานในการเล่น

### 1.2 วัตถุประสงค์

1. เพื่อทำการศึกษากonstrukสร้างและพัฒนาเกมสามมิติ
2. เพื่อทำการศึกษากonstrukสร้างเกมในรูปแบบออนไลน์ ซึ่งรองรับผู้เล่นได้เป็นจำนวนมาก
3. เพื่อทำการศึกษากonstrukทำงานของระบบเครือข่ายและพัฒนาการทำงานในลักษณะของ Network Load Balancing(NLB)
4. เพื่อศึกษาการใช้งานและออกแบบโมเดลสามมิติด้วย Milk shapes 3D 1.7.8

### 1.3 ขอบเขตของการศึกษา

โครงการนี้เป็นการศึกษาเพื่อทำการสร้างและพัฒนาเกมสามมิติออนไลน์ โดยทำการศึกษาดังแต่การสร้างโมเดลสามมิติ แล้วนำโมเดลที่ได้มาพัฒนาต่อบนโปรแกรมเพื่อเรนเดอร์ขึ้นสามมิติให้แสดงผลในรูปแบบของเกมผจญภัย นอกจากนี้ยังศึกษาในเรื่องของ Network เพื่อให้ผู้เล่นจำนวนมากสามารถเล่นไปพร้อมๆกันและรวมไปถึงการจัดการ Server เพื่อรองรับ Load ในจำนวนที่เพิ่มมากขึ้นด้วย ตัวเกมสามารถเล่นบนเครื่องคอมพิวเตอร์ส่วนบุคคลได้ในรูปแบบของเครื่อง Client โดยก่อนเริ่มเกมทุกครั้งจะต้องทำการติดต่อไปยังเครื่อง Server เพื่อ Login เข้าสู่เกม ข้อมูลของผู้เล่นทุกอย่างจะต้องถูกบันทึกไว้ใน Database ทำให้ผู้เล่นสามารถใช้คอมพิวเตอร์เครื่องใดก็ได้ที่มีการติดตั้งโปรแกรมที่ทำงานในส่วนของ Client เอาไว้ในการเล่น

## 1.4 ขั้นตอนการศึกษา

1. ศึกษาเกี่ยวกับแนวเกมผจญภัยสามมิติออนไลน์เพื่อทำการออกแบบเกม
2. วิเคราะห์และออกแบบข้อมูลต่างๆภายในเกม เช่น การดำเนินเกม ตัวละคร ค่าสถานะและระบบต่างๆภายในเกม เป็นต้น
3. ศึกษาข้อมูลต่างๆในการพัฒนาเกม เช่น ข้อมูลเกี่ยวกับการสร้างโมเดลสามมิติ ข้อมูลเกี่ยวกับการเขียน Code ด้วยภาษาต่างๆ เป็นต้น
4. จัดหาและเลือก Program ที่จะนำมาใช้ในการพัฒนาเกมสามมิติออนไลน์
5. ทำการสร้างและพัฒนาโปรแกรมตามที่ได้ออกแบบไว้
6. ทำการตรวจสอบและแก้ไขข้อผิดพลาดที่พบ

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ความเข้าใจในการสร้างและพัฒนาเกมสามมิติออนไลน์
2. ได้รับความรู้ในการออกแบบและสร้างโมเดลสามมิติ
3. ได้รับความรู้ความชำนาญในการพัฒนาโปรแกรมด้วยภาษา C++
4. ได้รับความรู้เกี่ยวกับการทำงานของระบบ Network Client-Server
5. ได้รับความรู้เกี่ยวกับเทคโนโลยี Network Load Balancing

## 1.6 ส่วนประกอบของปฏิญญานิพนธ์

ปฏิญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 7 บทด้วยกันคือ

บทที่ 1 เป็นบทที่กล่าวถึงที่มาของโครงการ วัตถุประสงค์ของโครงการ ของเขตของโครงการ ขั้นตอนการศึกษา ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปฏิญญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงการ ซึ่งประกอบด้วยความรู้พื้นฐานเกี่ยวกับสามมิติ การให้แสง และการทำเทคนิคต่างๆของสามมิติ ความรู้พื้นฐานเกี่ยวกับชุด API ของ DirectX ที่จะนำมาใช้ในการเขียนโปรแกรม รวมไปถึงการนำโมเดลสามมิติมาใช้ในโปรแกรมโดยผ่าน DirectX File นอกจากนี้ยังกล่าวถึงทฤษฎีต่างๆเกี่ยวกับ Network Load Balancing และ Clustering อีกด้วย

บทที่ 3 กล่าวถึงเครื่องมือต่างๆที่จะนำมาใช้พัฒนาโมเดลสามมิติ รวมไปถึงการใช้งานและอธิบายถึงการสร้างโมเดลสามมิติอย่างง่าย

บทที่ 4 กล่าวถึงการออกแบบ Software ส่วนต่างๆ บอกถึงเป้าหมายในการพัฒนาพัฒนาการของโปรแกรม รวมไปถึงโมเดลแสดงถึงการทำงานของ Software ด้วย

บทที่ 5 เป็นบทที่กล่าวถึงการนำความรู้เกี่ยวกับ Network Load Balancing มาใช้ในเกมออนไลน์ โดยออกแบบระบบ Network ให้เหมาะสมกับการนำมาใช้ในการทำเกมออนไลน์

บทที่ 6 แสดงถึงผลการทดลองทั้งหมดที่ได้จากโครงการ ในบทนี้อธิบายถึงตัวเกมทั้งหมด ตัวละคร ฉาก การทำงาน การควบคุม รวมไปถึงกฎต่างๆภายในเกม ในขณะเดียวกันก็มีผลการทดลองการทำงานของ Server ด้วย

บทที่ 7 บทสรุป เป็นบทที่สรุปปัญหาและอุปสรรคต่างๆที่ค้นพบภายในโครงการ รวมไปถึงแสดงถึงแนวทางการพัฒนาต่อของโครงการ

## บทที่ 2

# ทฤษฎีพื้นฐาน

### 2.1 ความรู้เกี่ยวกับการสร้างภาพสามมิติ

#### 1. ส่วนประกอบของ Model สามมิติ มีดังนี้

- Vertex คือ จุด (point) ซึ่งวางอยู่บนระนาบแกน โดยมีพิกัดเป็น  $(x, y, z)$
- Vertices คือ เซตของจุดที่ใช้แทนตำแหน่งของวัตถุ เช่น Mesh ใน 3D Space
- Edge คือ เส้นที่เชื่อมต่อระหว่าง Vertex หนึ่งไปยังอีก Vertex หนึ่ง
- Polygon คือ ระนาบที่เกิดจากการนำ Edge มาวางเรียงต่อกัน โดยที่ 1 Polygon จะต้องมียาวน้อย 3 Edge การทำภาพที่มีลักษณะโค้งจะใช้หลายๆ Polygon มาเรียงต่อกันให้ดูโค้งแทนการทำให้เป็นเส้นโค้งจริงเพื่อประหยัดเวลาในการคำนวณ ภาพ 3 มิติที่มีลักษณะโค้งที่มีจำนวนของ Polygon มากจะทำให้ภาพมีความโค้งมนสมจริงมากขึ้น แต่ก็ต้องใช้เวลาในการประมวลผลภาพมากยิ่งขึ้น
- Face คือ ส่วนประกอบที่อยู่ใน Mesh หรือ Poly ที่ถูกแบ่งครั้งนั่นเอง หรือก็คือ Vertices ที่มีตั้งแต่ 3 จุดขึ้นไปมาเชื่อมต่อกันเป็นรูปทรงต่างๆในแนวระนาบ Vertex เป็นตัวกำหนดมุมของ Face ทำให้ทุกๆ Vertex ใน Face จะต้องถูกกำหนดให้อยู่ในแนวระนาบ
- Mesh เกิดจากการรวมกันของ Face ที่เชื่อมต่อกัน ซึ่ง 1 Mesh สามารถมี Face ได้ตั้งแต่ 1 Face ขึ้นไป การรวมของ Face นี้ทำให้ง่ายต่อการจัดการวัตถุในการทำ Animation, Material และ Texture ชนิดการรวมกันของ Face มีดังนี้คือ

1. Fan คือ กลุ่มของรูปทรงสามเหลี่ยม ซึ่งทุกรูปมีการใช้ Vertex ร่วมกัน 1 จุด โดยการกำหนดให้ Vertex นั้นๆอยู่ระหว่างกลางของสามเหลี่ยมเหล่านั้นชนิดของการรวม Face แบบ Fan นี้คล้ายกับแบบ Strip คือมีการกำหนดค่า Vertices 3 ค่าแรกสำหรับสามเหลี่ยมแรก จากนั้นในสามเหลี่ยมต่อไปก็เพียงแค่เพิ่มขึ้นมารูปละ 1 Vertex เท่านั้น

2. Strip คือ กลุ่มของสามเหลี่ยม ซึ่งแต่ละรูปจะมีการใช้เส้นร่วมกันกับสามเหลี่ยมส่วนหน้าซึ่งหมายความว่า หลังจากที่มีการกำหนดค่า Vertices 3 ค่าแรกสำหรับสามเหลี่ยมแรก จากนั้นในสามเหลี่ยมต่อไปก็เพียงแค่เพิ่มขึ้นมารูปละ 1 Vertex นั่นเอง

3. List คือ กลุ่มของสามเหลี่ยม ซึ่งทุกรูปไม่มีการใช้เส้นหรือ Vertex ร่วมกันเลย กล่าวคือ ค่า 3 ค่าของสามเหลี่ยมทุกรูปต้องกำหนดเองทั้งหมด

-Materials เป็นค่าที่กำหนดว่าจะให้ Polygon Surface เป็นอย่างไรเมื่อกระทบกับแสง โดยใน Materials มีค่าต่างๆอยู่ด้วยกัน 4 ค่า ดังนี้

1. Diffuse Color เป็นค่าของสีที่มักกำหนดเป็น RGB บอกว่า Vertices จะเป็นสีอะไรเมื่อกระทบกับแสง
2. Ambient Color เป็นสีที่ตรงข้ามกับ Diffuse Color คือเป็นสีที่จะแสดงเมื่อแสงหายไป
3. Emissive Color เป็นสีที่แสดงเป็น Effect of Fluorescence (สีส่องสว่าง) เมื่อ Verticesกระทบแสง
4. Specular Color เป็นค่าที่ทำให้วัตถุส่องสว่าง โดยต้องกำหนดค่า Power ยิ่งมากยิ่งส่องสว่าง หากค่า Power เป็น 0 วัตถุจะไม่ส่องสว่าง

## 2.2 Light (แสง)

แสงทำให้เกิดการเปลี่ยนแปลงของสีของ Vertices โดย Module ทำให้เกิด Vertex Normal เพราะสิ่งนี้ขึ้นอยู่กับมุมของแหล่งกำเนิดแสงตามปกติจะมีแสงสีขาว เพราะเป็นการรวมกันอย่างหนาแน่นของสีทุกสีและโดยมากมักใช้รูปแบบของ RGB ในการกำหนดสีของแหล่งกำเนิดแสงใน 3D นั้นมีการตั้งค่า RGB ของแม่สีต่างๆดังนี้

แสงสีขาว	เป็น	1, 1, 1
แสงสีแดง	เป็น	1, 0, 0
แสงสีน้ำเงิน	เป็น	0, 0, 1

นอกจากนี้เรายังสามารถใช้ 3 สี นี้ในการผสมสีได้ ซึ่งแหล่งกำเนิดแสงมี 4 ชนิด ได้แก่

- Ambient Light คือแหล่งกำเนิดแสงที่ง่ายที่สุด เพราะไม่ต้องมีการกำหนดตำแหน่งของแหล่งกำเนิดแสง และยังให้ความสว่างทั่วทุก Object
- Point Light เป็นแหล่งกำเนิดแสงที่ทำการกระจายแสงไปทุกทิศทาง แต่ต้องระบุตำแหน่งของแหล่งกำเนิดแสง โดยไม่ต้องกำหนดทิศทางของแสง
- Directional Light เป็นแหล่งกำเนิดแสงที่มีประสิทธิภาพมากที่สุดเพราะ เป็นแหล่งกำเนิดแสงที่มีทิศทาง โดยต้องระบุตำแหน่งของแหล่งกำเนิดแสง
- Spot Light เป็นแหล่งกำเนิดแสงที่ต้องมีการระบุทั้งทิศทางและตำแหน่งของแหล่งกำเนิดแสง โดยการผลิตแสงจะเป็นรูปทรง

ในการกำหนด Flexible Vertex Format (FVF) หากเราต้องการให้จุด vertex ของเรามี พิกัด Normal ด้วยนั้นเราจะต้องเติม Flag D3DFVF\_NORMAL ลงไปด้วย จากนั้นจึงมากำหนด โครงสร้างข้อมูลของจุด vertex ให้มีพิกัด Normal เพิ่มเข้าไปด้วยตัวอย่างเช่น

```
#define CUSTOM_FVF
D3DFVF_XYZ|D3DFVF_NORMAL|D3DFVF_DIFFUSE|D3DFVF_TEX1
struct Vertex
{
    float x,y,z; // D3DFVF_XYZ
    float nx,ny,nz; // D3DFVF_NORMAL
    DWORD dif; // D3DFVF_DIFFUSE
    float u,v; // D3DFVF_TEX1
}
```

ต่อมาเราจะต้องมาคำนวณ Normal Vector จากจุด Vertex ของ Primitive (Polygon) โดยเราจะนำจุด Vertex ของ Primitive ในตำแหน่ง P0 มาลบกับ P1 และ P0 มาลบกับ P2 จากนั้นจึงนำผลลัพธ์มาคูณ (Cross-Product) กันอีกทีแล้วนำไป Normalize ต่อจึงจะได้ Normal Vector มาหนึ่งจุด ตอนนี้จะได้แค่ Face Normal เอง

```

/**Normal_vector = Normalize((p0-p1)*(p0-p2)
vec3d ComputeNormalVector(vec3d p0,vec3d p1,vec3d p2)
{
    vec3d vNormal,v1,v2;
    v1 = Subtract(p0,p1);
    v2 = Subtract(p0,p2);
    vNormal = CrossProduct(v1,v2);
    return Normalize(vNormal);
}

```



รูปที่ 1 แสดง Face ที่เกิดจาก Vertex สามจุด

ต่อมาเราจะมาหา Vertex Normal วิธีการที่เราใช้ในการหา Vertex Normal ก็คือการนำผลรวมของ Face Normal ของ Polygon ที่มีจุด Vertex (ที่จะหา Vertex Normal) นั้นประกอบอยู่ทั้งหมดหารด้วยจำนวน Polygon ที่มีจุด Vertex นั้นประกอบอยู่ เมื่อนำมาเขียนเป็นโค้ดจะได้ดังนี้

```

void ComputeVertexNormals(vec3d* pVertices,unsigned int* pIndices,vec3d* pNormal,
                        int VertexCount,int PrimitiveCount)
{
    vec3d* pTempNormal,vSum;
    float share = 0.0f;
    int i,j;
    pTempNormal = new vec3d[PrimitiveCount];
    pNormal = new vec3d[VertexCount];
    for(i=0;i<PrimitiveCount;i++)
    {
        pTempNormal[i] = ComputeNormalVector(pVertices[pIndices[i*3]]
                                            ,pVertices[pIndices[i*3+1]]
                                            ,pVertices[pIndices[i*3+2]]);
    }
    for(i=0;i<VertexCount;i++)
    {
        vec3d v = pVertices[i];
        vSum = Set(0.0f,0.0f,0.0f);
        share = 0.0f;
        for(j=0;j<PrimitiveCount;j++)
        {
            if(pVertices[pIndices[j*3]] == v|| pVertices[pIndices[j*3+1]] == v||
                pVertices[pIndices[j*3+2]] == v)
            {
                vSum = Add(vSum,pTempNormal[j]);
                share++;
            }
        }
        pNormal[i].x = vSum.x/share;
        pNormal[i].y = vSum.y/share;
        pNormal[i].z = vSum.z/share;
    }
    delete[] pTempNormal;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราได้ Vertex Normal มาแล้วเราก็จะนำ Vertex Normal ใส่ลงไปบันทึกจุด Vertex ของ Primitive ที่เราได้กำหนดโครงสร้างข้อมูลไว้ในก่อนหน้านี้ จากนั้นจึงทำการสร้าง Vertex Buffer และบรรจุ Vertex ที่มี Vertex Normal แล้วลงไป Buffer ต่อไป

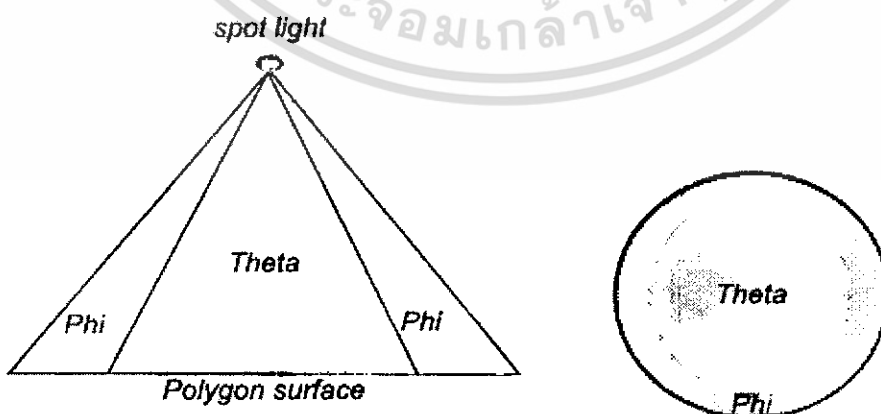
### แหล่งแสงใน DirectX SDK

แหล่งแสงใน DirectX SDK (Direct3D) มีหลายชนิดทั้ง ambient, directional, point และ spot light แหล่งแสงแต่ละประเภทยังมีรูปแบบและการทำงานแตกต่างกัน เริ่มจาก ambient light ซึ่งเป็นแหล่งแสงที่เป็น background light ในฉาก 3 มิติ นั้นหมายความว่าวัตถุ 3 มิติทุกๆ วัตถุจะได้รับแสงนี้ แหล่งแสงประเภทนี้จะไม่มีตำแหน่งใน world space เหมือนแหล่งแสงในประเภทอื่นๆ และเป็นแหล่งแสงที่ใช้พลังในการประมวลผลน้อยที่สุดที่กำหนด ambient light ทำได้โดยใช้เมธอด SetRenderState ดังตัวอย่างเมื่อ r g และ b เป็นค่าแดง สีเขียว น้ำเงิน ตามลำดับซึ่งมีค่าอยู่ในช่วง 0- 255

```
gD3dDevice->SetRenderState( D3DRS_AMBIENT, D3DCOLOR_XRGB(r,g,b) );
```

point light เป็นแหล่งแสงที่เป็นจุด (ชื่อก็บอกอยู่แล้ว) ที่มีตำแหน่งใน world space ลักษณะแสงของ point light จะเป็นแสงที่เปล่งออกมารอบตัวในทุกๆ ด้าน แหล่งแสงประเภทนี้จะ เป็นแหล่งแสงที่ใช้พลังในการประมวลผลค่อนข้างมาก แต่ก็ให้ผลลัพธ์ที่น่าพอใจ

spot light เป็นแหล่งแสงที่เปล่งออกมาในลักษณะเป็นรูปกรวย (ตามรูปครับ) โดยรูปกรวยนั้นจะมีสองส่วนคือ กรวยชั้นใน (Inner Cone) มีความสว่างกว่ากรวยชั้นนอก (Outer Cone) โดยองศาความกว้างของกรวยชั้นในนั้นเราจะเรียกว่า Theta และ องศาความกว้าง ของกรวยชั้นนอกซึ่งมีความกว้างมากกว่านั้นเราจะเรียกว่า Phi แหล่งแสงประเภทนี้จะ เป็นแหล่งแสงที่ใช้ พลังในการประมวลผลค่อนข้างมากเช่นกัน



รูปที่ 2 แสดงการฉายแสงของ Spot light

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Directional light เป็นแหล่งแสงที่มีลักษณะคล้ายๆ กับแสงจากดวงอาทิตย์ (ที่เปล่งออกมา) จะมาในลักษณะทางเดียวโดยตรง ซึ่งจะตรงกันข้ามกับ point light แสงชนิดนี้ไม่มีจุดสิ้นสุด และเป็นแหล่งแสงที่ใช้พลังในการประมวลผลน้อยกว่า point light และ spot light

### การกำหนด Material ให้แก่วัตถุ 3 มิติ

ก่อนที่จะสร้างแหล่งแสงให้แกโลก 3 มิติจะต้องมากำหนด Materials ให้แก่วัตถุ 3 มิติก่อน ถ้าหากใครเคยใช้โปรแกรม 3D Studio Max หรือโปรแกรม 3 มิติอื่นๆ ก็น่าจะเข้าใจหลักการของการกำหนด Materials ดี สำหรับ DirectX SDK การกำหนด Materials เป็นการกำหนดรูปแบบและคุณสมบัติของวัตถุ 3 มิติที่จะเกิดขึ้นเมื่อได้รับแสงมาตกกระทบบนพื้นผิว เช่น ถ้าหากวัตถุเป็นเหล็กหรือพลาสติกก็ย่อมจะมีความมันวาวเมื่อได้รับแสงซึ่งจะต้องตรงกันข้ามกับก้อนหินหรือไม้ที่จะไม่มีความมันวาวเมื่อได้รับแสง เป็นต้น ประโยชน์ของการกำหนด Materials ให้แก่วัตถุ 3 มิติคือจะทำให้รูปแบบของวัตถุ 3 มิติมีความหลากหลายและดูสมจริงมากยิ่งขึ้น การกำหนด Materials ให้แก่วัตถุ 3 มิตินั้นสามารถทำได้โดยใช้เมธอดของ Direct3DDevice

```
HRESULT SetMaterial(CONST D3DMATERIAL9 * pMaterial);
```

pMaterial เป็นพอยเตอร์ที่ชี้ไปยังโครงสร้างข้อมูล D3DMATERIAL9 ที่เราจะต้อง

กำหนดเข้าไปโดยโครงสร้างภายในของ D3DMATERIAL9 เป็นดังนี้

```
struct{
    D3DCOLORVALUE Diffuse;
    D3DCOLORVALUE Ambient;
    D3DCOLORVALUE Specular;
    D3DCOLORVALUE Emissive;
    float Power;
}
```

สำหรับโครงสร้างภายในของ D3DCOLORVALUE เป็นดังนี้

```
struct{
    float r;
    float g;
    float b;
    float a;
}
```

โดย Diffuse เป็นสีหลักที่จะต้องกำหนดให้กับ Materials เนื่องจากวัตถุ 3 มิติที่ได้จะมีสีตามสีที่กำหนดใน Diffuse ซึ่งค่าสี Diffuse ใน D3DMATERIAL9 จะมีความสัมพันธ์กับ Diffuse ใน D3DLight9 ต่อมาคือ Ambient เป็นสีที่มีความสัมพันธ์กับ ambient light ส่วน Specular เป็นสีที่สัมพันธ์กับ specular highlight ถ้าหากใครเคยเล่น DOOM 3 แล้วลองสังเกตผนังโลหะของสถานีอวกาศดูจะเห็นว่ามีความสว่างของพื้นผิวโลหะเมื่อเรานำไฟฉายไปส่องใกล้ๆ จะเป็น specular highlight และสีสุดท้ายคือ Emissive เป็นค่าสีที่วัตถุ 3 มิติเปล่งออกมาเมื่อได้รับแสง ซึ่งยังมีค่ามากการเปล่งแสงกลับออกมาก็จะยิ่งมาก วัตถุก็จะยิ่งมีความแวววาวมากขึ้น และสุดท้ายคือ Power เป็นปริมาณค่า specular ที่มีบนวัตถุ 3 มิติ

สำหรับค่า Diffuse ambient specular และ Emissive จะอยู่ในช่วง 0.0 - 1.0 การกำหนด Materials ให้กับวัตถุ 3 มิติที่คุณสามารถจะกำหนดให้มีความแตกต่างกันได้วิธีการกำหนดก็ง่ายๆ

```
gD3dDevice->SetMaterial( &m_material แรก);
```

Draw วัตถุ 3 มิติแรก

```
gD3dDevice->SetMaterial( &m_material ที่ 2);
```

Draw วัตถุ 3 มิติที่ 2

```
.....
```

```
gD3dDevice->SetMaterial( &m_material ที่ n);
```

Draw วัตถุ 3 มิติที่ n

## การสร้างแหล่งแสง

เมื่อเรากำหนด Vertex Normal และ Materials ให้กับวัตถุ 3 มิติแล้วเราก็พร้อมที่จะสร้างแหล่งแสงกันแล้ว โดยในการสร้างแหล่งแสงนั้นเราจะใช้เมธอด SetLight ในการกำหนดรูปแบบและคุณสมบัติของแสงนั้นๆ จากนั้นจึงบรรจุแหล่งแสงนั้นลงไปในโลก 3 มิติโดยใช้เมธอด LightEnable และอย่าลืมสั่งให้ DirectX ทำการคำนวณค่าแสงตกกระทบบนวัตถุ 3 มิติด้วยเมธอด SetRenderState โดยกำหนด State ให้เป็น D3DRS\_LIGHTING และ value ให้เป็น TRUE

```
gD3dDevice->SetRenderState( D3DRS_LIGHTING, TRUE );
HRESULT SetLight(
    DWORD Index,
    CONST D3DLight9 * pLight
);
```

Index เป็นหมายเลขของแหล่งแสงที่จะใช้ ค่าจะเริ่มจาก 0 ถึงจำนวนแหล่งแสงที่มากที่สุดที่ Device จะสามารถสร้างได้

pLight เป็นพอยเตอร์ที่ชี้ไปยังโครงสร้างข้อมูล D3DLight9 ที่เราจะต้องกำหนดเข้าไป โดยโครงสร้างภายในของ D3DLight9 เป็นดังนี้

```
struct {
    D3DLIGHTTYPE Type;
    D3DCOLORVALUE Diffuse;
    D3DCOLORVALUE Specula;
    D3DCOLORVALUE Ambient;
    D3DVECTOR Position;
    D3DVECTOR Direction;
    float Range;
    float Falloff;
    float Attenuation0;
    float Attenuation1;
    float Attenuation2;
    float Theta;
    float Phi;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Type เป็นชนิดของแสงที่เราจะใช้

point light ให้กำหนดเป็น D3DLIGHT\_POINT

spot light ให้กำหนดเป็น D3DLIGHT\_SPOT

directional light ให้กำหนดเป็น D3DLIGHT\_DIRECTIONAL

Diffuse Specular และ Ambient เป็นปริมาณค่าสีดังกล่าวที่เปล่งออกมาจากแหล่งแสงนี้

Position คือตำแหน่งของแสงใน world space หากกำหนดชนิดของแสงเป็น directional light จะไม่จำเป็นต้องกำหนด

Direction คือตำแหน่งใน world space ที่แสงจะส่องไปหากกำหนดชนิดของแสงเป็น point light จะไม่จำเป็นต้องกำหนด

Range คือระยะที่แสงส่องไปถึง

Falloff เป็นปริมาณการส่องสว่างระหว่างกรวยชั้นนอกกับกรวยชั้นในของ spot light

Theta เป็นองศาความกว้างของ กรวยชั้นในของ spot light มีค่าเป็น radians

phi เป็นองศาความกว้างของ กรวยชั้นนอกของ spot light มีค่าเป็น radians

```
HRESULT LightEnable(
    DWORD LightIndex,
    BOOL bEnable
);
```

LightIndex เป็นหมายเลขของแหล่งแสงที่จะใช้

bEnable กำหนดให้เป็น TRUE หากต้องการใช้งานและ FALSE ถ้าเลิกใช้งาน

ตัวอย่างข้างล่างนี้เป็นการกำหนดแสงแบบ spot light, point light และ directional light ตามลำดับ

```
*****spot light*****
D3DLIGHT9 light;
ZeroMemory( &light, sizeof(D3DLIGHT9) );
light.Type = D3DLIGHT_SPOT;
light.Diffuse.r = 0.0f;
light.Diffuse.g = 0.0f;
light.Diffuse.b = 0.5f;
light.Diffuse.a = 1.0f;
light.Position = position
light.Direction = direction
light.Range=1.0f;
light.Theta=0.5f;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

light.Phi=1.0f
light.Falloff=1.0f;
light.Attenuation0 = 1.0f;
gD3dDevice->SetLight( 0, &light );
gD3dDevice->LightEnable( 0, TRUE );

//*****point light *****
D3DLIGHT9 light;
ZeroMemory( &light, sizeof(D3DLIGHT9) );
light.Type = D3DLIGHT_POINT;
light.Diffuse.r = 0.0f;
light.Diffuse.g = 0.0f;
light.Diffuse.b = 0.5f;
light.Diffuse.a = 1.0f;
light.Position = position
light.Attenuation0 = 0.1f;
light.Range = 200.0f
gD3dDevice->SetLight( 1, &light );
gD3dDevice->LightEnable( 1, TRUE );

//*****directional light *****
ZeroMemory( &light, sizeof(D3DLIGHT9) );
light.Type = D3DLIGHT_DIRECTIONAL;
light.Diffuse.r = 0.0f;
light.Diffuse.g = 0.0f;
light.Diffuse.b = 0.5f;
light.Diffuse.a = 1.0f
light.Range = 1000.0f;
D3DXVECTOR3 vecDir;
D3DXVec3Normalize(&vecDir,& direction);
light.Direction = vecDir;
gD3dDevice->SetLight( 2, &light );
gD3dDevice->LightEnable( 2, TRUE );

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### จำนวนแหล่งแสงในฉาก 3 มิติ

จำนวนแหล่งแสงที่ใช้นั้นมีจำนวนจำกัดไม่สามารถกำหนดจำนวนแหล่งแสงได้ตามอำเภอใจ เราสามารถตรวจสอบจำนวนแหล่งแสงสูงสุดที่ใช้ได้ของ Device (การ์ดจอที่ใช้รันโปรแกรม) นั้นๆ ได้โดยใช้เมธอด GetDeviceCaps ซึ่งจำนวนแหล่งแสงมากที่สุดที่สามารถกำหนดได้จะอยู่ในตัวแปร MaxActiveLights ของโครงสร้างข้อมูล D3DCAPS9 ครับ

### Dynamic Light

ถ้าหากสังเกตการสร้างแหล่งแสงใน DirectX นั้นจะเห็นว่าจะต้องกำหนดสี ชนิด และ ตำแหน่ง รวมทั้งพารามิเตอร์ปลีกย่อยต่างๆ (ตามแต่ชนิดของแหล่งแสง) ของแสงก่อนถึงจะสร้างแสงได้ ทำให้แหล่งแสงที่สร้างขึ้นเป็นแสงที่ไม่มีเปลี่ยนแปลง แต่ถ้าหากสังเกตเกมบางเกมจะเห็นได้ว่าแหล่งแสงของเขาสามารถเปลี่ยนทั้งตำแหน่งและสีได้ในทุกๆ เวลา ซึ่งแสงแบบนี้เราจะเรียกว่า Dynamic Light

### 2.3 Shading

คือการทำรายละเอียดของตัวพื้นผิว เช่น ความมันวาว การสะท้อนของพื้นผิว หรือความโปร่งแสง ทึบแสงของวัตถุ เป็นการให้สีเป็นลำดับชั้น

- Flat Shading Lighting เป็นการรายละเอียดพื้นผิวที่มีสีเสมอกันทั่วทั้ง polygon
- Vertex Shading หรือ Gouraud Shading เป็นการให้สีแก่ vertex แต่ละจุดตามสีที่ได้กำหนดเอาไว้แล้ว

## 2.4 Texture

คือลวดลายของพื้นผิว โดยที่จะเป็น Bitmap ที่เป็น Pattern หรือ Image มักจะเก็บในรูปแบบไฟล์ BMP, PCX หรือ GIF เพื่อเป็นการใส่รายละเอียดให้แก่พื้นผิวของวัตถุ ทำให้วัตถุมีความสมจริง

- Texture Coordinate ใช้กำหนดการเชื่อมต่อกันระหว่าง Vertices ของ Face กับ Pixel ของ Bitmap โดย Texture Coordinate นี้ใช้แทน 2 มิติของ Coordinate System
- Texture Mapping คือ การวาดรูปลงบนพื้นผิวของ Face หรือ Polygon และในการทำ Texture Mapping นี้ต้องคำนึงถึงการคำนวณค่าต่างๆด้วย จึงต้องมีการกำหนดค่าของ Vertices ด้วย จากการที่ Texture Coordinate กำหนด Pixel ของ Texture ที่จะวาดลงใน ส่วนของ Face แล้วก็จะมีการ Wrapping เพื่อ Generate Texture Coordinate สำหรับ Object นั้นซึ่งการ Wrapping นั้นมี 5 ชนิดคือ

1. Flat Warp จะทำการวาดลงบน Face โดยตรง
2. Cubical Warp จะทำการ Warp Texture ใน Cube รอบๆ Object เหมือนกับการแปะลวดลายบนวัตถุที่มีลักษณะเป็นกล่อง
3. Cylindrical Warp จะทำการ Warp Texture ใน Cyline รอบๆ Object คือจะนำลวดลายในการ Map รอบโมเดลในลักษณะการห่อเหมือนทรงกระบอก ซึ่งเหมาะกับวัตถุที่มีลักษณะเป็นทรงกระบอกหรือเป็นแท่งๆ
4. Spherical Warp จะทำการ Warp Texture ใน Sphere Object ซึ่งจะนำภาพมาแปะในลักษณะห่อรอบทรงกลม เหมาะกับการแปะลวดลายบนวัตถุทรงกลม
5. Shrink Wrap คล้ายกับการ Map แบบ Spherical แต่โปรแกรมจะรวมจุดปลายของลวดลายที่ นำมา Map เข้าหากันเป็นจุดเดียว

## 2.5 Camera

Camera คือ มุมกล้องที่ทำให้เราสามารถมองเห็นวัตถุจากทิศทางต่างๆ เปรียบเสมือนว่าเรากำลังนำกล้องไปตั้งไว้บริเวณพื้นที่ที่มีวัตถุดังกล่าว ดังนั้นการเปลี่ยนมุมกล้องก็จะทำให้ภาพที่ปรากฏบนจอมีลักษณะแตกต่างกันไป โดยมุมกล้องจะถูกกำหนดโดย Viewport ซึ่งมีค่าต่างๆอยู่ 3 ค่าด้วยกันดังนี้

1. Eyepoint เปรียบเสมือนเป็นจุดที่กล้องตั้งอยู่
2. Lookat เป็นจุดที่กล้องมองไป

Upvector คือ Vector ที่ชี้บอกทิศด้านบนของกล้อง

## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

### 2.6 DirectX

Direct X เป็น API ของ Microsoft พัฒนาเพื่อใช้จัดเตรียม Interface สำหรับควบคุม ฮาร์ดแวร์มัลติมีเดียบนระบบ Microsoft Windows ได้อย่างมีประสิทธิภาพ และเป็นเครื่องมือให้ โปรแกรมเมอร์ทำงานกับคำสั่งและโครงสร้างข้อมูลในระดับใกล้ฮาร์ดแวร์ โดยไม่ต้องสร้างโค้ด ติดต่อกับระดับล่างซึ่งวิธีติดต่อกันจะแตกต่างกันไปตามประเภทของอุปกรณ์ การเขียนโค้ดที่เป็นอิสระ จากอุปกรณ์ในลักษณะนี้ ช่วยให้โปรแกรมเมอร์สามารถสร้างซอฟต์แวร์เพื่อทำงานดังกล่าวได้เป็น อย่างดี แม้ผู้ใช้จะปรับเปลี่ยนอุปกรณ์ตัวใหม่และเพิ่มการ์ดเร่งความเร็วแบบสามมิติ, เสียง, อุปกรณ์ อินพุตและอื่นๆก็ตาม

Microsoft direct X SDK (Software Development Kit) นั้นเป็นเครื่องมือที่ใช้ในการ พัฒนาซอฟต์แวร์ทางมัลติมีเดียของบริษัท Microsoft ซึ่งรวมถึงสามารถพัฒนาเกมคอมพิวเตอร์ได้ อีกด้วยโดยเริ่มต้นเราจะเห็นเฉพาะเฮดเดอร์ไฟล์และไลบรารีไฟล์ของ Direct Draw เท่านั้น หลังจากนั้น ต่อมา งานด้าน 3 มิติ เริ่มเข้ามามีบทบาทมากขึ้น ทางบริษัท Microsoft จึงได้ทำการติดต่อกับ ทางบริษัทอังกฤษที่ชื่อว่า Reality Lab เพื่อขอซื้อ Rendermorphics มาจัดการในงานกราฟิก 3 มิติ ไว้ใน Microsoft DirectX SDK และพัฒนาเวอร์ชันใหม่ขึ้นเรื่อยๆ อันได้แก่ Microsoft DirectX SDK เวอร์ชัน 3, เวอร์ชัน 5, เวอร์ชัน 6, เวอร์ชัน 7, เวอร์ชัน 8 และปัจจุบันนี้อยู่ใน เวอร์ชัน 9 ทำให้มี Interface ใหม่ ๆ ขึ้นมา รวมถึงการเพิ่ม Method บางอย่างเข้าไปอีก เพื่อตอบสนองการใช้งาน ให้มากที่สุดนั่นเอง

72968

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

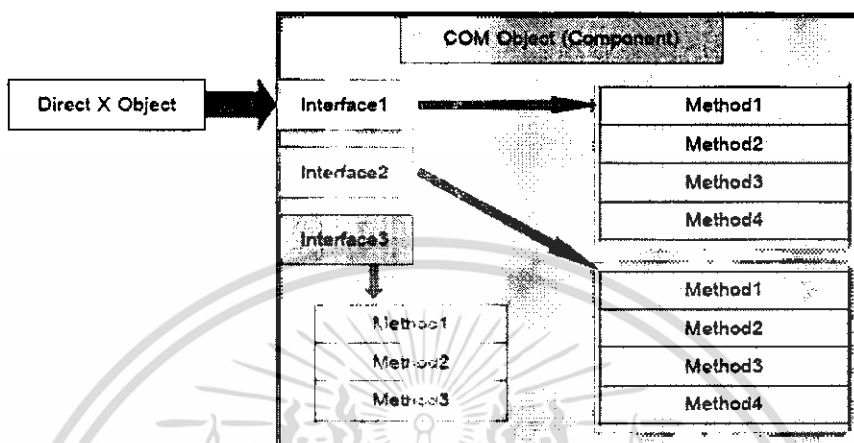
## Component Object Model (COM)

COM เป็นวิธีที่ Microsoft คิดขึ้นมาเพื่อให้ Object ต่างๆ ที่อาจจะไม่มีความเกี่ยวข้องกันสามารถติดต่อสื่อสารแลกเปลี่ยนข้อมูลกันได้ โดยไม่ขึ้นกับภาษาที่ใช้เขียน, ไม่ขึ้นกับ OS, ไม่ขึ้นกับระบบของเครื่องคอมพิวเตอร์ Component คือการรวมเอา Object ย่อยๆ ต่างๆ ที่มีหน้าที่การทำงานในแบบเดียวกัน มารวมกันเป็น Component โดยเรียก Object ย่อยๆ เหล่านั้นใหม่ว่า Interface

Interface ของ COM คล้ายคลึงกับ Abstract Class ของภาษา C++ คือมีแต่ส่วนที่เป็นสัญลักษณ์ ความหมาย ข้อกำหนดความสามารถที่จะทำ แต่จะไม่มีโค้ดที่ใช้ทำงานแต่อย่างใด ทั้ง Interface ของ COM และฟังก์ชันแบบวิซวลของ C++ ทั้งสองใช้ตัวไวย่ตัวหนึ่งที่มีชื่อว่า vtable สำหรับพักแอดเดรสของฟังก์ชันหรือ Method สำหรับนำไปใช้ในการทำ Interface การที่ object จะสามารถใช้ Method ที่เป็นของ Interface ของ COM ได้ จะต้องเรียกใช้ Method QueryInterface เพื่อตรวจสอบว่า Interface ที่ต้องการใช้มีอยู่ใน Object หรือไม่ แล้วรับพอยเตอร์ของ Interface ดังกล่าวกลับมาใช้งานที่ 3591 . พอยเตอร์ที่ได้มาจากการติดต่อ vtable ด้วย Method QueryInterface5 ใน DirectX เวอร์ชัน 9.0 จะประกอบด้วย Component 8 ตัว คือ DirectX Graphics, DirectInput, DirectPlay, DirectSound, DirectMusic, DirectShow, DirectSetup, DirectX Media Objects ซึ่งจะไม่มีการมี Component ที่ชื่อ DirectDraw, Direct3D โดยจะรวมเอา DirectDraw และ Direct3D ไปเป็น Interface อันเดียวคือ Direct3D Interface ซึ่งอยู่ใน DirectX Graphics Component

## DirectX Object, COM Object, Interface และ Method

Interface แต่ละอันก็จะมี Method หรือฟังก์ชันการทำงานต่างๆ การจะเรียกใช้ Method นั้นต้องสร้าง DirectX Object ที่ชี้ไปยัง Interface ที่เราต้องการ โดยการสร้าง DirectX Object นี้สามารถทำได้โดยการเรียกใช้ Function หรือ Method ที่ DirectX เตรียมไว้ให้



รูปที่ 3 แสดงการเรียกใช้งานของ DirectX

จากรูป เปรียบเสมือนกล่องที่มีรูหรือช่องให้เสียบหลายๆ รู แต่ละรูก็เปรียบเสมือน Interface โดยแต่ละ Interface ก็มี Method ซึ่งก็คือหน้าที่การทำงานแบบต่างๆ เมื่อเราเปลี่ยนให้ DirectX Object เข้าชี้ไปที่ Interface ใด เราก็จะสามารถเรียกใช้งาน Method ใน Interface นั้นได้ ถ้าจะใช้ Method อื่นๆ ที่ไม่ได้อยู่ใน Interface ที่ DirectX Object ชี้อยู่ ก็ให้หา DirectX Object มาเพิ่มเพื่อจะเรียกใช้ Interface ได้และถ้าต้องการยกเลิกการใช้ Interface ใด เราก็ต้องถอดตัวชี้ ออก ข้อกำหนดหนึ่งของระบบ COM ก็คือ ถ้ามีการเผยแพร่ หรือแจกจ่าย COM ให้คนอื่นนำไปใช้แล้ว (เช่น ในกรณีที่ Microsoft ออก DirectX เวอร์ชันล่าสุดให้พวกเราได้ดาวน์โหลดมาใช้กัน) ถ้าหากในภายหลังต้องการจะแก้ไข Interface บางอันให้ทำงานได้ดียิ่งขึ้น จะต้องห้ามเข้าไปแก้ไขใน Interface เดิมให้ใช้วิธีสร้าง Interface ใหม่เพิ่มขึ้นมาแทน (โดยอาจจะสืบทอดมาจาก Interface เดิม) เพราะถ้าไปแก้ไขอาจจะทำให้โปรแกรมเก่าๆ ที่เขียนด้วย DirectX SDK เวอร์ชันเก่า ซึ่งมีการเรียกใช้ Interface อันเดิมใช้งานไม่ได้

นอกจากนี้ยังต้องตั้งชื่อ Interface ที่สร้างใหม่ไม่ให้ซ้ำกับของเดิมด้วย จุดนี้เองเราจึงเห็นชื่อ Interface ใน DirectX ใช้วิธีใส่ตัวเลขเวอร์ชันกำกับไว้ข้างท้าย เช่น IDirect3D8, IDirect3D9 เป็นต้น DirectX จะใช้ตัวอักษร I นำหน้าชื่อ Interface) ดังนั้น ใน Component ของ DirectX เวอร์ชันใหม่ๆ ก็จะมี Interface ของเวอร์ชันเก่าๆ รวมอยู่ด้วย จึงทำให้เกมที่สร้างด้วย DirectX เวอร์ชันก่อนหน้าสามารถรันบน DirectX เวอร์ชันใหม่ๆ ได้ แต่ผลลัพธ์ของเกม เช่น Effect ต่างๆ หรือแม้กระทั่งความเร็วของเกม ก็ยังคงใกล้เคียงกับที่เคยรันด้วย DirectX เวอร์ชันเดิม เพราะ Interface มันเก่าก็ยังไม่เปลี่ยนแปลง Method ต่างๆ ก็ไม่เปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## API ของ DirectX 9.0 SDK

DirectX 9.0 SDK ถูกพัฒนาให้มีความสามารถเพิ่มขึ้นจาก เวอร์ชันก่อนๆ โดยสามารถแบ่งออกเป็นส่วนประกอบหลักๆได้ดังนี้

1 DirectX Graphics เป็นการรวมกันระหว่าง DirectDraw และ Direct 3D โดยมีจุดเด่นตรงที่การประมวลผล กราฟิกแบบขนาน และสามารถจัดการกับกราฟิกให้มีความสมจริงกว่า เวอร์ชันที่ผ่านมาทำให้ผู้พัฒนาเกม สามารถพัฒนาเกม 3มิติ ได้ง่ายขึ้น แต่สำหรับการพัฒนาเกม 2 มิติ กลับทำได้ยุ่งยากขึ้น

2 DirectInput เป็นอีกองค์ประกอบหนึ่งของ Direct X ที่ช่วยจัดการในส่วนของการจัดการอุปกรณ์แป้นพิมพ์ (Keyboard), เมาส์ (Mouse) และ จอยสติ๊ก (Joy Stick) ซึ่งเป็นส่วนที่ช่วยให้โปรแกรมเมอร์ สามารถที่จะติดต่อกับอุปกรณ์ดังที่ได้กล่าวมาได้โดยง่ายโดยในการเขียนโปรแกรมติดต่อกับอุปกรณ์ Keyboard หรือ Mouse ก็จะมีขั้นตอนของการเขียนโปรแกรมดังนี้ คือ

- สร้าง Keyboard หรือ Mouse Device
- Set Cooperation level ให้กับ Keyboard เพื่อกำหนดรูปแบบการใช้งานอุปกรณ์
- Set รูปแบบข้อมูลให้กับ Keyboard Device หรือ Mouse Device
- Acquire Keyboard Device หรือ Mouse Device

ในเวอร์ชันนี้ได้มีการเพิ่มประสิทธิภาพใหม่เข้าไป คือสามารถใช้ Action Mapping ได้ซึ่งใช้กำหนดค่าต่างๆให้กับอุปกรณ์ควบคุมและยังสามารถจัดการเชื่อมต่อระหว่าง InputAction กับ อุปกรณ์ควบคุม(Input Device)ได้และการเปลี่ยนแปลงค่าทำได้ง่ายขึ้น โดยจะขึ้นอยู่กับชนิดของ อุปกรณ์ควบคุม(Input Device) นั้นๆ นอกจากนี้ยังได้เพิ่มการสนับสนุนอุปกรณ์ต่างๆได้มากขึ้น

3 DirectPlay ถูกเพิ่มประสิทธิภาพในการสนับสนุนการทำงานในรูปแบบเน็ตเวิร์กหรือระบบเครือข่ายให้มีความหลายหลายในการใช้งาน อีกทั้งยังเพิ่มการป้องกันความผิดพลาดที่อาจเกิดขึ้นได้กับระบบนอกจากนี้DirectPlayยังเพิ่มการสนับสนุนการสื่อสารด้วยเสียงระหว่างผู้เล่น (Voice Communication)

4 DirectSound เป็นชุด API ที่มีเครื่องมือในการใช้จัดการกับระบบเสียงแบบสเตอริโอและแบบ 3 มิติ ได้อย่างมีประสิทธิภาพประกอบด้วยความสามารถในการจัดการหน่วยความจำ การผสมเสียงของฮาร์ดแวร์ DirectSound ได้รับการออกแบบมาเพื่อดึงความสามารถของฮาร์ดแวร์ระบบ การรวมเสียงแบบ 3 มิติเข้าไปในเกมหรือการซิมูเลชัน ทำให้แอปพลิเคชันสามารถให้เสียงได้สมจริงสมจัง การได้ยินเสียงจากทางซ้าย ทางขวา หรือจากทางด้านบน โดยเฉพาะอย่างยิ่งเสียงที่มาจากรอบ ๆ ตัวจะช่วยสร้างสภาพแวดล้อมที่ตื่นตาตื่นใจมากขึ้น

5 DirectMusic เป็นชุด API ที่ทำงานกับข้อมูลประเภท Message-base Musical Data ซึ่งเป็นข้อมูลที่แปลงมาจาก Wave Sample ด้วยซินธิไซเซอร์ (Synthesizer) ทั้งแบบฮาร์ดแวร์หรือซอฟต์แวร์ โดยปกติถ้าเป็นซอฟต์แวร์จะใช้โปรแกรม Microsoft Software Synthesizer เพื่อสร้าง Wave Sample ให้กับ DirectSoundการใช้เสียงดนตรีที่ได้จากซินธิไซเซอร์จะเป็นไปตามมาตรฐาน DLS (Downloadable Sound) นอกจากนี้ DirectMusic ยังมีกลไกที่ใช้สร้างเพลงได้ตามที่กำหนด

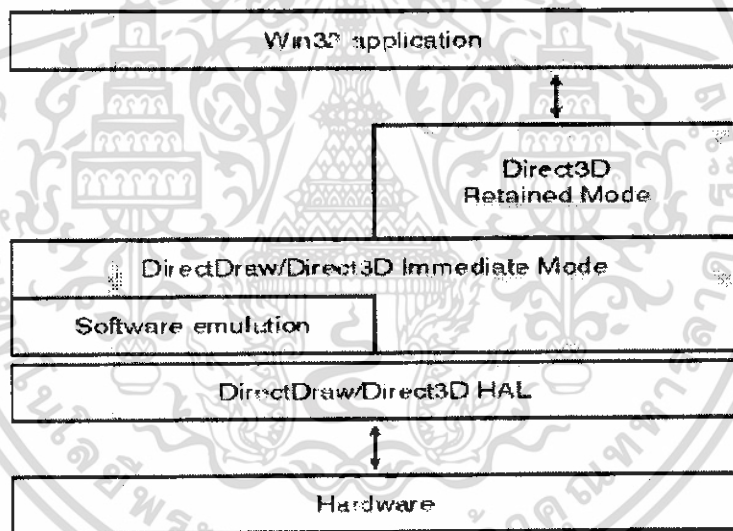
6 DirectShow ได้ถูกเพิ่มเข้ามาตั้งแต่ Version DirectX 8.0 นี้โดยในเวอร์ชันก่อนหน้า DirectShow จะถูกแยกต่างหากจาก DirectX โดยจะทำหน้าที่ในการเล่นไฟล์มัลติมีเดีย ประเภท ภาพวิดีโอเป็นหลัก แต่เมื่อรวมเข้ากับ DirectX แล้ว ทำให้การเล่นไฟล์มัลติมีเดียประเภทภาพวิดีโอสามารถควบคุมได้ง่ายและมีประสิทธิภาพมากขึ้น

7 DirectSetup API ชุดนี้จะทำหน้าที่เกี่ยวกับการติดตั้งคอมโพเนนต์ DirectX Runtime ให้กับระบบโดยอัตโนมัติถ้าหากระบบดังกล่าวยังไม่ได้รับการติดตั้ง แม้ในปัจจุบันมักจะมีการติดตั้งคอมโพเนนต์มากับระบบปฏิบัติการแล้วก็ตาม แต่สำหรับ DirectX แล้วไม่มีใครซอฟต์แวร์ได้อนุญาตให้ใช้โดยที่ไม่ต้องเสียค่าใช้จ่าย จึงไม่ต้องกังวลเรื่องของลิขสิทธิ์ โดยสามารถทำการติดตั้งที่ครั้งก็ได้

8 Direct Media แม้ว่าโปรแกรมเมอร์สามารถเขียนโปรแกรมติดต่อกับ ActiveX Control โดยตรงได้โดยใช้ฟังก์ชันของ COM Interface แต่มันก็ออกจะยุ่งยากไปบ้าง ดังนั้นเพื่อความง่ายในการใช้งาน โปรแกรมเมอร์สามารถเลือกใช้ DirectX Media SDK ซึ่งมีคลาสของ MFC ที่มีชื่อว่า CMediaPlayer ให้โปรแกรมเลือกใช้แทนที่จะเรียก COM Interface โดยตรง ทำให้การใช้งาน Media Player Control ได้ง่ายมากขึ้น

### การทำงานของ Direct3D

Direct3D มีขึ้นครั้งแรกใน DirectX 2.0 ประมาณช่วงปี ค.ศ. 1996 หลังจากนั้นภายในระยะเวลาอันสั้น Direct3D ได้กลายเป็นชุด API สำหรับการทำงานที่ได้รับความนิยมอย่างสูงในท้องตลาดที่เกมและซิมูเลชันส่วนใหญ่เลือกใช้ รวมทั้งได้รับการสนับสนุนจากผลิตภัณฑ์เร่งความเร็ว 3 มิติ ปัจจุบันฮาร์ดแวร์เร่งความเร็วกราฟิกพัฒนาไปจนสามารถแสดงโพลีก่อนได้หลายล้านโพลีก่อนต่อวินาที ฮาร์ดแวร์เร่งความเร็วที่นำมาเชื่อมการทำงานกับ Direct3D มีความเร็วมากขึ้น และช่วยลดต้นทุนของแอปพลิเคชัน 3 มิติ ทำให้ไม่ต้องทุ่มทุนสูงนักเพื่อแลกกับประสิทธิภาพในการทำงานที่สูงขึ้น เช่นเดียวกับ API ทั้งหมดของ DirectX ชุด API ของ Direct3D ไม่ได้ออกแบบมาให้เพียง API สำหรับการพัฒนาเกมเพื่อรองรับความสามารถที่จะมีขึ้นในอนาคตเท่านั้น แต่ยังได้รับการออกแบบมาให้มีความเข้ากันได้ทั้งหมดกับเวอร์ชันที่ผ่านมา ดังนั้นเกมที่พัฒนาด้วย DirectX เวอร์ชันเก่าก็จะสามารถรันได้ใน DirectX เวอร์ชันที่จะมีขึ้นในอนาคต



รูปที่ 4 แสดงการติดต่อจากระดับ Application ไปถึงระดับ hardware โดยผ่าน DirectX

Direct3D ในยุคแรกมี API อยู่สองโหมดคือ โหมด Immediate(IM) และโหมด Retained(RM) ในโหมด IM (การเรนเดอร์อ็อบเจกต์สามารถทำได้โดยจับพลัน ได้ตามความต้องการของโปรแกรมเมอร์) เป็นโหมดที่ใช้งานยากแต่จะมีความยืดหยุ่นสูง เป็น API ในระดับล่างสำหรับใช้เขียนเกมที่ทำงานได้เร็วและมีประสิทธิภาพเท่าที่เครื่องจะเป็นไปได้บนระบบ ในโหมด RM (API จะเก็บขึ้นลงในฐานข้อมูลแล้วนำมาเรนเดอร์ทั้งหมดในคราวเดียวกัน) เป็นโหมดที่สร้างขึ้นมาเป็นเลเยอร์ที่อยู่บนสุดของโหมด IM โดยในโหมดนี้จะจัดเตรียมบริการต่าง ๆ เช่น การจัดการ Texture, การโหลด Object File, การจัดลำดับเฟรม และการทำ อ็อบเจกต์เคลื่อนไหว การศึกษา และการใช้งานโหมด RM นั้นจะง่ายกว่าเมื่อเทียบกับการใช้งานในโหมด IM แต่ถ้าหากโปรแกรมเมอร์คนใดต้องการประสิทธิภาพและความยืดหยุ่นในการทำงานก็มักจะเลือกใช้การทำงานในโหมด IM ดังนั้นการทำงานในโหมด RM จึงได้หยุดพัฒนาลงใน Direct X version 6.0 และหันไปมุ่งพัฒนาโหมด IM ให้มีความสามารถและใช้งานง่ายในเวอร์ชันต่อมา ด้วยเหตุนี้การทำงานในโหมด RM จึงไม่สนับสนุนเทคโนโลยีใหม่ เช่น Multitexturing, Bump Mapping, Hardware Transformation และ Lighting ดังนั้นความสามารถทั้งหมดของโปรแกรม 3 มิติ ควรเขียนขึ้นมาโดยใช้โหมด IM และในการทำงานร่วมกับฮาร์ดแวร์เร่งความเร็วกราฟิกของแอปพลิเคชัน Direct3D จะใช้การติดต่อกับฮาร์ดแวร์ผ่านทาง HAL (Hardware Abstraction Layer) แต่ถ้าในพีเจอนั้นไม่มีในฮาร์ดแวร์ Direct3D จะใช้ HEL (Hardware Emulation Layer) แทน

## 2.7 X File

เป็น File ประเภทหนึ่งโดยมีนามสกุลเป็น .x ซึ่งใช้ในการเก็บข้อมูลต่างๆ ของ Mesh ตั้งแต่จุด Vertices สีของ Vertices รายละเอียดการ Texture รวมไปถึง Frame Animation ด้วย

โดยใน X File นั้นจะมีการประกาศ Template ต่างๆ เพื่อใช้ในการระบุ Data type ของข้อมูล โดยมีรูปแบบดังนี้

```
template <template-name> {
  <GUID>
  <member 1>;
  ...
  <member n>;
  [restrictions]
}
```

ในการประกาศ Template ให้เริ่มด้วย Template ตามด้วยชื่อของ Template ต่อมาทำการกำหนดค่า GUID (Globally unique identifier) ซึ่งเป็นชุดของตัวเลขและตัวอักษร ใช้ในการระบุและแยกแยะ Template ในการ Coding ตามด้วยการประกาศ Member Data Type ต่างๆ ตามจำนวนที่ต้องการ

Type	Size
WORD	16 bits
DWORD	32 bits
FLOAT	IEEE float
DOUBLE	64 bits
CHAR	8 bits
UCHAR	8 bits

BYTE	8 bits
STRING	NULL terminated string

รูปที่ 5 แสดง Data type ที่ใช้ใน X File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง Template

```
template MY_GAME_INFO{
  <AA1308FD-FF98-4f6e-9A55-CD083178672F>
  STRING GameName;
  STRING GameMaker;
  DWORD Version;
}
```

ตัวอย่าง Data Object ที่ใช้ในการเก็บข้อมูล

```
MY_GAME_INFO Game01
{
  "My Test Game";
  "Alan Thorn";
  1;
}
```

นอกจาก Template ที่เราทำการสร้างเองได้แล้วใน DirectX ยังมี Standard Template ที่ถูกประกาศเอาไว้แล้วเตรียมเอาไว้ให้ใช้อยู่แล้วหลาย Template ด้วยกันเช่น

```
template Animation
{
  < 3D82AB4F-62DA-11cf-AB39-0020AF71E433 >
  [...]
}
template AnimationKey
{
  < 10DD46A8-775B-11CF-8F52-0040333594A3 >
  DWORD keyType;
  DWORD nKeys;
  array TimedFloatKeys keys[nKeys];
}
```

เป็นต้น

## 2.8 Data structure

เนื่องจากในตัวยุคเกมต้องมีการใช้และควบคุมจัดการกับข้อมูลต่างๆเป็นจำนวนมาก จึงจำเป็นต้องมีการจัดการกับข้อมูลที่ติดด้วย

ในตัวยุคเกมใช้ Class ทั้งหมด 2 Class หลักๆในการจัดการข้อมูล ดังนี้

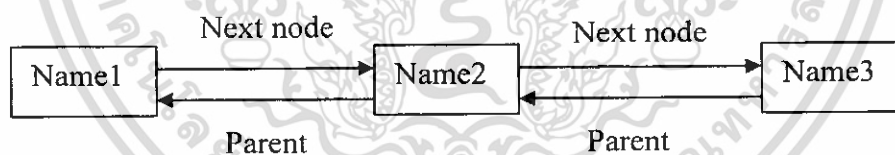
1. StoreMesh Class เป็น Class ที่ใช้ในการเก็บข้อมูลต่างๆ ของตัวละครทุกตัว โดยข้อมูลที่จัดเก็บมี ดังต่อไปนี้ พิกัดที่อยู่ของตัวละคร (x, y, z), ค่าการหันหน้าของตัวละคร (ry), ชื่อของตัวละคร, ค่าความเร็วในการโจมตี, ค่าDelayต่างๆ

2. Loadresource Class ใช้ในการเก็บข้อมูลของตัวโมเดล Frame Animation ต่างๆ โดยในช่วงแรกของโปรแกรมจะทำการโหลดข้อมูลทั้งหมดเก็บไว้ใน Class นี้ให้เรียบร้อยเพื่อให้สามารถนำไปใช้ในภายหลังได้โดยไม่ต้องโหลดซ้ำอีก

### Link List

เนื่องจากเกมนี้เป็นเกม Online จำนวนผู้เล่นจึงมีไม่แน่นอน ทางผู้จัดทำจึงเลือก Link List มาใช้ในการเก็บข้อมูล ทำให้ไม่ต้องจองเนื้อที่มากเกินไปจนเกินกว่าที่จำเป็น

ในที่นี้เราได้นำ Link List มาใช้ใน StoreMesh Class โดยเมื่อมีผู้เล่นเข้ามาเพิ่มก็จะทำการ Add Node เพื่อเพิ่มจำนวนผู้เล่น ลักษณะการทำงานของ Link list จะเป็นการเก็บข้อมูลใน Class หนึ่ง แล้วจะมีตัวแปรใน Class นั้นเป็น Pointer ทำหน้าที่ชี้ไปยังตัวต่อไป ดังรูป



ตัวอย่าง Class ที่มีการใช้ Link List

Class test

```

{
    public: LPCSTR Name;

    FLOAT x;

    FLOAT y;

    StoreMesh *nextmesh;

    StoreMesh *parent;

};
  
```

## 2.9 Network Load Balancing และ Clustering

### Cluster คืออะไร

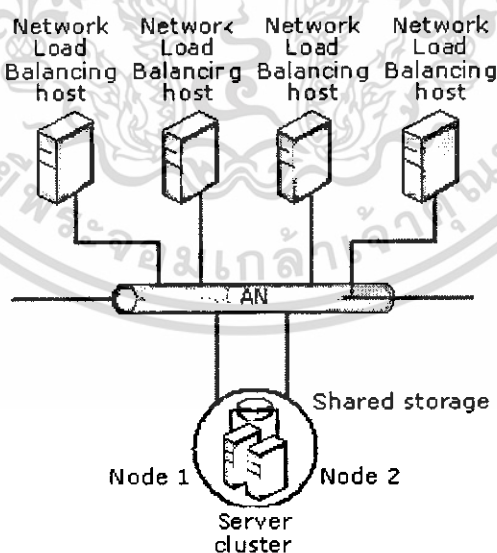
Cluster คือ การรวมกลุ่มเซิร์ฟเวอร์หลายๆ โหนดเข้าด้วยกันเสมือนว่าเป็นเซิร์ฟเวอร์ตัวเดียว ซึ่งในมุมมองของผู้ใช้อาจไม่มีโอกาสรับรู้ถึงการมีอยู่ของเซิร์ฟเวอร์ทั้งหลายภายในกลุ่มของ Cluster แต่จะเข้าใจว่าเป็นเซิร์ฟเวอร์ตัวเดียวกัน

Cluster ถูกสร้างขึ้นเพื่อรองรับระบบงานหรือแอปพลิเคชันเซิร์ฟเวอร์ต่างๆ ที่ใช้ในองค์กร โดยที่ทุกๆ เซิร์ฟเวอร์ในกลุ่มจะรู้จักแอปพลิเคชันเซิร์ฟเวอร์ที่รันอยู่บน Cluster

เทคโนโลยีของ Cluster นั้นแบ่งออกเป็น 2 แบบ ได้แก่

### 2.9.1 บริการแบบ Clustering (MSCS)

ลักษณะการทำงานของบริการแบบนี้ จะประกอบด้วยเซิร์ฟเวอร์หลักที่ทำงานอยู่ (Active) และเซิร์ฟเวอร์สำรองที่คอย Standby อยู่ โดยปกติ เซิร์ฟเวอร์หลักที่ Active จะเป็นโหนดที่ให้บริการแก่ผู้ใช้ เมื่อใดก็ตามที่เซิร์ฟเวอร์หลักไม่สามารถปฏิบัติงานได้ เช่น เกิดความผิดพลาดและดาว์นลงไป กระบวนการที่เรียกว่า FAILOVER จะเกิดขึ้นเพื่อกระตุ้นให้เซิร์ฟเวอร์ที่ Standby อยู่ขึ้นมารับช่วงทำงานแทนเพื่อให้บริการแก่ผู้ใช้ต่อไปอย่างต่อเนื่องไม่ขาดตอน ผู้ใช้เองจะรู้สึกหรือรับรู้ว่ามีเซิร์ฟเวอร์สำรองขึ้นมาทำงานแทน



รูปที่ 6 แสดงโครงสร้าง Clustering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

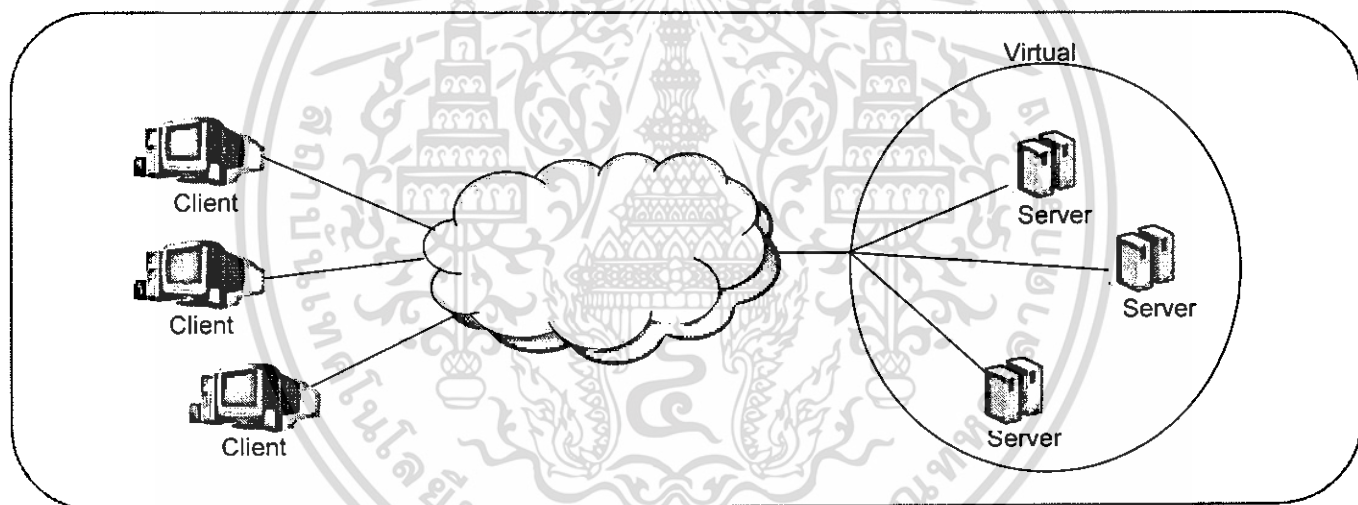
## 2.9.2 บริการแบบ Network Load Balancing (NLB)

Network Load Balancing คือ การเพิ่มขีดความสามารถในเรื่องการใช้งานและการขยายระบบ โดยเน้นในเรื่องการให้บริการพื้นฐานบนระบบเครือข่าย เช่น TCP/IP และการให้บริการต่างๆ เกี่ยวกับเว็บไซต์ ซึ่งสามารถแบ่งได้เป็น 2 แบบหลัก ดังนี้

แบบ Scalability หมายถึง การที่ระบบสามารถถูกเพิ่มขยายเพื่อรองรับปริมาณที่เพิ่มขึ้นของโหลดหรือ ทราฟฟิกที่วิ่งเข้ามาได้อย่างต่อเนื่อง หากโหลดมาก เราสามารถเพิ่มจำนวนของเซิร์ฟเวอร์เข้าไปเพื่อช่วยกันทำงานและแบ่งเบาภาระซึ่งกันและกันได้

แบบ Availability หมายถึง การที่ระบบสามารถให้บริการแก่ผู้ใช้ได้อย่างต่อเนื่อง ไม่หยุดชะงักในกรณีที่มียางเซิร์ฟเวอร์ทำงานผิดพลาดไป เซิร์ฟเวอร์อื่นก็สามารถรับหน้าที่ทำงานแทนต่อได้

บริการแบบ NLB นี้เหมาะกับการประยุกต์ใช้ "เซิร์ฟเวอร์ฟาร์ม" หรือกลุ่มของเซิร์ฟเวอร์ที่รองรับระบบงานเฉพาะด้าน อย่างเช่น การทำเซิร์ฟเวอร์ฟาร์มให้กับเกมเซิร์ฟเวอร์

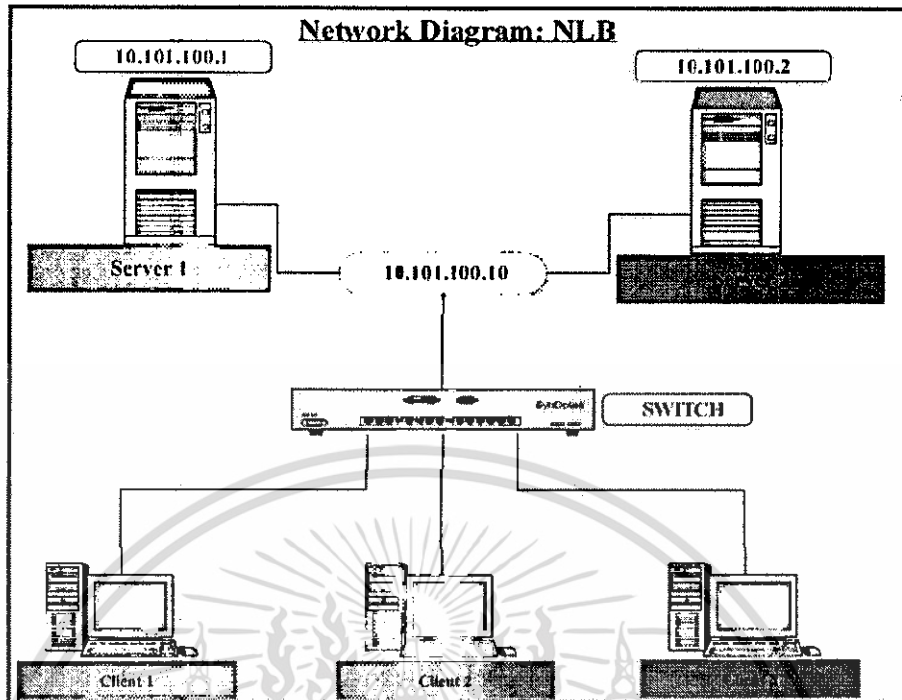


รูปที่ 7 แสดงหลักการทำงานของ Network Load Balancing

จากรูป

1. เมื่อเซิร์ฟเวอร์ได้รับคำร้องขอจากเครื่อง Client
2. มีการส่งกระจายคำร้องขอไปยังเซิร์ฟเวอร์ที่อยู่ภายใน Cluster เดียวกัน
3. จะมี Server หนึ่งที่สามารถจัดการกับคำร้องขอของเครื่อง Client ซึ่งผู้ใช้สามารถใช้งานได้อย่างต่อเนื่องตามปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8 แสดงการทำงานของ Network Diagram NLB

จากรูปแสดงการทำงานโดยรวมจะเป็นไปในลักษณะว่ามีการสร้าง Virtual Server (เซิร์ฟเวอร์เสมือน) ในที่นี้คือ 10.101.100.10 ซึ่ง Client จะทำการติดต่อเข้ามาที่ IP ของ Virtual Server นี้และ Virtual Server จะทำหน้าที่ Balance Load ที่เข้ามาให้กับ Server ทั้งสองตัว

ซึ่งข้อดีของโครงสร้างลักษณะนี้ คือ ในกรณีที่เกิดความผิดพลาดขึ้นกับ Server เครื่องใดเครื่องหนึ่ง และ Server อื่นยังสามารถทำงานได้ ระบบจะยังคงทำงานต่อไปได้ โดย Client Side อาจไม่ทราบเลยว่ามี Server ใดเกิดความผิดพลาดไป

## บทที่ 3

### Graphic Tool (Milk shapes 3D 1.7.8)

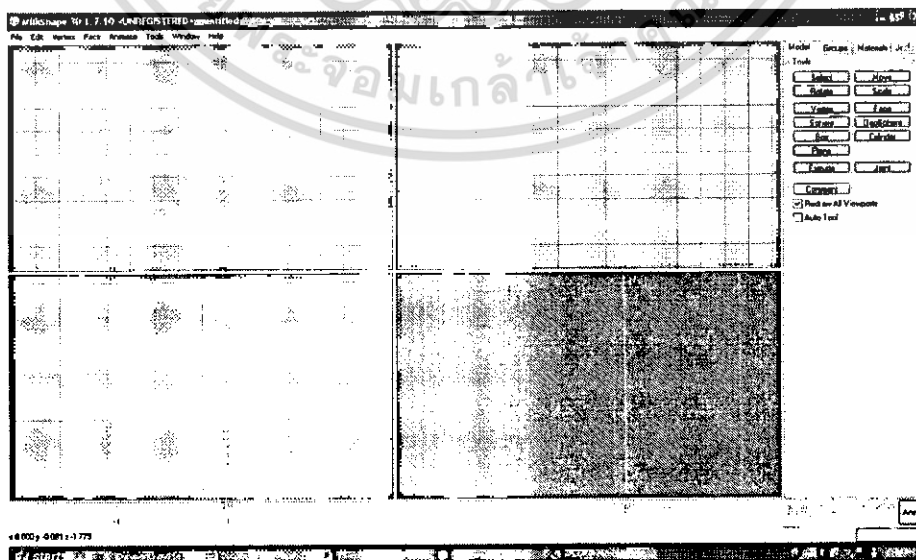
#### 3.1 เหตุผลที่เลือกใช้

1. เป็นโปรแกรมที่สามารถ import และ export ได้หลายนามสกุลที่รองรับกับการสร้างเกมส์โดยไม่ต้องใช้ plug in
2. เป็นโปรแกรมที่สามารถสร้าง model mesh ที่เป็น low polygon ได้เหมาะสมกับการนำมาใช้ในเกมส์
3. เป็นโปรแกรมที่มีความสามารถในการแกะ texture ให้กับ character ที่สร้างได้ง่ายไม่ซับซ้อนและสามารถแก้ไขได้ง่าย
4. เป็นโปรแกรมที่ค่อนข้างสร้างขึ้นมาเพื่อรองรับสำหรับการสร้างเกมส์ จึงทำให้สามารถสร้างการเคลื่อนไหวได้ไม่ยากนัก

#### 3.2 การสร้าง model สามมิติ

ในส่วนของการสร้างภาพสามมิตินั้นจากที่ได้กล่าวไว้แล้วว่าเราจะเลือกใช้โปรแกรม Milkshape เป็น tool ในการสร้างโดยจะแบ่งขั้นตอนออกเป็น 3 ส่วนหลักๆคือ

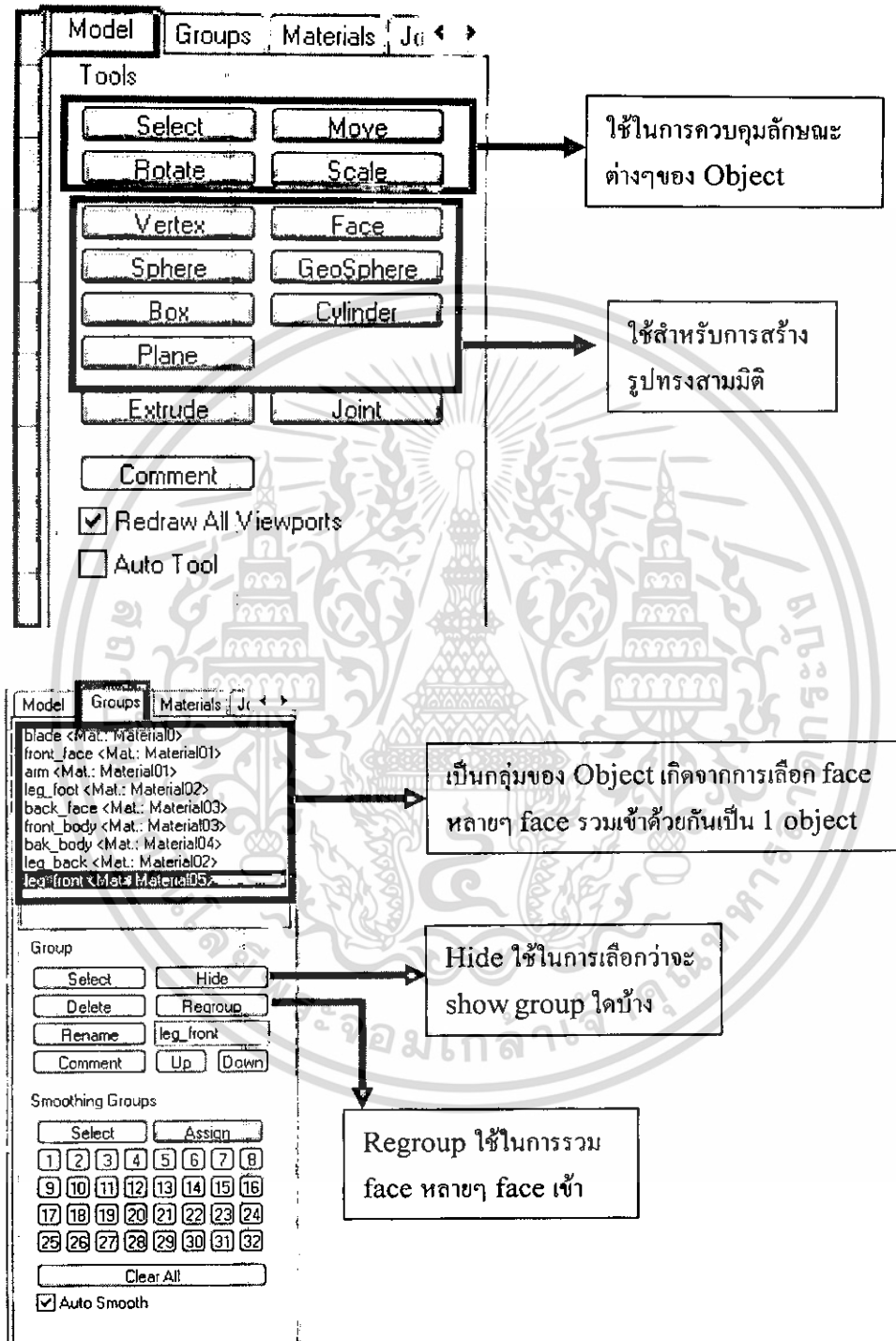
1. การขึ้นรูป model
2. การสร้างและการแกะ texture
3. การทำ animation



รูปที่ 9 แสดง Interface ของโปรแกรม Milk shapes

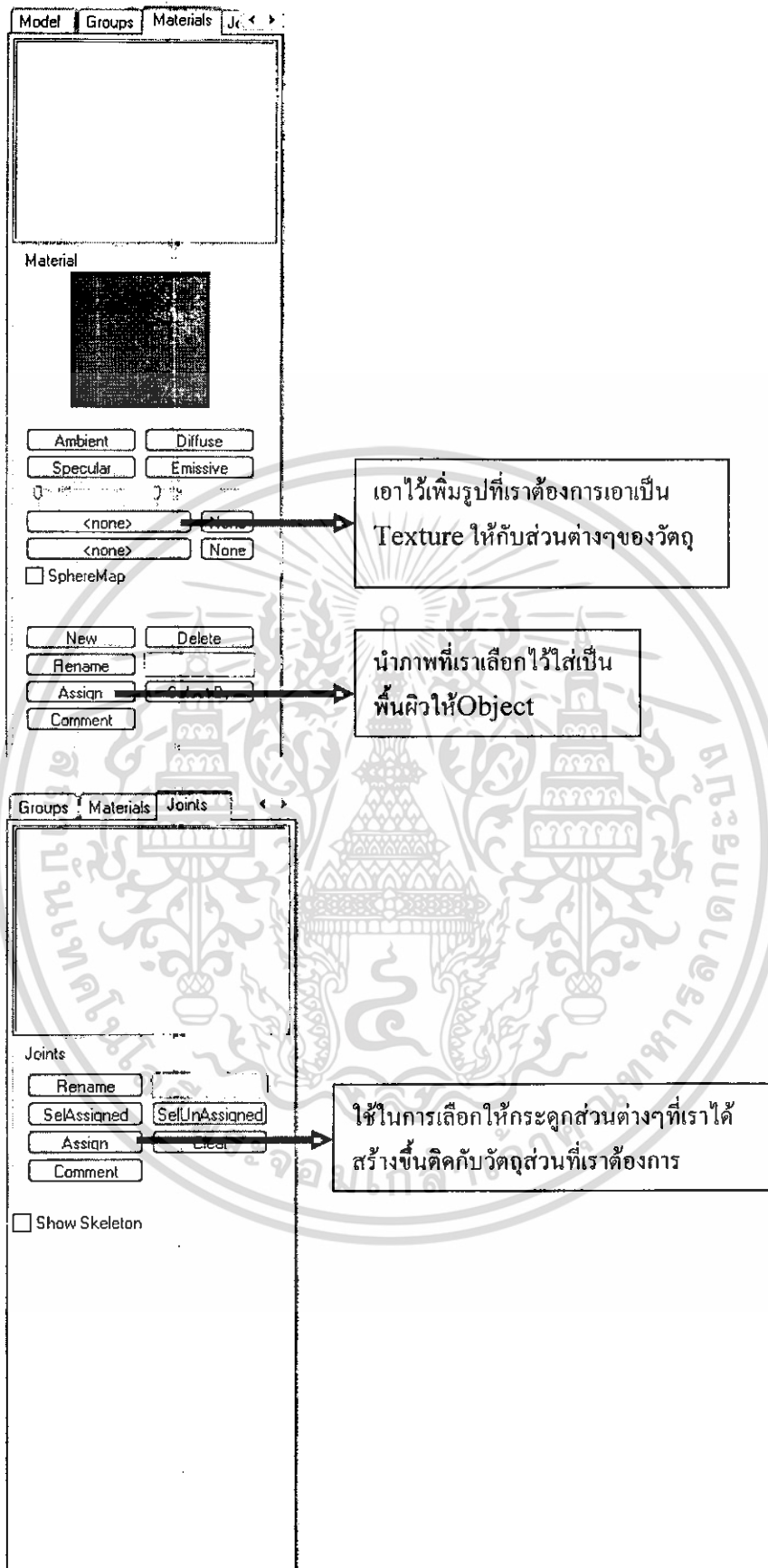
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปเป็นหน้าต่างโปรแกรมซึ่งจะมีแถบเครื่องมือที่ใช้หลักๆ ในการสร้าง Mode I แล้วอยู่ทางบนขวามือและทางขวาล่างจะจะเป็นอุปกรณ์ในการบังคับการเล่น animation



รูปที่ 10 แสดงการใช้งานเบื้องต้น Milk shapes

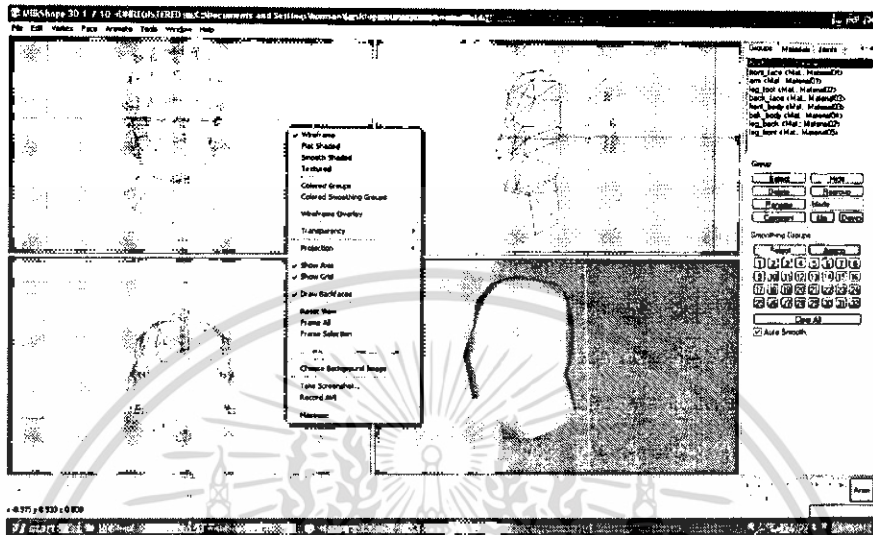
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 11 แสดงการใช้งานเบื้องต้น Milk shapes

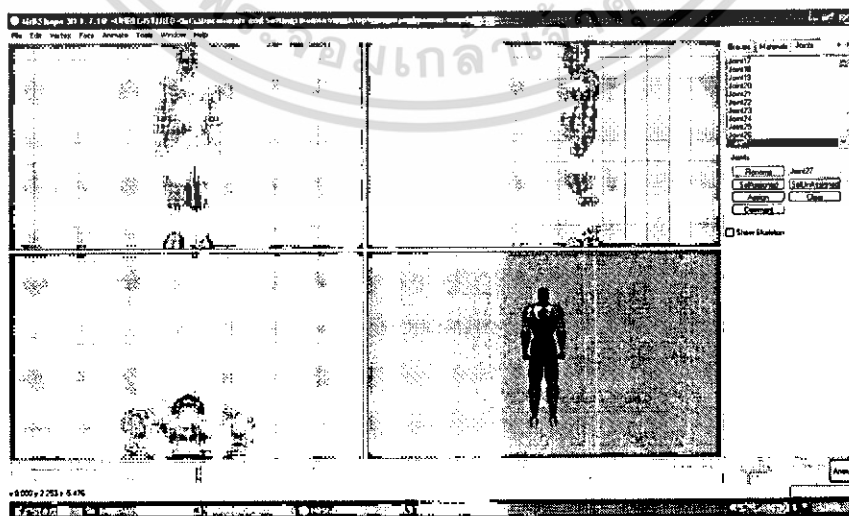
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการสร้าง Model เริ่มต้นด้วยการใช้เครื่องมือสร้างรูปทรงทางเรขาคณิตก่อนโดยเลือกรูปทรงที่มีความใกล้เคียงกับส่วนที่เราต้องการวาด เช่น ถ้าเราต้องการเริ่มวาดหัวก่อนเราก็จะเริ่มจากการสร้างรูปทรงกลมก่อนดังรูป



รูปที่ 12 แสดง Interface การขึ้นรูปเบื้องต้น

ใน Tool แบบนี้เราสามารถเลือกได้ว่าจะมองมุมมองใดและในลักษณะใด เช่น wireframe ก็จะทำให้เราสามารถที่จะเห็นโครงสร้างของ vertex และ face ทำให้สามารถ edit ได้ง่ายขึ้น จากนั้นเมื่อสร้างโครงสร้างพื้นฐานขึ้นมาแล้วเราก็จะใช้เครื่องมือ select เพื่อเลือกส่วนของวัตถุแต่จะต้องเลือกว่าจะเลือกแบบใด vertex, face หรือว่า group จากนั้นเมื่อเราเลือกส่วนของวัตถุได้แล้ว จากนั้นเราก็จะใช้เครื่องมือ move, scale หรือ rotate เพื่อใช้ในการเคลื่อนย้าย ปรับขนาด หรือว่าหมุนส่วนของวัตถุที่เราได้เลือก ให้เป็นตามที่เราต้องการ

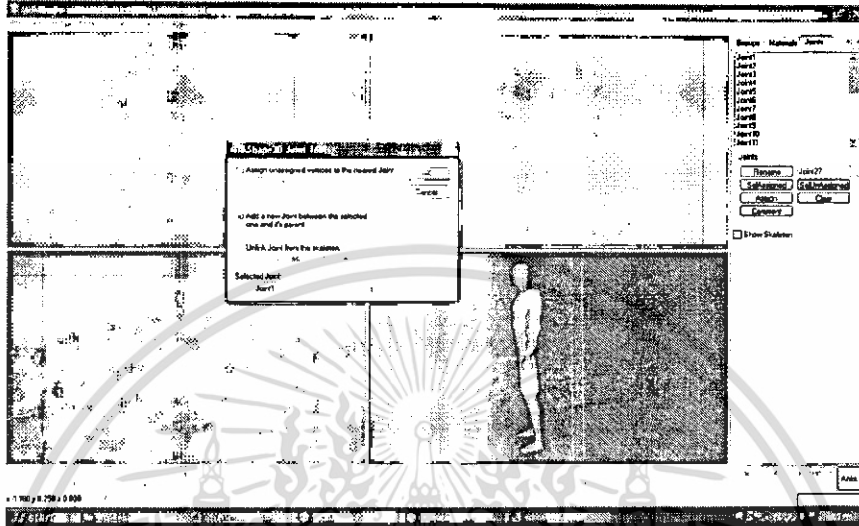


รูปที่ 13 แสดง Interface การขึ้นรูปตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

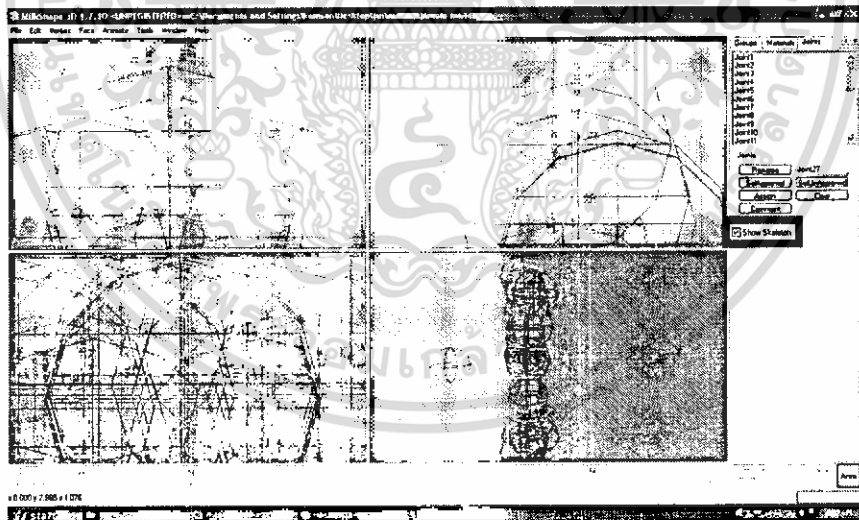
## การสร้างภาพเคลื่อนไหว

สุดท้ายเมื่อเราทำการสร้างโครงสร้างทั้งหมดจนเสร็จเรียบร้อยแล้วต่อไปเราก็ทำการสร้างโครงกระดูกให้กับ Model ด้วยเครื่องมือที่แถบเครื่องมือ tool จากนั้นเลือก joint tool จะปรากฏหน้าต่างดังรูป



รูปที่ 14 แสดงการสร้างภาพเคลื่อนไหว 1

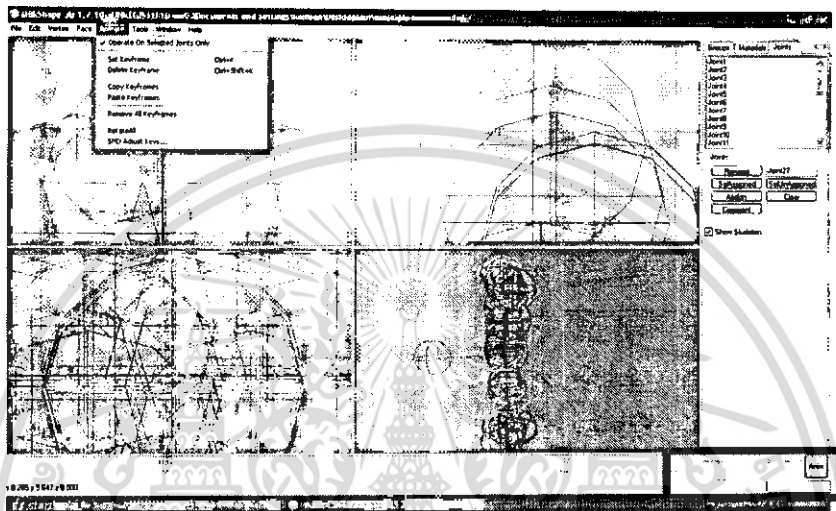
จากนั้นเราจะสามารถสร้างกระดูกได้ที่ละข้อต่อไปเรื่อยๆ ดังรูป



รูปที่ 15 แสดงการสร้างภาพเคลื่อนไหว 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้จากรูปว่าเมื่อเราแสดงส่วนของกระดูกอยู่บนหน้าต่างการทำงานจะทำให้เราสามารถแก้ไข Model ได้ลำบากเราจึงสามารถที่จะกดเพื่อให้เห็นหรือไม่แสดงกระดูกได้ เมื่อเราทำการสร้างกระดูกเสร็จเรียบร้อยแล้วก็จะมาทำการเชื่อมต่อบนส่วนของ model ที่เราสร้างขึ้นเข้ากับโครงกระดูกที่เราสร้างขึ้นด้วยแถบเครื่องมือทางขวาแถบ joint แล้วคลิกเลือก joint number จากนั้นเลือก assign ให้กับโครงสร้างส่วนที่เราต้องการ

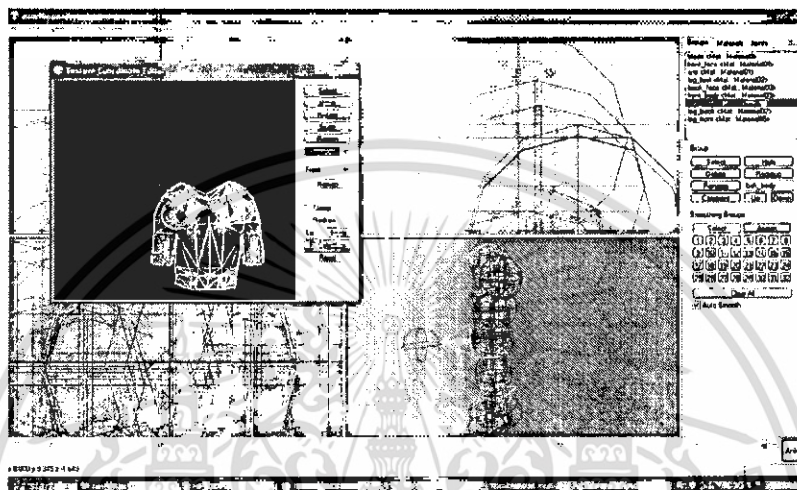


รูปที่ 16 แสดงการสร้างภาพเคลื่อนไหว 3

เมื่อเราทำการยึดโครงกระดูกเข้ากับเข้ากับ Model แล้ว ต่อไปเราจะทำการสร้างการเคลื่อนไหวให้กับตัวละครโดยที่เราจะทำการสร้างท่าทางให้ model ด้วยการปรับท่าทางแต่ละท่าให้กับแต่ละ frame ให้มีความต่อเนื่องกันโดยเราจะต้องเลือก set keyframe ทุกครั้งที่เราทำการกำหนด frame ให้กับแต่ละท่าทางของ model โดยอย่าลืมว่าก่อนที่เราจะทำการสร้างภาพเคลื่อนไหวนั้นเราจะต้องทำการเปิดปุ่ม Anime ก่อน

## การปะ Texture

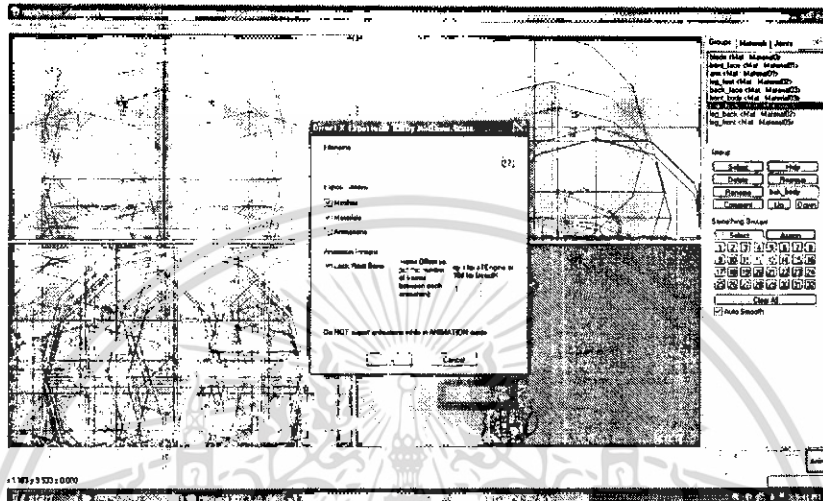
เป็นส่วนของการตกแต่งความสมจริงให้กับ Model ด้วยการปะ texture ให้กับ model ด้วยแถบ Window>texturecoordinator จากนั้นจะปรากฏหน้าต่างดังรูป เพื่อจะกำหนดพิกัดให้ การสร้างรูปภาพได้ถูกตำแหน่งและเราสามารถที่จะเลือกว่าจะเอามุมมองใดจากแถบเลือก ด้านขวาโดยโปรแกรมจะทำการคลี่โครงmodel ออกมาเป็นแผ่นเพื่อให้วาดพื้นผิวลงไป



รูปที่ 17 แสดงการปะ Texture 1

จากนั้นเราก็สร้างไฟล์รูปภาพจากโปรแกรม Photoshop ลงบนตำแหน่งดูด้วยการ print screen จากหน้าต่างนี้ไปแล้วเมื่อได้รูปภาพแล้วเราก็ทำการใส่ภาพให้กับวัตถุส่วนที่เราต้องการโดยมีข้อควรระวังตรงที่หากว่าเราทำการแยกส่วนประกอบของmodel ออกเป็นหลายๆส่วนยิ่งมากก็จะทำให้ไฟล์ที่ได้มีขนาดใหญ่ซึ่งมีผลต่อการกระตุกในการเล่นเกมส์ แต่จะได้model ที่มีรายละเอียดมาก และมีความสมจริงมากกว่าการใช้texture เพียงน้อยชิ้น

ขั้นตอนสุดท้ายเมื่อเราจะทำการนำ Model ที่เราสร้างนั้นมาใช้งานก็จะต้องทำการ export จากไฟล์ที่เราทำงาน (.ms3d) ไปเป็นไฟล์ .X ด้วยการเลือกที่ file>export>directx 8.0 จากนั้นเราก็จะต้องป้อนค่าให้กับหน้าต่างที่ปรากฏขึ้นมาดังรูป try1 for JT Engine or 160 for directx โดยเปลี่ยนค่า 1 ให้เป็นค่า 160 เพื่อให้เป็นfileที่สามารถนำไปใช้กับ directx



รูปที่ 18 แสดงการ Export File เป็น .x เพื่อใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทวิจารณ์สำหรับการสร้าง Model สามมิติ และ animation

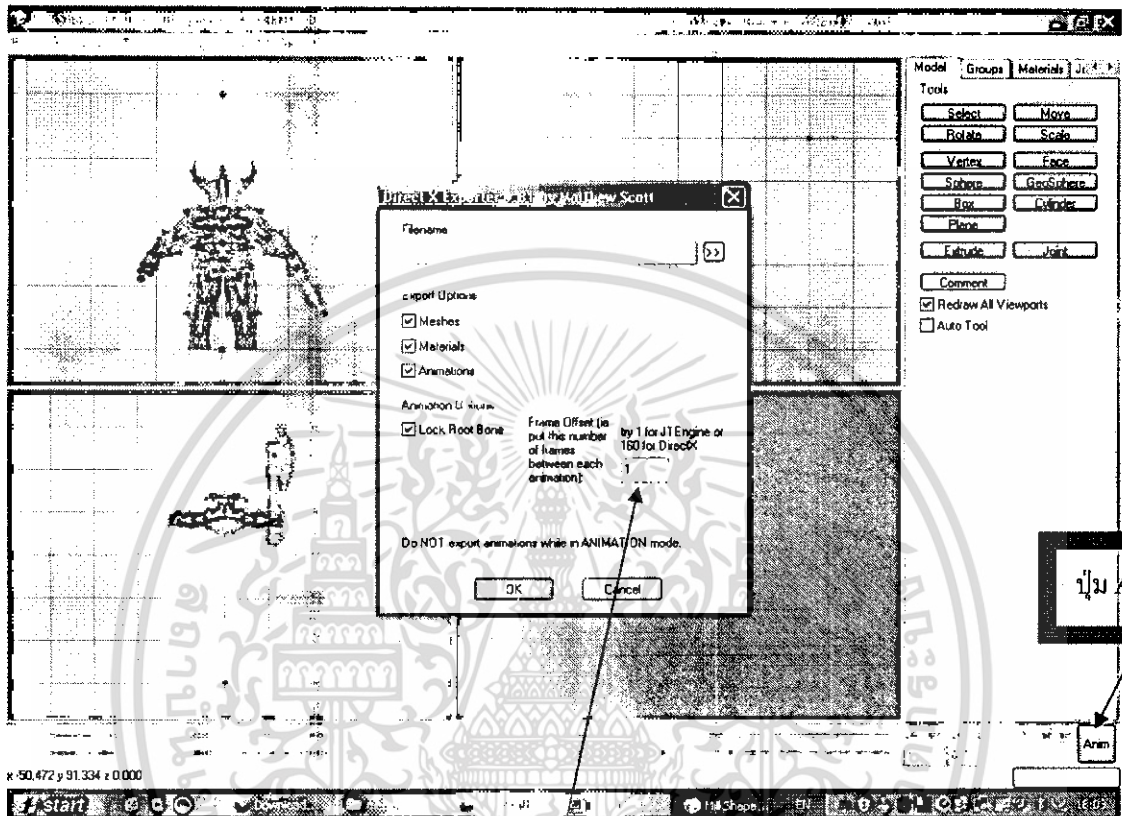
1. โดยทั่วไป การขึ้นรูปโมเดลจากรูปทรงเรขาคณิตนั้นจะใช้สี่เหลี่ยมเป็นหลัก ซึ่งอาจจะยากไป หากจะทำให้เป็นโมเดลดังที่ต้องการ เนื่องจากจำเป็นต้องมีการใช้เทคนิคหลายอย่างเพื่อต่อเติม รูปทรงสี่เหลี่ยมธรรมดาๆ ให้กลายเป็นโมเดลความละเอียดสูง แต่การสร้างโมเดลในอีกรูปแบบหนึ่ง ที่ง่าย และรวดเร็ว นั่นก็คือการใช้รูปทรงกระบอกนั่นเอง รูปทรงกระบอกมีคุณสมบัติที่เหมาะสมในการขึ้นรูปคือ

- มีจำนวน Vertex ที่มาก ง่ายต่อการดัดแปลง
- มีพื้นฐานมาจากรูปวงกลม ทำให้มีความนุ่มนวลอยู่ในตัว
- สามารถกำหนดปริมาตร polygons ได้ง่ายตั้งแต่เริ่มแรก

แต่ถึงแม้จะมีข้อดี แต่ข้อเสียของการใช้รูปทรงกระบอกก็มี คือ จำเป็นต้องอาศัยความชำนาญหากจะสร้างโมเดลที่มีความซับซ้อน และอาจจะต้องอาศัยรูปทรงกระบอกหลายอันเพื่อสร้างโมเดล 1 ชิ้น



2. การ export file จากโปรแกรม Milk shapes เพื่อนำมาใช้ในเกมส์จะต้องเลือกให้ถูกเพื่อให้สามารถนำมาใช้กับ direct ได้ไม่ใช่ใช้กับ dark basic เพราะเนื่องจากว่าปกติมันจะ set default ไว้เป็น dark basic



รูปที่ 19 แสดงการ Export File เป็น .x เพื่อใช้งาน 2

โดยที่เราจะต้องเปลี่ยนค่าพารามิเตอร์ในช่อง ให้เป็น 160 เพื่อ export เป็น directX และเวลาที่ทำการ export นั้นจะต้องปิดปุ่ม animation ก่อนเพื่อให้ตัวละครอยู่ในท่าปกติและไม่ให้การ export ออกมาผิดปกติ

3. การนำ mesh บางตัวที่มีการทำที่มีรายละเอียดมาก เช่น ต้นไม้ เราจะไม่สร้าง model ที่มีลักษณะเป็นรอยเว้าแหว่ง หรือมีรูปทรงเป็นไปไม่ได้แต่เราจะสร้างเป็นลักษณะและจะใช้ texture มาแปะแทนแต่ว่ารูปไปไม้จริงๆจะแปะอยู่บนพื้นสีดำดังนั้นเราจึงต้องไม่ให้ส่วนที่เป็นพื้นแบกกราวไม้แสดงผลเพื่อให้เห็นเฉพาะส่วนที่เป็นก้านและใบไม้ ด้วยการตั้งค่าสั่ง transparency แล้วเซตส่วนที่ไม่ต้องการให้มีค่าต่ำๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การทำให้ฉากมีแผนที่ในการใช้แบบ DirectX นั้นจะต้องสร้าง sky box ขึ้นมาเองเพื่อทำการเขียนโปรแกรมให้แสดงรูปท้องฟ้าจะแปอยู่ใน plane ของกล่องจำลองที่เสมือนกับครอบฉากอยู่หึ่งฉากซึ่งค่อนข้างจะยุ่งยากต่างจากใน dark basic ที่มี plug in สำหรับการสร้างท้องฟ้าขึ้นมาได้อัตโนมัติ
5. การใช้ DirectX ที่ version ต่างกันมีผลให้การแสดงผลมีความผิดเพี้ยนจากที่ควรจะเป็นทั้งในเรื่อง position ขนาด และที่สำคัญจะมีความผิดเพี้ยนของในเรื่อง animation ที่จะมีความเร็วที่ผิดปกติดด้วย

ในการทำงานกับโปรแกรม MilkShape3D นั้น การเลือกระดับการทำงานกับโมเดลเป็นสิ่งสำคัญมาก ระดับการทำงานในที่นี้หมายถึงว่า เราจะทำงานกับ vertex, surface, joint หรือทั้ง Object ซึ่ง MilkShape3D ก็มีคำสั่งช่วยในการทำงานแบบนี้อยู่ นั่นก็คือ select options เป็นปุ่มที่กำหนดว่าจะเลือกทำงานในระดับไหน นอกจากนี้ปุ่มบน mouse ก็ช่วยได้มากทีเดียว

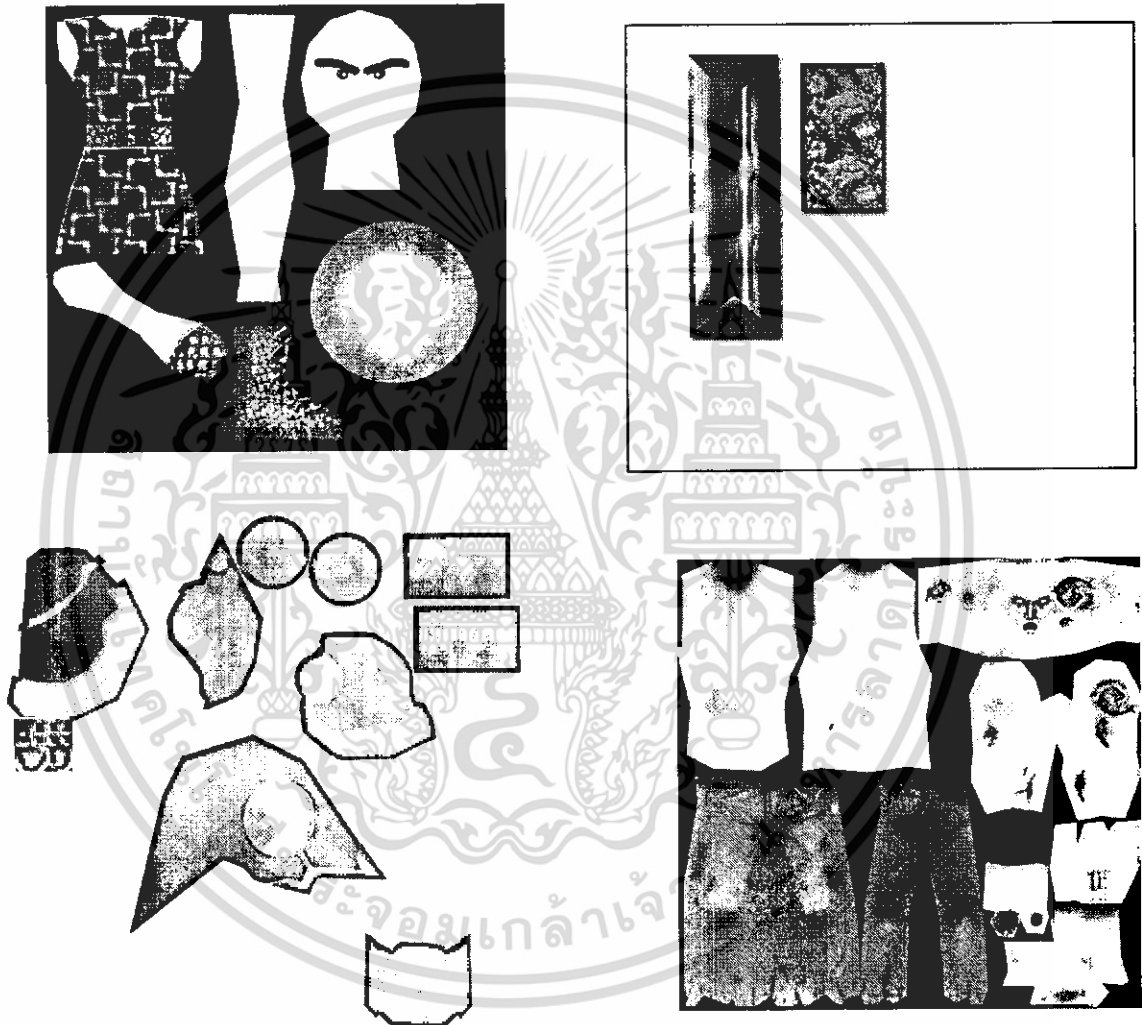
1. การเลือกการทำงาน คลิกที่ปุ่ม select บน Tool bar และเลือกระดับการทำงานที่ปุ่ม Select Options ซึ่งจะมีให้เลือก 4 ระดับคือ

- Vertex คือให้เลือกเฉพาะจุด Vertex
- Face คือให้เลือกเฉพาะ Surface
- Group คือให้เลือก Object หรือโมเดลที่อยู่ใน Group เดียวกัน
- Joint คือให้เลือกเฉพาะ Joint

นอกจากนี้ยังมีปุ่ม Ignore back surface เพื่อไม่ให้มีการเลือกในจุดใดๆ หรือ surface ใดๆ ที่ซ้อนอยู่ด้านหลัง

## การประยุกต์และการนำไปพัฒนาต่อ

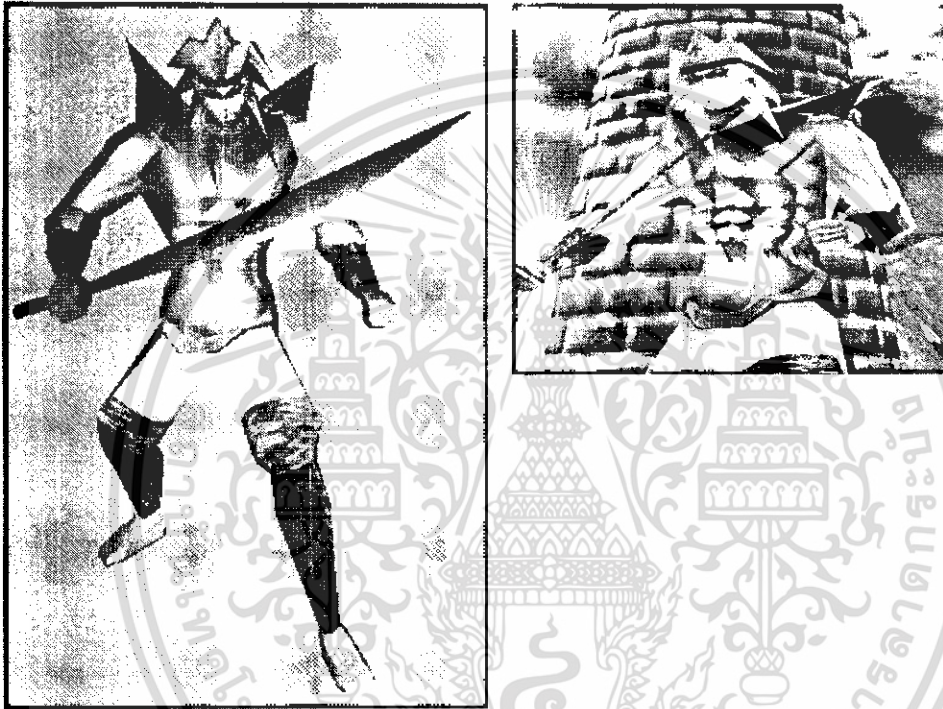
1. การสร้างตัวละครเราสามารถที่จะสร้าง (ขึ้นรูป) ตัวละครสามมิติเพียงตัวเดียวแต่เราสามารถที่จะทำให้มีตัวละครที่แตกต่างกันได้จำนวนมากมาด้วยการสร้าง texture ที่แตกต่างกันปะเข้าไปใน model ตัวเดียวกันสามารถนำไปประยุกต์ในการสร้างชุดให้กับตัวละครได้หลายๆชุด (เปลี่ยนสี เปลี่ยนแบบ) โดยที่เป็นตัวละครตัวเดิม



รูปที่ 20 แสดงภาพการประยุกต์ภาพ Texture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การนำ model ไป render ในฉาก DirectX นั้นหากจะต้องการให้ภาพออกมาสวยงามสมจริง เหมือนกับที่แสดงผลอยู่ในโปรแกรมสร้างภาพสามมิติ หรือโปรแกรมดูภาพสามมิติ mesh viewer ที่มีติดมากับ Microsoft DirectX SDK นั้นเราจะต้องให้แสง (light) และเงา(shading) อย่างถูกต้องเนื่องใน DirectX นั้นเราจะต้องกำหนดจำนวนแหล่งแสง และประเภทของแหล่งแสงเอง รวมทั้งตำแหน่งที่วางระยะการส่องซึ่งต่างจากใน game engine หรือโปรแกรมดูภาพสามมิติที่จะมีการสร้างแสงให้อัตโนมัติซึ่งจะทำให้ภาพที่ได้ออกมามีความสมจริง



รูปที่ 21 แสดงการเปรียบเทียบภาพก่อนและหลังฉายแสง

จะเห็นได้ว่าภาพทั้ง 2 ภาพมีความแตกต่างกันคือภาพในทางซ้ายจะเป็นภาพที่ render จากโปรแกรม mesh viewer ซึ่งจะมีแสงและเงาทำให้ภาพดูมีมวลมีน้ำหนักดูมีความสมจริงกว่าภาพทางขวามือที่เป็นภาพที่ render จาก DirectX ซึ่งจะขาดน้ำหนักดูแบนกว่าภาพทางซ้าย

## บทที่ 4

### การออกแบบและพัฒนา Software

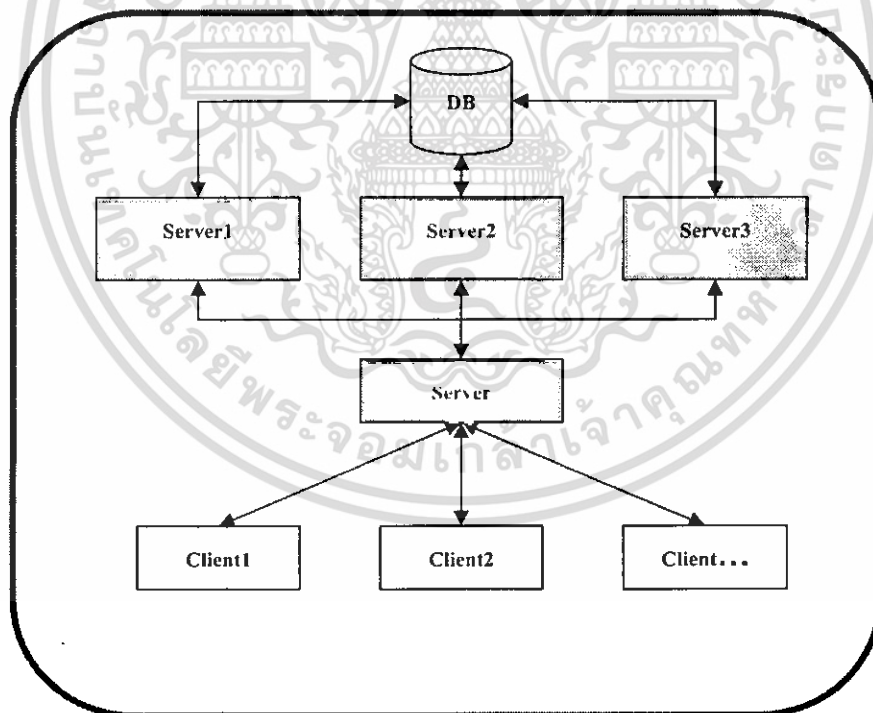
เนื้อหาภายในบทนี้จะทำการพูดถึง ระบบการทำงานของ Software ทั้งหมดที่มีภายในโครงงาน โดยภายในโครงงานนี้ได้ใช้ภาษา C++ ในการพัฒนาทั้งหมด และใช้ API DirectX มาช่วยในการเขียนโปรแกรม ในส่วนของ Programming สามารถแบ่งเนื้อหาออกเป็นส่วนต่างๆ ได้หลายส่วน ดังนี้

#### 4.1 เครื่องมือที่ใช้ในการพัฒนาโปรแกรม

- Visual Studio 2003
- DirectX SDK 9.0 เป็น API

#### 4.2 ส่วนการทำงานของ Software

การทำงานของ Software นั้นแบ่งออกเป็นหลายส่วนด้วยกัน ดังรูปต่อไปนี้



รูปที่ 22 แสดงโครงสร้างของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.1 ส่วนโปรแกรม Client

เป็นโปรแกรมที่ใช้สำหรับติดตั้งบนเครื่อง Client เพื่อเข้าทำการเล่นเกม รายละเอียดของตัวเกมรวมไปถึงการทำงานและการแสดงผลทั้งหมดจะถูกควบคุมด้วยโปรแกรมชุดนี้ ผู้เล่นจำเป็นจะต้องควบคุมตัวละครผ่านโปรแกรมชุดนี้ เมื่อโปรแกรมเริ่มทำงานโปรแกรมชุดนี้จะทำการติดต่อไปยัง Program Server

#### 4.2.2 ส่วนโปรแกรม Server Load Balance

เป็นโปรแกรมที่ใช้สำหรับแยก Connection ต่างๆให้ติดต่อกับ Server ภายในอย่างถูกต้อง และมีจำนวน Connection ใกล้เคียงกัน โดยสังเกตจาก Map ของผู้เล่นและข้อมูลใน Database ว่าขณะนั้นมีจำนวน Connection ใน Server เท่าใด

#### 4.2.3 ส่วนโปรแกรม Server Game1,2,3

เป็นโปรแกรมที่ทำงานคล้ายกันต่างกันที่แต่ละ Server จะจัดการกับผู้เล่นคนละ Map และทำการส่งข้อมูลเฉพาะ Map ให้กับผู้เล่นเท่านั้น

#### 4.2.4 ส่วนโปรแกรม Database

เป็นโปรแกรมที่ใช้ในการควบคุมการทำงานต่างบนฝั่ง Database โดยโปรแกรมชุดนี้หลักๆแล้วจะทำการควบคุมตัวศัตรูทั้งหมดภายในเกมให้เคลื่อนไหวเข้ามาต่อสู้กับผู้เล่นรวมไปถึงควบคุมการเคลื่อนไหวต่างๆของตัวศัตรูอีกด้วย

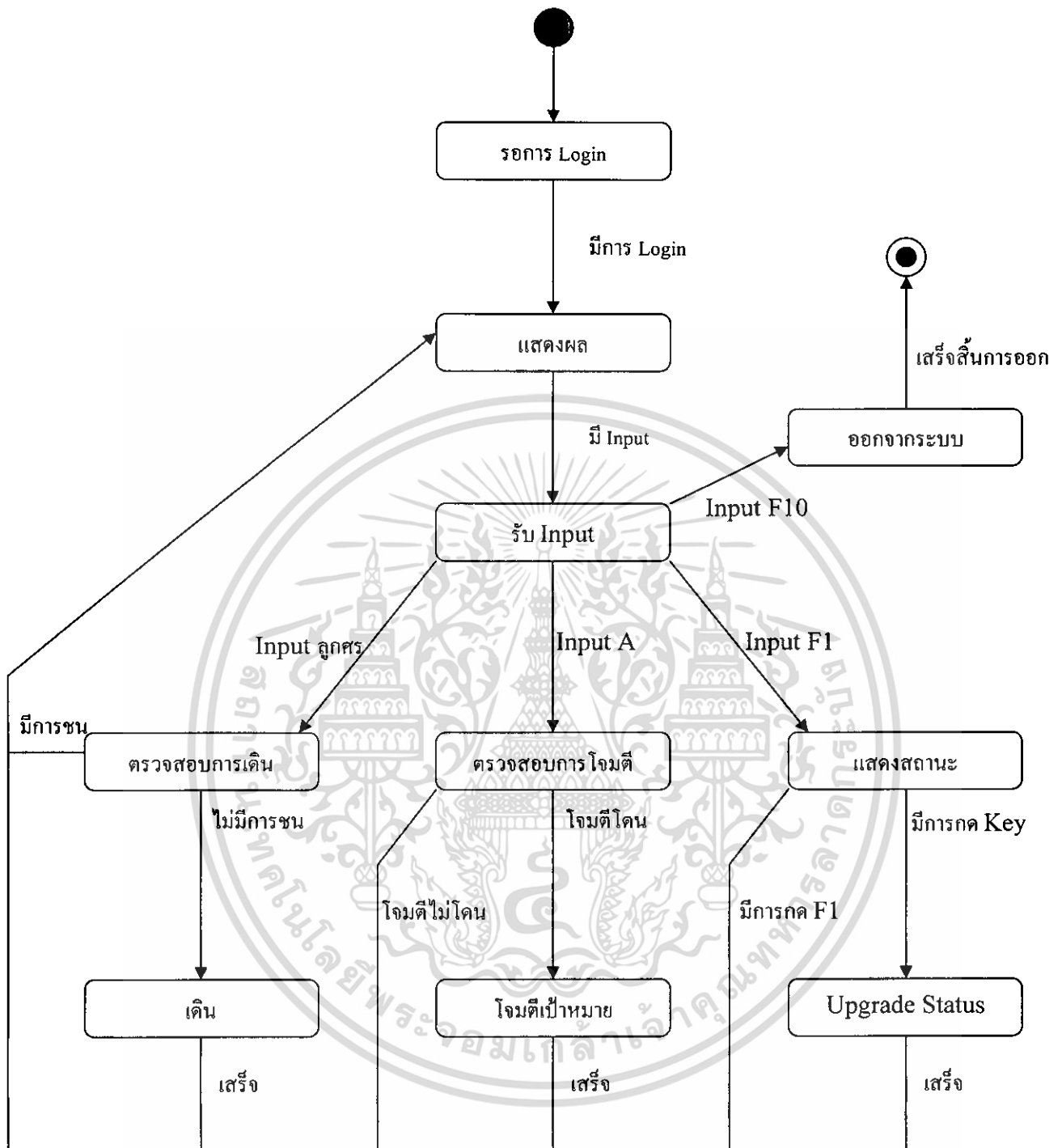
### 4.3 การทำงานของ Game (Client Programming)

#### เป้าหมายในการพัฒนา

1. พัฒนาโปรแกรม Client ที่สามารถแสดงภาพสามมิติรวมไปถึงแสดงการเคลื่อนไหว (Frame Animation) ได้
2. โปรแกรมสามารถรับ Input ผ่าน Key Board และนำมาใช้ในการควบคุมการทำงานของโปรแกรมได้
3. พัฒนาโปรแกรมให้สามารถรองรับผู้เล่นจำนวนมากได้โดยทำการติดต่อกันผ่าน Server
4. พัฒนาโปรแกรมให้สามารถเล่นเป็นเกมแนวผจญภัยสามมิติและผู้เล่นสามารถเล่นได้ อย่างสนุกสนาน

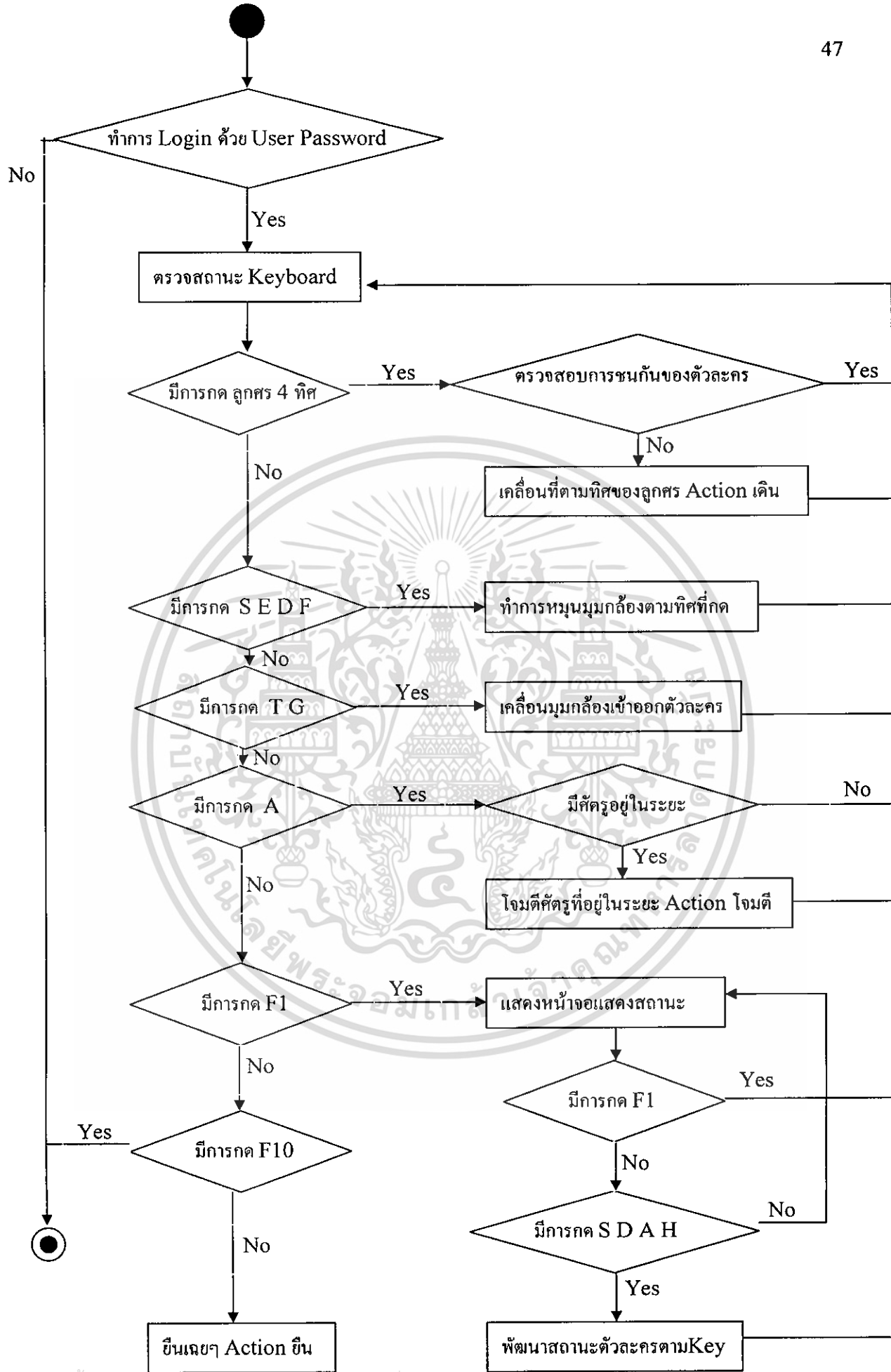
### พัฒนาการของโปรแกรม

1. เขียนโปรแกรมเพื่อสร้าง Window ที่รองรับการเรนเดอร์ขึ้น สามมิติ
2. เขียนฟังก์ชันรับข้อมูลจาก X file เพื่อนำข้อมูล Mesh Object มาใช้
3. เขียน Class เพื่อใช้ในการควบคุมการทำงานของ Mesh เช่น
  - แสดงผลโมเดลสามมิติ (Render Mesh)
  - ควบคุม Position ของ Mesh (Translation Mesh)
  - การหมุน Mesh (Rotation)
  - การกำหนดขนาดย่อขยาย Mesh (Scaling)
4. เขียนโปรแกรมโดยใช้ Direct Input เพื่อรับข้อมูลจาก Key Board เพื่อควบคุมตัวละคร
5. เขียนโปรแกรมควบคุมมุมมองกล้องภายในโลกสามมิติโดยสามารถควบคุมได้ดังต่อไปนี้
  - หมุนมุมมองกล้องไปทางซ้ายและขวาของตัวละคร
  - หมุนมุมมองกล้องไปทางด้านบนและล่างของตัวละคร
  - Zoom เข้าและออกจากตัวละคร
6. เขียนโปรแกรมเพื่อรองรับการทำงานของ Frame Animation รวมไปถึงกำหนดการเคลื่อนไหวของ Object ให้สอดคล้องกับ Input ที่ได้รับมา
7. นำการทำงานทั้งหมดในข้างต้นมาประยุกต์เขียนโปรแกรมเกมรวมไปถึงการควบคุมการทำงานส่วนต่างๆ ของเกม ดังนี้
  - แสดงตัวละครพร้อมการเคลื่อนไหวให้สอดคล้องกับ Input ที่ได้รับ
  - มีการตรวจสอบการชนกันของตัวละคร โดยทำการสร้างกล่องครอบตัวละครไว้เพื่อใช้ในการตรวจสอบ
  - ตรวจสอบการโจมตีโดยหากมีการต่อสู้กันเกิดขึ้นจะต้องมีการลดเลือดกัน
  - มีการกำหนดค่าสถานะ (Status) ต่างๆ ภายในเกมให้กับตัวละครต่างๆ
8. เขียนโปรแกรมเพื่อแสดงสถานะต่างๆ ผ่าน Interface โดยใช้การแสดงผลสองมิติ
9. ทำการเพิ่มเสียงเพลง รวมไปถึงเสียงต่างๆของตัวละครภายในเกม โดย Direct Sound
10. พัฒนาโปรแกรมติดต่อบริ-ส่งข้อมูลระหว่าง Client-Server เพื่อให้การเคลื่อนไหวและการทำงานของตัวละครถูกรายงานผลไปยัง Server ในขณะเดียวกัน Client ก็สามารถรับข้อมูลของผู้เล่นคนอื่นๆ รวมไปถึงศัตรูได้
11. ทำการตรวจสอบ และแก้ไขข้อผิดพลาดที่พบ พร้อมกับพัฒนาในส่วนของคุณภาพ



รูปที่ 23 State Diagram ของ Software Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

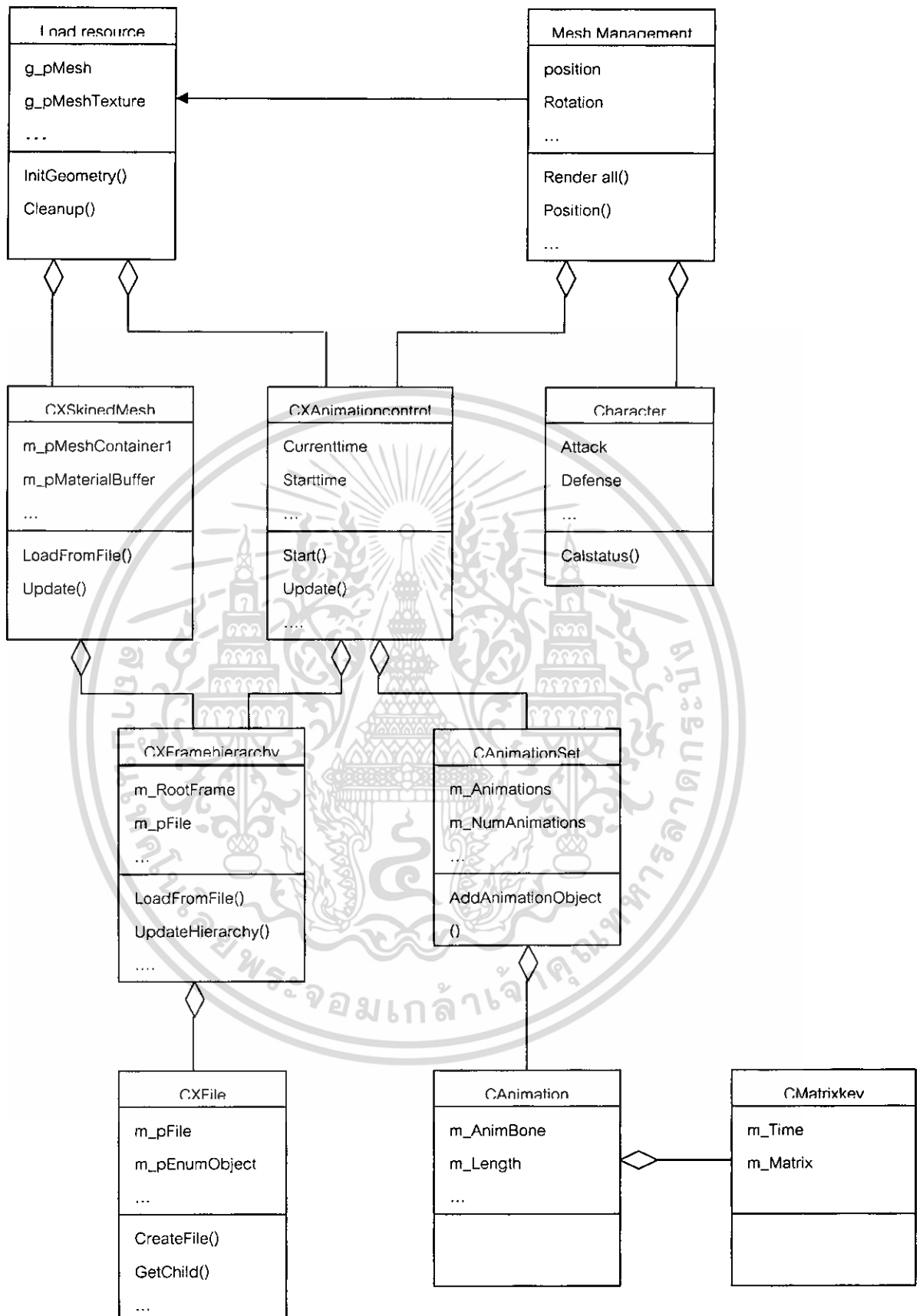


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ ไม่อนุญาตให้แก้ไขหรือใช้ประโยชน์ในการทำ  
 รูปที่ 24 Flow Chart แสดงการทำงานหลักๆของ Program Client  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Class ที่ใช้ใน Program Client

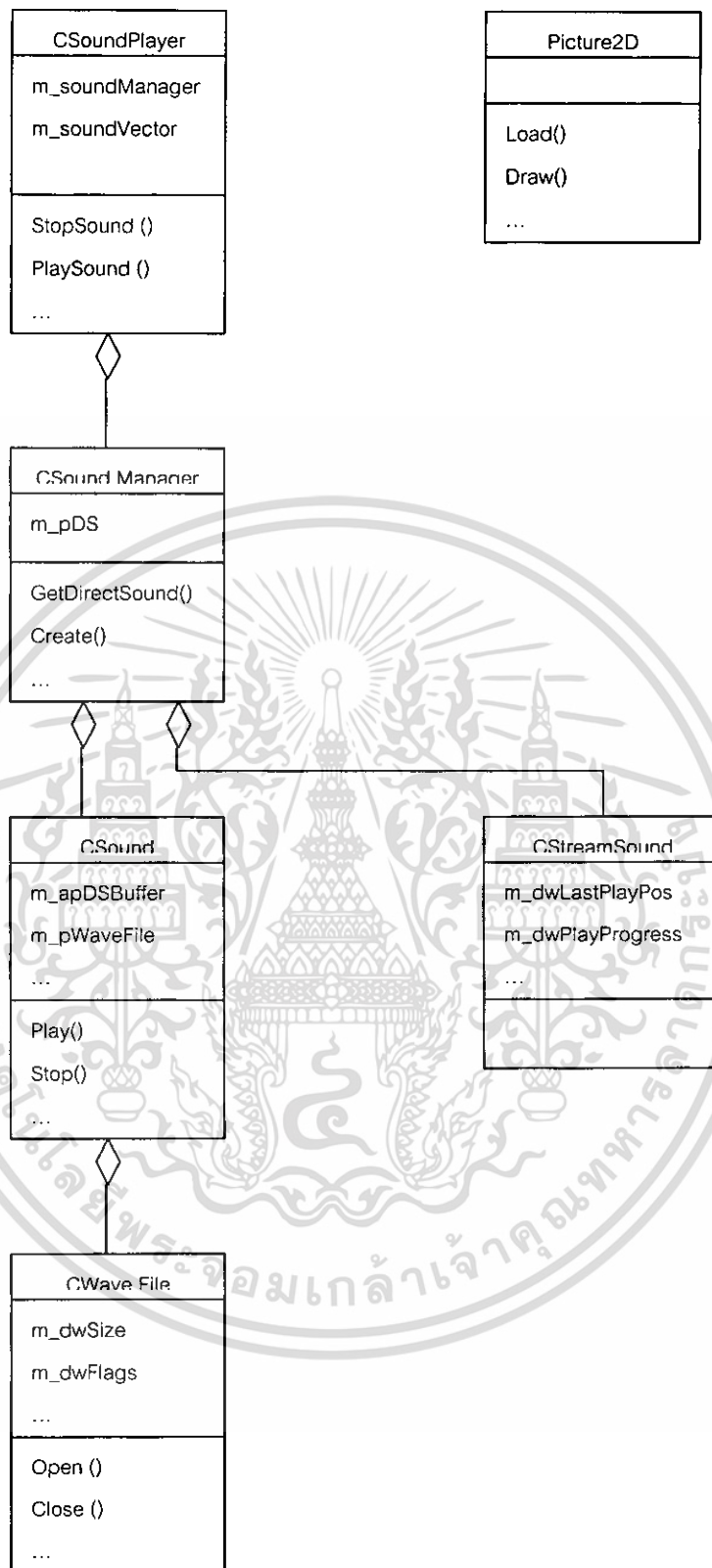
ในโปรแกรมนั้นมีการสร้าง Class และนำมาใช้อยู่หลาย Class ด้วยกัน โดยมีรายละเอียดดังต่อไปนี้

1. Load Resource Class ใช้ในการเก็บข้อมูลเกี่ยวกับตัวโมเดลสามมิติต่างๆ รวมไปถึงการทำ Animation อีกด้วย
2. Mesh Management Class เป็น Class ที่ใช้ในการควบคุม object ที่จะทำการแสดงผลสามมิติ เช่น แสดงตำแหน่งไหน แสดงFrameใด ขนาดเท่าไร เป็นต้น รวมไปถึงทำการแสดงผลบนหน้าต่างด้วย
3. CXSkinnedMesh Class เป็น Class ที่ใช้ในการจัดเก็บตัว Model
4. CXFramehierarchy Class ใช้ในการจัดการ Frame เก็บ Root Frame
5. CXFile Class เป็นตัวจัดการและติดต่อ DirectX File
6. CXAnimationcontrol Class เป็น Class ทำหน้าที่ในการควบคุม และUpdate Animation ทั้งหมด
7. CAnimationSet Class เป็น class ที่จัดเก็บ Animation ทั้งหมดของ Object 1 Object เอาไว้ ภายใน 1 AnimationSet ประกอบด้วย Animation หลายๆอันด้วยกัน
8. CAnimation Class เป็น Class ที่รวบรวมAnimation ไว้
9. CMatrixkey Class เป็นตัวระบุว่า Animation นี้ควรนำ Matrix ไตมาใช้
10. Character Class เป็น Class ที่ใช้เก็บสถานะต่างๆของตัวละคร 1 ตัวเช่น ค่าพลังในการโจมตี ตั้งรับ เลือดที่เหลืออยู่ เป็นต้น
11. CSoundPlayer Class ใช้ในการจัดการเสียงตั้งแต่ Load ไฟล์เสียงเข้ามา จนถึงเริ่มเปิดเสียงต่างๆ
12. Picture2D Class ใช้ในการแสดงผลภาพ 2 มิติต่างๆรวมถึง interfaceด้วย



รูปที่ 25 Class Diagram แสดงถึง Class ที่ใช้ในการจัดการที่มุดที่จะใช้ในการแสดงผล

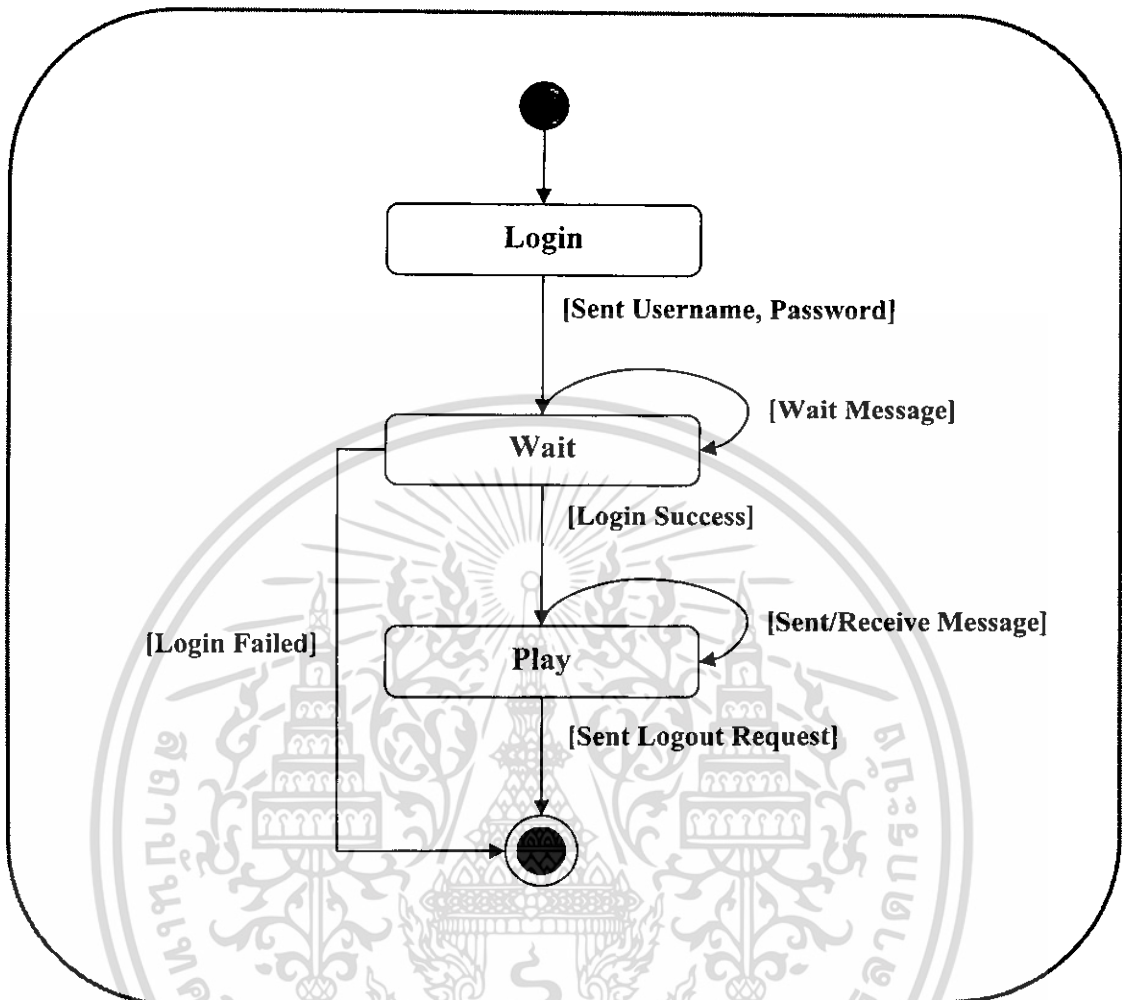
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 26 แสดง Class Diagram แสดงถึง Class ที่ใช้ในการจัดการเสียงและแสดงภาพ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## State การทำงานของ Client Application ในแง่การติดต่อกับระบบ



รูปที่ 27 แสดง State Diagram ของ Client Application

จากรูป จะเห็นได้ว่าเริ่มต้นโปรแกรมจะอยู่ในสถานะ 'Login' ซึ่งเป็นสถานะที่รอรับ Username และ Password จากผู้เล่นจากนั้นจะทำการส่ง Username และ Password ที่รับมาไปยัง Server เพื่อทำการตรวจสอบและร้องขอเข้าสู่ระบบ จากนั้นสถานะของ Client จะเปลี่ยนเป็น 'Wait' และรอการตอบรับจาก Server การตอบรับจาก Server มี 2 กรณี ดังนี้

- Message ที่ตอบกลับมา คือ 'Success' Client จะเปลี่ยนสถานะของตนเองเป็น 'Play' และทำการรับ-ส่งข้อมูลกับ Server เข้าเล่นเกมได้
- Message ที่ตอบกลับมาเป็นข้อความอื่นหรือ 'Time out' Client จะทำการ Terminate ตนเองทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

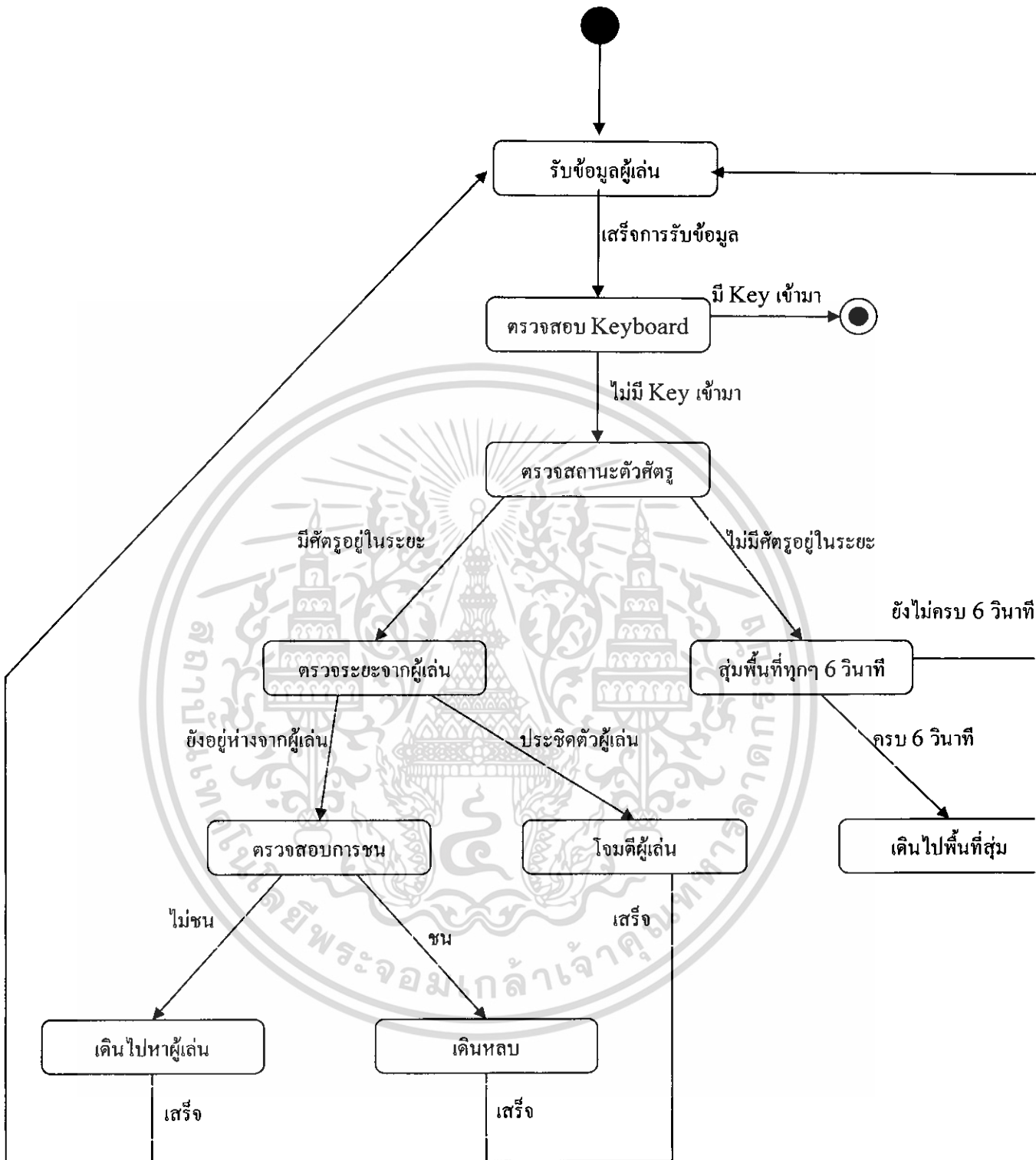
#### 4.4 การทำงานของ Program ควบคุมข้อมูลฝั่ง Database

##### เป้าหมายในการพัฒนา

1. โปรแกรมถูกพัฒนาเพื่อใช้ในการควบคุมการทำงานของตัวศัตรูต่างๆภายในเกม ดังนี้
  - ควบคุมการเคลื่อนไหวของตัวศัตรูให้เดินบนพื้นที่สุ่ม
  - ควบคุมการเคลื่อนที่ของตัวศัตรูให้เข้าโจมตีผู้เล่นเมื่อมีผู้เล่นอยู่ในระยะ
  - หากมีการต่อสู้กับผู้เล่นจะมีการคำนวณค่าสถานะต่างๆเพื่อลดเลือดของผู้เล่น
2. โปรแกรมถูกพัฒนาเพื่อใช้ในการกำหนดค่าสถานะรวมไปถึงข้อมูลต่างๆ ของตัวศัตรู

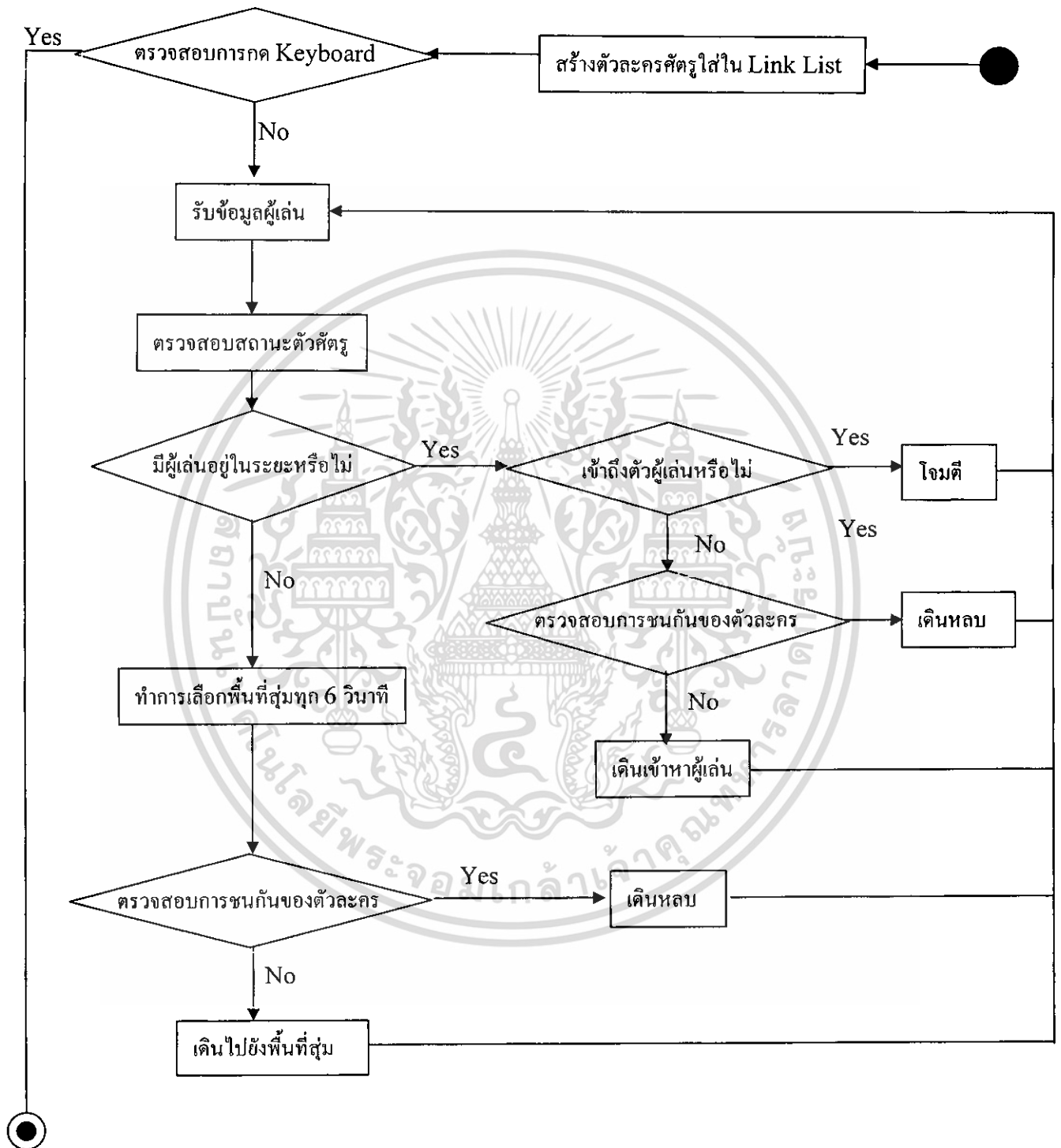
##### พัฒนาการของโปรแกรม

1. เขียนโปรแกรมที่ใช้ในการติดต่อกับ Database ได้
2. เขียนโปรแกรมเพื่อสร้างตัวละครศัตรูและกำหนดสถานะต่างๆให้กับตัวละครศัตรู
3. เขียนโปรแกรมเพื่อใช้ควบคุมตัวละครศัตรู ดังต่อไปนี้
  - ควบคุมการเคลื่อนไหวพื้นฐาน
  - ตรวจสอบการชนกันของตัวละครแต่ละตัวได้
  - ให้ศัตรูเดินในพื้นที่สุ่มในขณะที่ไม่มีผู้เล่นอยู่ในระยะ
  - ศัตรูสามารถเดินไปโจมตีผู้เล่นได้เมื่อผู้เล่นอยู่ในระยะ
4. เขียนโปรแกรมเพื่อใช้ติดต่อข้อมูลรับส่งกับ Database เพื่อให้สามารถทราบความเคลื่อนไหวของผู้เล่นแต่ละคนและให้ตัวละครศัตรูตอบโต้กลับไปได้อย่างถูกต้อง



รูปที่ 28 State Diagram ของ Software ที่ใช้ในการควบคุมตัวศัตรูฝั่ง Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 29 Flow Chart แสดงการทำงานของ Program ที่ใช้ในการควบคุมตัวศัตรู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.5 การทำงานของ Server Load Balancing (Server Programming)

##### เป้าหมายในการพัฒนา

1. โปรแกรมถูกพัฒนาขึ้นเพื่อทำการแยกแยะประเภทของผู้เล่นว่าอยู่ในสถานะ (Map) ไตและทำการแบ่งจำนวน Connection ให้แต่ละ Server มีจำนวนใกล้เคียงกัน

##### พัฒนาการของโปรแกรม

1. ออกแบบ Table ที่ใช้ในการเก็บข้อมูล Connection ที่มีในระบบ
2. เขียนโปรแกรมติดต่อกับ Database
3. พัฒนาโปรแกรมเพื่อทำการ Update ข้อมูลภายใน Table ที่ได้ออกแบบไว้
4. พัฒนาโปรแกรมในส่วน Balance Connection และทำการรับส่งข้อมูลกับ Client

#### 4.6 การทำงานของ Server (Server Programming)

##### เป้าหมายในการพัฒนา

1. โปรแกรมถูกพัฒนาขึ้นเพื่อทำการรับส่งข้อมูลกับ Client และ Update ข้อมูลลง Database

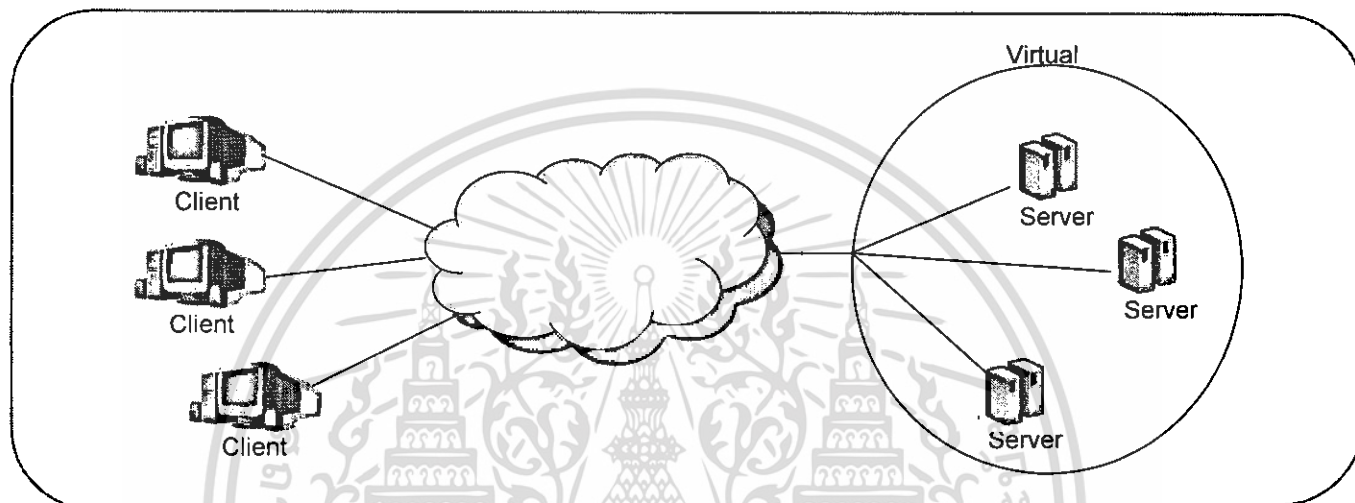
##### พัฒนาการของโปรแกรม

1. ออกแบบ Table ที่ใช้ในการเก็บข้อมูลสถานะต่างๆ ของผู้เล่น
2. เขียนโปรแกรมติดต่อกับ Database
3. พัฒนาโปรแกรมให้สามารถติดต่อกันระหว่าง Server และ Client
4. พัฒนาโปรแกรมเพื่อทำการ Update ข้อมูลภายใน Table ที่ได้ออกแบบไว้
5. พัฒนาโปรแกรมให้ Update สถานะของ Server (ตนเอง) เพื่อให้ Server Load Balancing รับรู้ว่า Server ไตยังคงทำงานอยู่

## บทที่ 5

# Network Load Balancing กับเกมออนไลน์สามมิติ

ในส่วนของ Server Side นั้นทางฝ่ายผู้จัดทำได้นำเทคโนโลยี Network Load Balancing (NLB) เข้ามาช่วยในการเพิ่มประสิทธิภาพให้กับ Server Side ซึ่งมีโครงสร้างพื้นฐาน ดังรูป



รูปที่ 30 แสดงหลักการทำงานของ Network Load Balancing

จากรูป

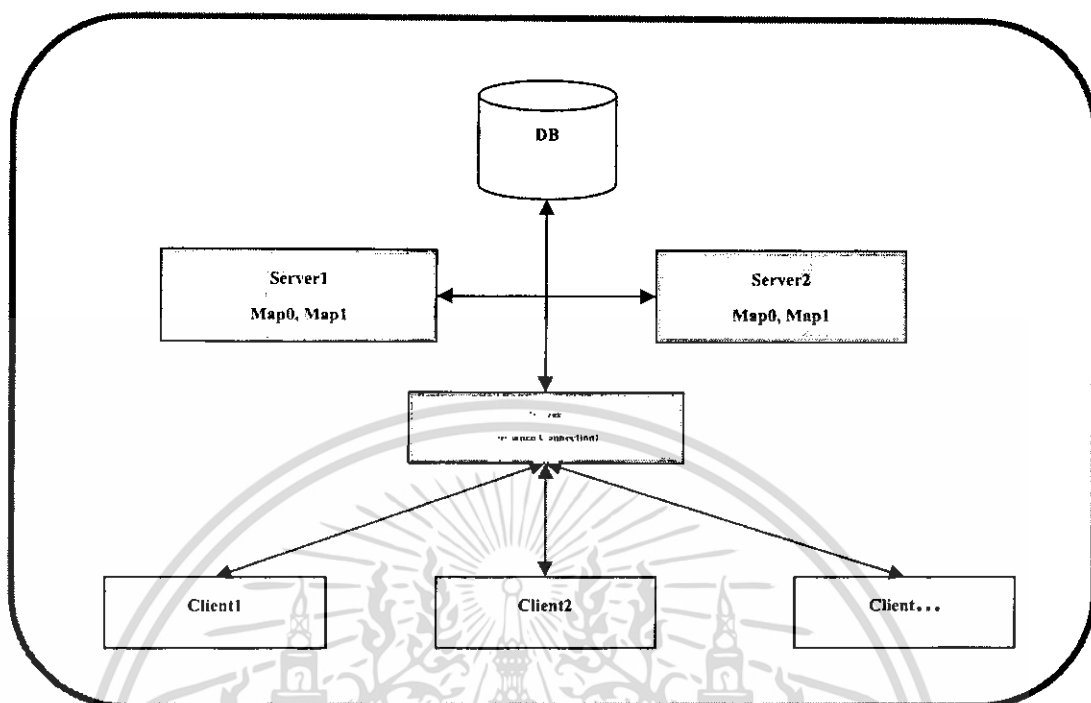
1. เมื่อเซิร์ฟเวอร์ได้รับคำร้องขอจากเครื่อง Client
2. มีการส่งกระจายคำร้องขอไปยังเซิร์ฟเวอร์ที่อยู่ใน Cluster เดียวกัน
3. จะมี Server หนึ่งที่สามารถจัดการกับคำร้องขอของเครื่อง Client ซึ่งผู้ใช้สามารถใช้งานได้อย่างต่อเนื่องตามปกติ

ซึ่งทางฝ่ายผู้จัดทำได้ทำการออกแบบรูปแบบของการ Balance Load ออกเป็น 3 แบบ ดังนี้

- 5.1 Server Connection
- 5.2 Map Connection
- 5.3 Server & Map Connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.1 Server Connection



รูปที่ 31 แสดงโครงสร้าง Server Connection

การ Balance Load รูปแบบนี้จะไม่สนใจสถานะใดๆของ Client/Player แต่จะทำการ Balance Connection ให้เข้า Server แต่ละเครื่องเท่าๆกัน เท่านั้น

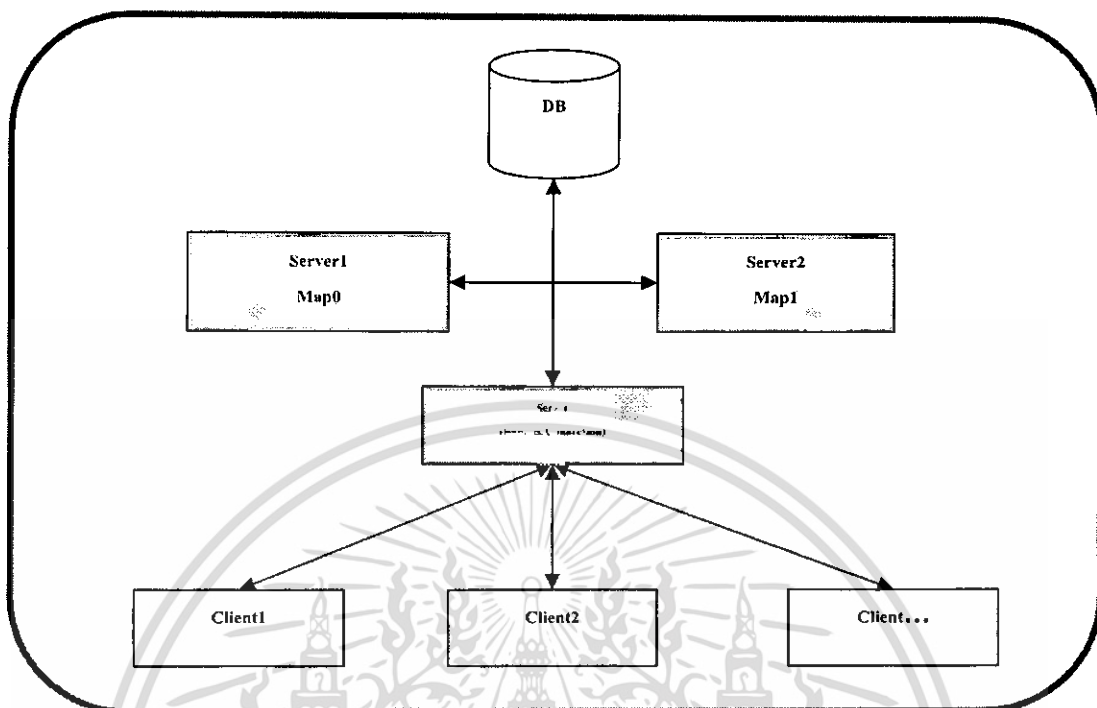
### ข้อดี

1. Server แต่ละเครื่องถูกใช้อย่างเต็มประสิทธิภาพ
2. หาก Server เครื่องใดเครื่องหนึ่งเกิดเหตุผิดพลาดขึ้นเกมยังสามารถให้บริการได้

### ข้อเสีย

1. Client แต่ละเครื่องจะได้รับข้อมูลเกินกว่าที่เครื่องตนเองต้องการ เนื่องจาก Server จะทำการส่งข้อมูลแบบ Multicast ดังนั้นข้อมูลที่ Client ได้รับนั้นจะได้รับทั้งของ Map0 และ Map1 ซึ่งในความเป็นจริงผู้เล่นสามารถอยู่ได้เพียง Map เดียว ณ เวลาใดเวลาหนึ่ง

## 5.2 Map Connection



รูปที่ 32 แสดงโครงสร้าง Map Connection

การ Balance Load รูปแบบนี้จะทำการตรวจสอบสถานะของ Player ก่อนว่าผู้เล่นอยู่ใน Map ไດ จากนั้นจะทำการติดต่อให้ Client ที่รัน Player นี้ทำการติดต่อไปที่ Server Map ที่ควบคุม Map ที่ผู้เล่นอยู่

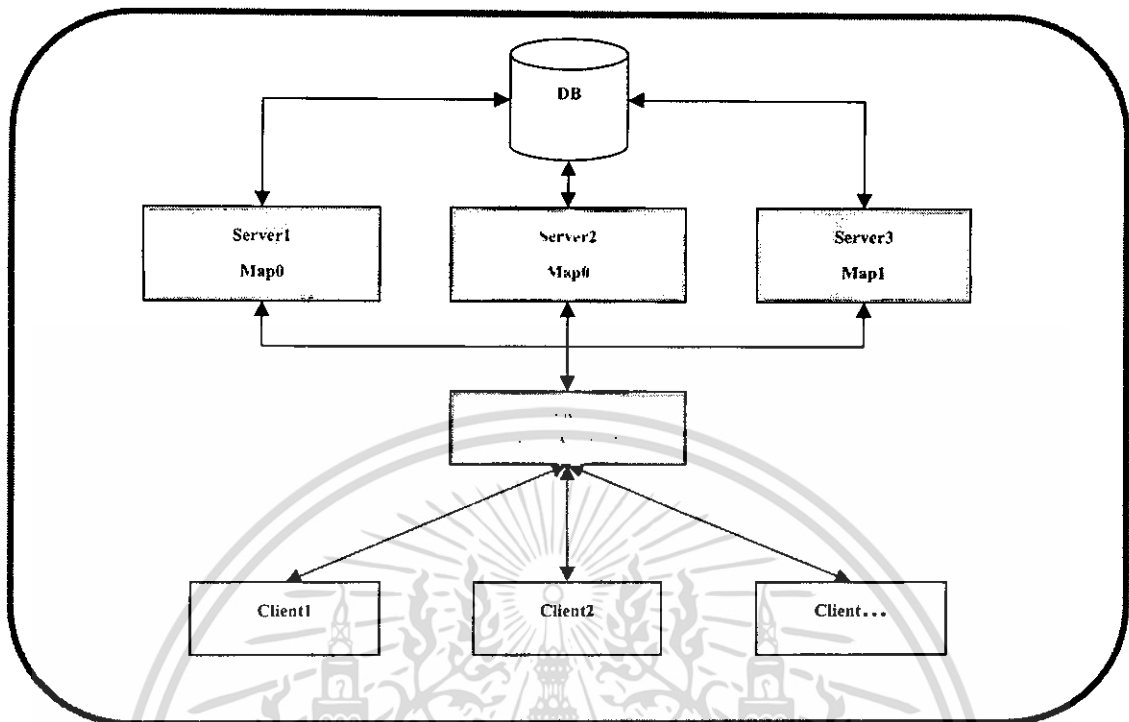
### ข้อดี

1. Client แต่ละเครื่องจะได้รับข้อมูลที่ตนเองต้องการเท่านั้น

### ข้อเสีย

1. Server แต่ละเครื่องจะไม่ถูกใช้งานอย่างเต็มประสิทธิภาพ ยกตัวอย่างในกรณีที่มีผู้เล่นอยู่ใน Map0 ทุกคนจะเห็นได้ว่า Server2(Map1) จะไม่ถูกใช้งานใดๆ
2. ในกรณีที่ Server ไດเกิดความผิดพลาดขึ้นเกมที่ Map นั้นจะไม่สามารถใช้งานได้

### 5.3 Server & Map Connection



รูปที่ 33 แสดงโครงสร้าง Server & Map Connection

การ Balance Load รูปแบบนี้เป็นการนำจุดเด่นของแบบ Server Connection และ Map Connection มาผสมผสานกัน

#### ข้อดี

1. Server แต่ละเครื่องถูกใช้อย่างเต็มประสิทธิภาพ
2. หาก Server เครื่องใดเครื่องหนึ่งเกิดเหตุผิดพลาดขึ้นเกมยังสามารถให้บริการได้
3. Client แต่ละเครื่องจะได้รับข้อมูลที่ตนเองต้องการเท่านั้น

## บทที่ 6

### ผลการทดลอง

#### 6.1 เนื้อหาของเกม

เป็นเกมแนว 3D Action RPG โดยเน้นการผจญภัยของตัวละครในโลกแฟนตาซี ตัวละครจะต้องทำการต่อสู้กับศัตรูเพื่อทำการพัฒนาค่าประสบการณ์ โดยใช้วิธีวิ่งฟันด้วยอาวุธต่างๆ หรือใช้อาวุธ เวทมนตร์ จากระยะไกลในการโจมตี ในขณะที่เดียวกับจะมีระบบไอเทมเพื่อใช้ในการสนับสนุนต่างๆ เพื่อเพิ่มความหลากหลายในการเล่นเกมนิ่งให้มากยิ่งขึ้น

#### 6.2 จุดเด่นของเกม

เป็นเกมการต่อสู้ผจญภัยบนโลกสามมิติ โดยทำการต่อสู้เพื่อรวบรวมค่าประสบการณ์และมีพัฒนาการของตัวละคร และผู้เล่นสามารถผจญภัยไปพร้อมกับผู้เล่นคนอื่นๆ ได้พร้อมๆ กัน

#### 6.3 การดำเนินเกม

ซึ่งในช่วงเริ่มต้นเกมผู้เล่นแต่ละคนสามารถเลือกตัวละครที่ต้องการจะเล่นได้ ซึ่งในแต่ละอาชีพนั้นจะมีค่าความสามารถและจุดเด่นแตกต่างกันไป เพื่อเพิ่มความหลากหลายให้กับตัวเกม การที่จะสามารถพัฒนาตัวละครได้นั้นผู้เล่นจำเป็นที่จะต้องต่อสู้กับศัตรู เพื่อเพิ่มค่าประสบการณ์ ในเกมจะมีตัวกำหนดระดับของผู้เล่นเรียกว่า level ทุกครั้งที่ผู้เล่นเก็บค่าประสบการณ์จนสามารถเลื่อนระดับ level ได้ผู้เล่นจะได้รับ point เพื่อนำไปเลือกพัฒนาตัวละครได้เองตามที่ต้องการ ในการต่อสู้เพื่อเก็บค่าประสบการณ์นั้นผู้เล่นสามารถต่อสู้คนเดียวหรือร่วมต่อสู้ไปพร้อมๆ กับผู้เล่นคนอื่นก็ได้ เพื่อเพิ่มความสนุกสนานในการเล่นเกม

## 6.4 ตัวละคร

ตัวละครที่ผู้เล่นสามารถควบคุมเพื่อผจญภัยไปบนโลกสามมิติได้นั้น มีหลักๆ 2 ตัว ดังนี้



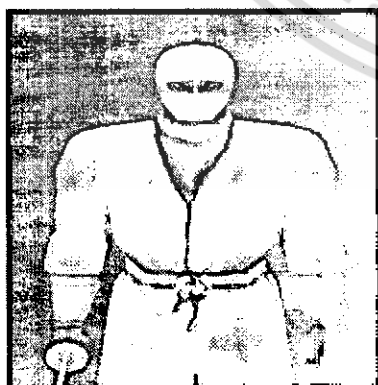
1. นักดาบ เป็นผู้เล่นที่มีความสามารถในการต่อสู้ประชิดตัวได้ดีที่สุดในเกม สามารถโจมตีได้อย่างรุนแรงในขณะเดียวกันก็มีการตั้งรับที่ดีด้วย จึงเรียกได้ว่าเป็นตัวละครที่มีความสมดุลมากที่สุดตัวหนึ่งในเกม



2. นิจจา เป็นตัวละครที่มีความว่องไวสูงเมื่อเทียบกับตัวละครอื่นๆ ในเกม สามารถหลบหลีกการโจมตีของศัตรูได้ดี และสามารถตอบโต้กลับไปได้อย่างแม่นยำ แต่มีข้อเสียคือค่อนข้างอ่อนแอไม่สามารถทนการโจมตีได้มากนัก นิจจาดังนี้จึงจัดเป็นตัวละครที่มีทักษะดีที่สุดในเกม

## 6.5 คู่ต่อสู้ภายในเกม

ภายในเกมยังมีศัตรูที่เราต้องทำการต่อสู้ด้วยเพื่อทำการเก็บค่าประสบการณ์ ดังนี้



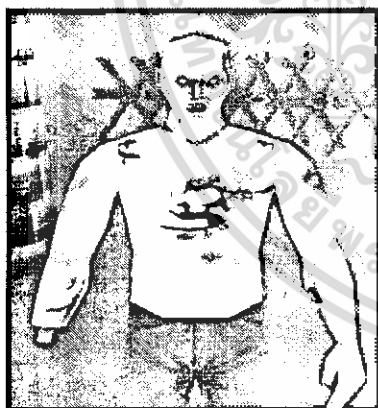
1. นินจาขาว เป็นนินจาฝึกหัดยังไม่ค่อยมีความชำนาญในการต่อสู้มากนัก ผู้เล่นในระดับแรกๆ สามารถสู้กับศัตรูตัวนี้เพื่อเก็บค่าประสบการณ์ได้เป็นอย่างดี



2. นินจาแดง เป็นนินจาที่ผ่านการฝึกมาแล้วมีทักษะและมีมือมากกว่านินจาขาว สามารถโจมตีได้อย่างรวดเร็วแม่นยำ เหมาะสมกับผู้เล่นระดับกลางที่จะมาต่อสู้ด้วย



3. ซอมบี้ขาว เป็นผีดิบที่ไม่มีความรู้สึก แต่สามารถโจมตีได้อย่างหนักหน่วงรุนแรง โดยใช้เท้าและมือทั้งสองในการโจมตี ในขณะที่เดียวกันก็มีความทนทานสูงทำให้ไม่สามารถเอาชนะได้ง่ายๆ ศัตรูตัวนี้จึงเหมาะกับผู้เล่นที่มีการพัฒนาแล้วเท่านั้น

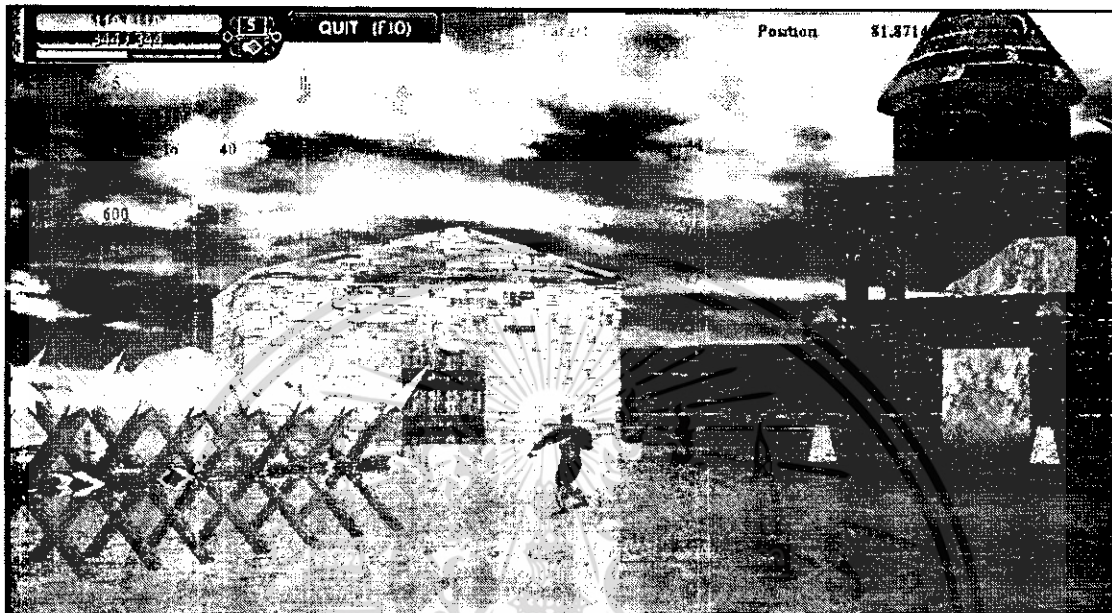


4. ซอมบี้แดง เป็นผู้นำฝูงผีดิบ มีความโหดเหี้ยม และแข็งแรงอย่างมาก ผู้เล่นในระดับทั่วไปไม่สามารถรับมือได้ง่ายๆ จึงควรใช้ผู้เล่นหลายๆคนช่วยกันต่อสู้จึงจะสามารถจัดการได้

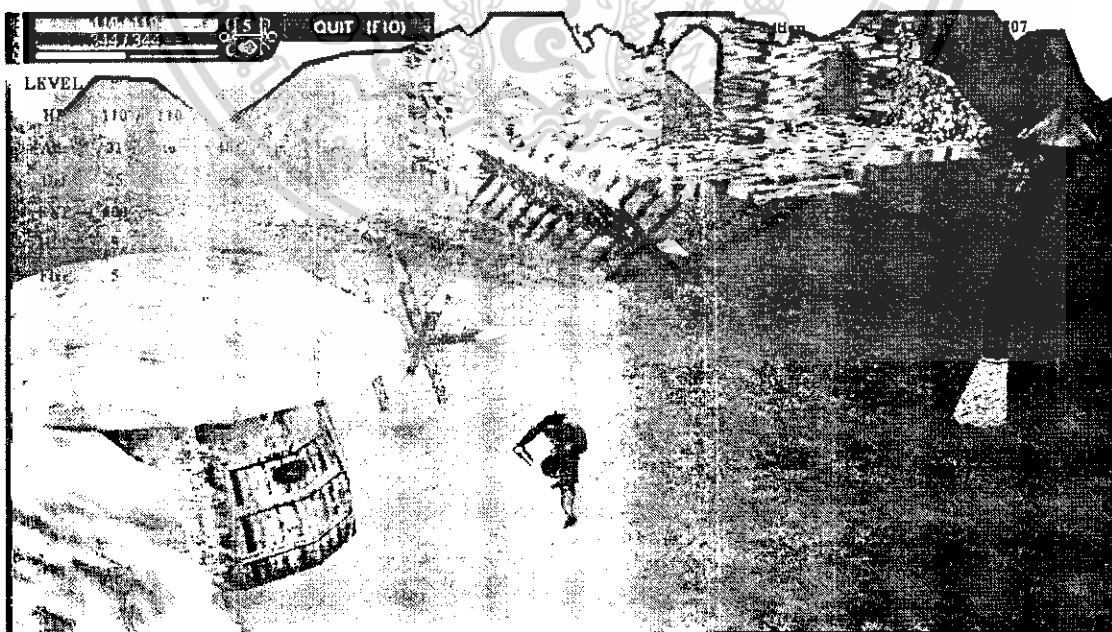
## 6.6 สถานที่ต่างๆ ภายในเกม

มีด้วยกันทั้งสิ้น 2 จากด้วยกัน ดังนี้

1. จากภายในกำแพงเมือง เป็นสถานที่ภายในเมืองเล็กๆ มีบ้านพัก โบสถ์ และหอคอยตั้งอยู่



2. จากภายในป่า หากผู้เล่นเดินออกไปจากกำแพงเมืองก็จะเข้าสู่ป่าซึ่งเต็มไปด้วยซากปรักหักพังและฝูงซอมบี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.7 ค่าสถานะ (Status)

1. ค่าสถานะหลักที่ผู้เล่นสามารถเลือกปรับแต่งได้เองเมื่อ มีการเพิ่มเลเวล

STR (Strength) ค่าความแข็งแกร่ง เป็นเป็นค่าที่มีความสำคัญอย่างมากโดยเฉพาะกับพวกนักรบสายต่อสู้ประชิดตัวเพราะเป็นค่าที่มีผลต่อพลังโจมตี ค่าป้องกัน เลือด รวมไปถึงน้ำหนักที่รับได้ด้วย

AGI ค่าความว่องไว เป็นค่าที่เพิ่มความรวดเร็วในการหลบหลีกศัตรูรวมไปถึงความเร็วในการโจมตีด้วย

DEX ค่าความแม่นยำ เป็นค่าที่ส่งผลถึงค่า Hit ซึ่งเป็นตัวบ่งบอกว่าจะสามารถโจมตีโดนหรือไม่ ในขณะที่เดียวกันต่อส่งผลถึงความรุนแรงในการโจมตีอีกด้วย

HP เลือด เป็นค่าที่ทำการเพิ่มเลือดของผู้เล่นให้มากยิ่งขึ้นทำให้ผู้เล่นมีความอึดสามารถต่อสู้กับศัตรูได้นานยิ่งขึ้นเหมาะกับพวกต่อสู้ประชิดตัวมากกว่า

2. ค่าสถานะอื่นๆ ค่าสถานะเหล่านี้จะขึ้นอยู่กับสถานะหลักที่ผู้เล่นปรับแต่งไว้

Atk ค่าพลังความรุนแรงในการโจมตี

Def ค่าป้องกัน

Flee ค่าความว่องไวในการหลบการโจมตี

Hit ค่าความแม่นยำในการโจมตี

Blood Max ค่าเลือดสูงสุด

## 6.8 ค่าพื้นฐานและสูตรที่ใช้ในการคำนวณ (ในแต่ละ Level)

$$\text{Atkmin} = (3 * \text{str}) + (2 * \text{lv} * (\text{dex} / 50))$$

$$\text{atkmax} = (3 * \text{str}) + (2 * \text{lv})$$

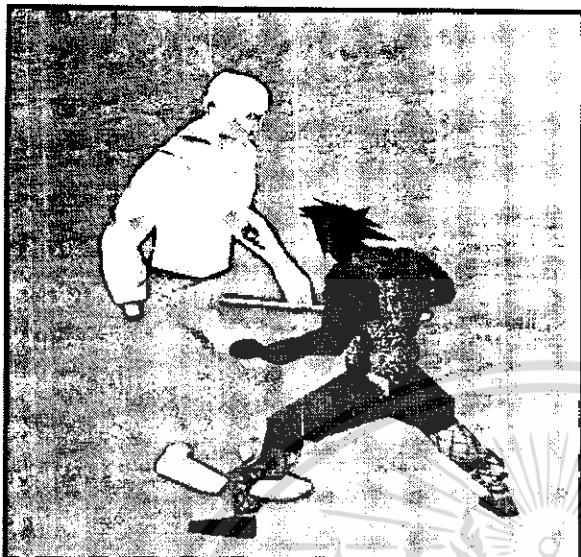
$$\text{defmin} = 2 * \text{str} + \text{lv}$$

$$\text{bloodmax} = 5 * \text{str} + 8 * \text{lv} + 10 * \text{hp}$$

$$\text{hit} = \text{dex}$$

$$\text{flee} = \text{agi}$$

## 6.9 การต่อสู้ภายในเกม



ในการต่อสู้กับศัตรูต่างๆภายในเกมผู้  
สามารถทำได้โดย เคลื่อนที่เข้าไปหา  
เป้าหมายแล้วทำการโจมตี เท่านั้น  
ก็สามารถสร้างความเสียหายให้กับเป้าหมาย  
ได้ แต่ในขณะเดียวกันหากศัตรูตอบโต้  
กลับมาเลือดที่เรามีอยู่ก็จะลดลงเช่นเดียวกัน  
หากเลือดของเราหมดลงก็จะตาย ไปเกิดใหม่  
ที่จุดเกิดทันที

การคำนวณที่ใช้คิดในการต่อสู้

1. ความเสียหายที่ได้ = ค่าพลังโจมตีของเรา(Atk) - ค่าการตั้งรับของเป้าหมาย(Def)
2. โอกาสที่จะโจมตีพลาด = ค่าหลบหลีกเป้าหมาย(Flee) - ค่าความแม่นยำของเรา(Hit)
3. โอกาสที่เราจะหลบได้ = ค่าหลบหลีกของเรา(Flee) - ค่าความแม่นยำของเป้าหมาย(Hit)

## 6.10 การควบคุมภายในเกม

ภายในเกมเราสามารถควบคุมตัวละครได้ผ่านทาง Keyboard โดยแต่ละปุ่มมีการทำงาน  
ดังนี้

ส่วนควบคุมตัวละคร

1. up down left right (ปุ่มลูกศร) ใช้ในการควบคุมตัวละครให้เดินไปในทิศทางต่างๆ  
โดยตัวละครสามารถเดินได้ใน 8 ทิศทาง
2. ปุ่ม A เป็นปุ่มที่ใช้สั่งให้ตัวละครทำการโจมตีไปข้างหน้า

ส่วนควบคุมมุมมอง

1. ปุ่ม E D S F เป็นปุ่มที่ใช้ในการควบคุมมุมมองให้เคลื่อนที่ไปในทิศทาง บน ล่าง ซ้าย  
ขวา ของตัวละครตามระดับ
2. ปุ่ม T G เป็นปุ่มที่ใช้ในการซูมเข้าและออกจากตัวละคร

การออกจากเกม

ในการออกจากเกมสามารถทำได้โดยกดปุ่ม F10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.11 Interface ภายในเกม

ภายในเกมมีการใช้ Interface ในการแสดงสถานะของผู้เล่นโดยมีรายละเอียดดังต่อไปนี้



1. แถบสีแดงบนสุดแสดงถึงแถบเลือดของผู้เล่นที่ยังเหลืออยู่หากแถบนี้เต็มแสดงว่าเลือดของผู้เล่นเต็มอยู่
2. แถบสีน้ำเงิน แสดงถึงมานา (ค่าที่ใช้ในการร่ายเวทมนตร์) รองรับการพัฒนาในอนาคต
3. แถบสีเขียวล่างสุด แสดงถึงค่าประสบการณ์ที่ผู้เล่นกำลังมีอยู่หากแถบเต็ม ผู้เล่นจะได้เลื่อนระดับ
4. ตัวเลขทางซ้ายขวามือ เป็นตัวเลขแสดงระดับของผู้เล่น (Level)

### หน้าต่างแสดงสถานะของผู้เล่น

นอกจากนั้นภายในเกมยังมีหน้าต่างที่ใช้ในการแสดงสถานะต่างๆของผู้เล่นโดยละเอียดอีกด้วย โดยมีรายละเอียดดังต่อไปนี้



แสดงระดับของผู้เล่น	ค่าความแข็งแรง(Str)
แสดงถึงเลือดที่มีอยู่ / เลือดเต็ม	ค่าความว่องไว (Agi)
แสดงถึงพลังโจมตีน้อยสุดถึงมากที่สุด	ค่าความแม่นยำ (Dex)
แสดงค่าการป้องกัน	เลือด (HP)
แสดงความแม่นยำ	
แสดงค่าการหลบหลีก	Point ที่เหลืออยู่

\*\* ทางขวาคือสถานะพื้นฐาน ทางซ้ายคือค่าสถานะที่ผ่านการคำนวณแล้ว

\*\*\*Point คือ ค่าที่สามารถนำไปพัฒนาค่าสถานะพื้นฐานทางขวามือของหน้าต่างได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.12 กฎต่างๆ ภายในเกม

1. ก่อนผู้เล่นจะเข้าสู่ตัวเกมจะต้องทำการ Login เสียก่อน โดยผู้เล่นจะต้องทำการระบุ Username และ Password เพื่อระบุตัวตน
2. ภายในเกมมีแผนที่ทั้งหมด 2 แห่งด้วยกันผู้เล่นสามารถเดินข้ามจากแผนที่หนึ่งไปอีกแผนที่หนึ่งได้ในจุดที่กำหนดไว้
3. ผู้เล่นสามารถดูค่าสถานะต่างๆ ของตัวละครได้ที่ Interface ที่มีมบ่นซ้ายสุดของจอ
4. ผู้เล่นสามารถดูสถานะโดยละเอียดได้บนหน้าแสดงสถานะโดยสามารถเรียกดูได้โดยการกดปุ่ม F1
5. หากผู้เล่นมีการเพิ่มระดับ ผู้เล่นจะมีค่า Status เพิ่มมากขึ้นเองอัตโนมัติ ในขณะเดียวกันก็จะได้รับ Point 2 point ให้ผู้เล่นสามารถนำไปเลือกพัฒนาตัวละครได้เองตามแบบที่ผู้เล่นต้องการ
6. ผู้เล่นสามารถเลือกใช้ Point เพื่อพัฒนาตัวละครได้ที่หน้าแสดงสถานะ (ปุ่มF1)
7. ผู้เล่นสามารถควบคุมมุมมองได้ตามใจชอบโดยมุมมองสามารถหมุนได้ 4 ทิศทาง และสามารถทำการ ซุมเข้าออกจากตัวละครได้
8. ในการต่อสู้จะคิดค่าพลังในการโจมตีโดยทำการสุ่มค่าพลังการโจมตีระหว่างค่าการโจมตีสูงสุดและต่ำสุด
9. ค่าความเสียหายที่ทำได้ขึ้นอยู่กับพลังการโจมตีและป้องกันของแต่ละฝ่าย
10. ในกรณีที่ค่าความเสียหายต่ำมากๆค่าความเสียหายที่จะเกิดขึ้นน้อยที่สุดคือ 5
11. ผู้เล่นรวมถึงศัตรูสามารถหลบการโจมตีได้โดยโอกาสที่จะสามารถหลบการโจมตีพันหรือไม่ขึ้นอยู่กับค่าการหลบหลีกและค่าความแม่นยำของทั้งสองฝ่าย
12. ในกรณีที่ผู้เล่นตายในการต่อสู้ ผู้เล่นจะถูกนำไปยังจุดที่กำหนดไว้ พร้อมทั้งฟื้นฟูค่าสถานะต่างๆจนเต็ม แต่จะถูกหักค่าประสบการณ์ลง 5 เปอร์เซ็นต์
13. ผู้เล่นสามารถฟื้นฟูตัวเองได้ โดยทำการยืนอยู่พักอยู่เฉยๆเลือดของผู้เล่นจะค่อยๆเพิ่มขึ้นเอง
14. ภายในเกมมีศัตรูอยู่หลายประเภทด้วยกัน ศัตรูแต่ละตัวมีค่าความสามารถที่แตกต่างกันออกไปผู้เล่นจึงควรเลือกต่อสู้กับศัตรูที่เหมาะสมกับตัวเอง
15. หากผู้เล่นต้องการออกจากเกม จะต้องทำการ Logout ออกจากเกมอย่างถูกวิธี โดยสามารถออกจากเกมโดยการกดปุ่ม F10

### 6.13 ค่าพื้นฐานของแต่ละตัวละคร

ตัวละครผู้เล่น ค่าพื้นฐานของผู้เล่นตอนเริ่มต้น

	STR	AGI	DEX	HP
นักดาบ	10	5	5	2
นินจา	5	10	7	0

\*\* ค่าสถานะเหล่านี้ของผู้เล่นจะถูกนำไปคำนวณเป็นค่าสถานะต่างๆ ตามสูตรที่มี

ตัวละครศัตรู ค่าต่างๆรวมไปถึงค่าประสบการณ์ที่ศัตรูแต่ละตัวให้

	BloodMax	Atkmax	Atkmin	Def	Hit	Free	exp
นินจาขาว	50	30	35	20	0	10	10
นินจาแดง	70	40	50	35	5	15	25
ซอมบี้ขาว	120	50	70	50	10	20	120
ซอมบี้แดง	300	60	80	55	50	50	700

\*\*\*ศัตรูเหล่านี้ไม่มีค่าพื้นฐานจะมีแต่ค่าสถานะเท่านั้น

## 6.14 การทดลองการทำงานของ Game Server

### วัตถุประสงค์การทดลอง

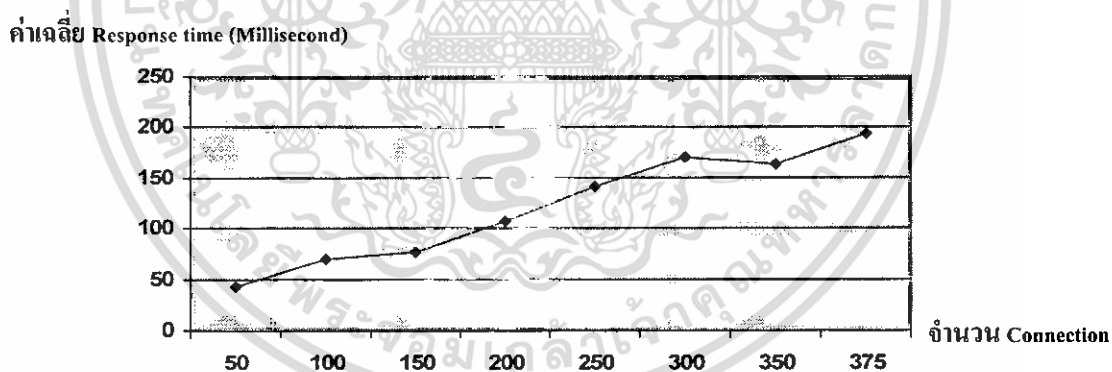
1. เพื่อศึกษาจำนวน Connection ที่ Server สามารถรองรับได้
2. เพื่อศึกษาผลกระทบด้านต่างๆในการ Run โปรแกรม Game Server

### ขั้นตอนการทดลอง

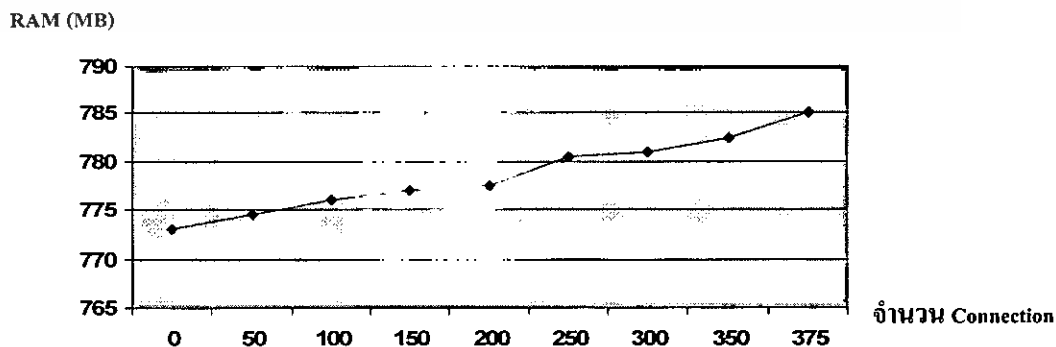
1. ทำการสร้างโปรแกรม Client Console ที่มีการรับ-ส่งข้อมูลเหมือน Game Application ที่ใช้ในงาน
2. ทำการสร้างโปรแกรมที่ทำหน้าที่ในการสร้าง Multi Process เพื่อจำลองจำนวน Client Console มากๆได้ในหนึ่งเครื่อง
3. ทำการรันโปรแกรม Server ที่ใช้งานจริงบน CPU: AMD Sempron(TM) 2800+ 1.68 GHz , RAM: 1.5 GB
4. ทำการรันโปรแกรม Multi Process ที่สร้างขึ้นและเก็บผลการทดลอง

### ผลการทดลอง

จากการทดลองพบว่า Server ทำการรันสามารถรองรับ Client ได้ประมาณ 375 Connections ซึ่งสามารถดูแนวโน้มได้จากกราฟ ดังนี้



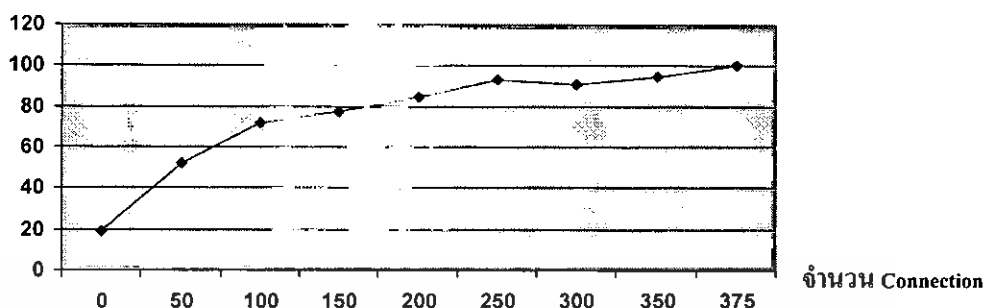
รูปที่ 34 แสดงความสัมพันธ์ระหว่างจำนวน Connection กับ Response Time



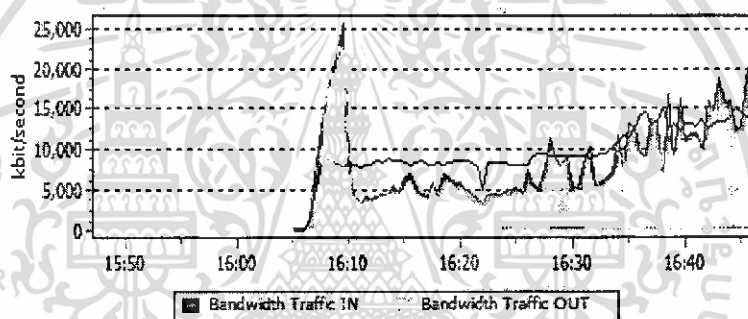
รูปที่ 35 แสดงความสัมพันธ์ระหว่างจำนวน Connection กับ RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CPU Usage (100%)



รูปที่ 36 แสดงความสัมพันธ์ระหว่างจำนวน Connection กับ CPU Usage



รูปที่ 37 แสดงความสัมพันธ์ปริมาณข้อมูลเข้า-ออก Server

### สรุปผลการทดลอง

จากการทดลองจะเห็นได้ว่าจำนวน Connection ที่รับได้มีจำนวนเพียง 375 Connections เท่านั้น ซึ่งเกิดจากการที่ CPU ประมวลผลไม่ทันและโปรแกรมที่เราใช้ในการทดลองนั้นมีลักษณะการรับ-ส่ง ข้อมูลตลอดเวลา (Worst Case) อีกทั้งเครื่องที่ทำการรัน Game Server นั้นมีความสามารถค่อนข้างต่ำ

## บทที่ 7

### บทสรุป

#### 7.1 ปัญหาและอุปสรรคที่เกิดจากการดำเนินงาน

1. ปัญหาที่เกิดจากการเลือก Graphic Tool เนื่องจาก Tool ที่ใช้ในการพัฒนาภาพสามมิติ ซึ่งในปัจจุบัน Tool ที่ใช้มีความหลากหลายและมีจุดเด่นจุดด้อยแตกต่างกัน จึงเป็นการยากที่จะเลือก Tool ที่เหมาะสมกับความต้องการของงานได้อย่างเหมาะสม
2. ปัญหาในส่วนของ Hardware เนื่องจากชิ้นงานที่จัดทำขึ้นเป็นลักษณะของภาพสามมิติ จึงมีความจำเป็นที่จะต้องใช้เครื่องมือที่มีความสามารถสูง โดยเฉพาะส่วนเช่น การ์ดจอ, CPU, RAM เป็นต้น
3. ปัญหาในส่วนของ การติดต่อระหว่าง Client-Server เนื่องจากการทำงานโดยส่วนใหญ่ ของเกมออนไลน์ Server Side เป็นส่วนที่มีความสำคัญและมีการ Interrupt การทำงาน เป็นจำนวนมากประกอบกับต้องทำการส่งข้อมูลที่เปลี่ยนแปลงให้กับ Client Side ตลอดเวลา ดังนั้นการ Synchronous Time ระหว่าง Client-Server จึงมีความจำเป็นมาก เช่น หากส่งข้อมูลช้าภาพที่ได้จะเกิดการกระตุก หรือในทางตรงกันข้ามการส่งข้อมูลด้วยความถี่ที่สูงขึ้นจะทำให้ Client Side ไม่สามารถประมวลผล Message ที่รับมาได้ทัน
4. ปัญหาในส่วนของ การออกแบบการบริหารจัดการข้อมูลภายในตัวโปรแกรม หากออกแบบไม่ดี อาจส่งผลกระทบการเขียนโปรแกรมเกิดความสับสน เช่น การตรวจสอบหาตัวแปรบางตัวไม่พบ เป็นต้น ดังนั้นจึงควรพัฒนาเป็น Class รวมไปถึงแยกส่วนการทำงานต่างๆ ออกจากกันอย่างชัดเจน เพื่อให้ง่ายต่อการเข้าใจและจดจำ

## 7.2 แนวทางในการพัฒนาต่อ

### 1. Graphic

- Transparent (การทำให้สามารถมองเห็นทะลุผ่านพื้นผิวบางส่วนได้) เช่น ต้นไม้
- Motion Skybox (เพิ่มการเคลื่อนไหวของท้องฟ้า) เพื่อความสมจริง
- ทำการปรับปรุงระบบแสง Light, Fog and Shading
- เพิ่ม Effect ในการใช้เวทย์มนตร์ต่างๆ หรือใช้สิ่งของภายในเกม

### 2. Game Detail

- เพิ่มจำนวนแผนที่ภายในเกมเพื่อให้มีความหลากหลายของสถานที่มากขึ้น
- เพิ่มประเภทของตัวละครศัตรูรวมไปถึงแบ่งระดับของศัตรูด้วย
- เพิ่มในส่วนของ Item และ Skill ต่างๆภายในเกมส์
- เพิ่ม Chat System ทำให้ผู้เล่นสามารถทำการสนทนากันได้ภายในเกม
- เพิ่มตัวละครที่เล่นไปพร้อมกับผู้เล่นโดยใช้ (AI) เข้ามาช่วย ทำให้ผู้เล่นสามารถเล่นเกมได้อย่างสนุกสนานมากยิ่งขึ้น
- ขณะต่อสู้สามารถทำให้ผู้เล่นตรวจสอบเลือดที่เหลือของศัตรูหรือผู้เล่นอื่นได้
- ขณะต่อสู้สามารถตรวจค่าความเสียหายที่ทำได้
- พัฒนาเสียงเพิ่มเข้าไปในตัวเกมเพื่อเพิ่มอรรถรสในการเล่น
- พัฒนาระบบ Party เพื่อให้ผู้เล่นสามารถเล่นร่วมกันได้เป็นกลุ่ม

## บรรณานุกรม

- [1] Alan Thorn "DirectX Graphics: The Definitive Guide to Direct3D"
- [2] Jason Jurecka " Working with the DirectX .X File Format and Animation in DirectX 9.0" [Online] .Available : <http://www.gamedev.net/reference/articles/article2079.asp>
- [3] Itsara Konsombut "DirectX 9.0 with C++"
- [Online] .Available : <http://www.gamedevx.com/article.php?type=3>
- [4] ชัยดำรงค์ อุทธิรัมย์ "Direct3D พลังพัฒนาแห่งเกมสามมิติ" กรุงเทพฯ : สามย่าน.COM, 2544 ISBN 974-88353-0-8
- [5] กัทรารุร เหมือนทอง "เว็บไซต์โหลดบาลลาลจริง"

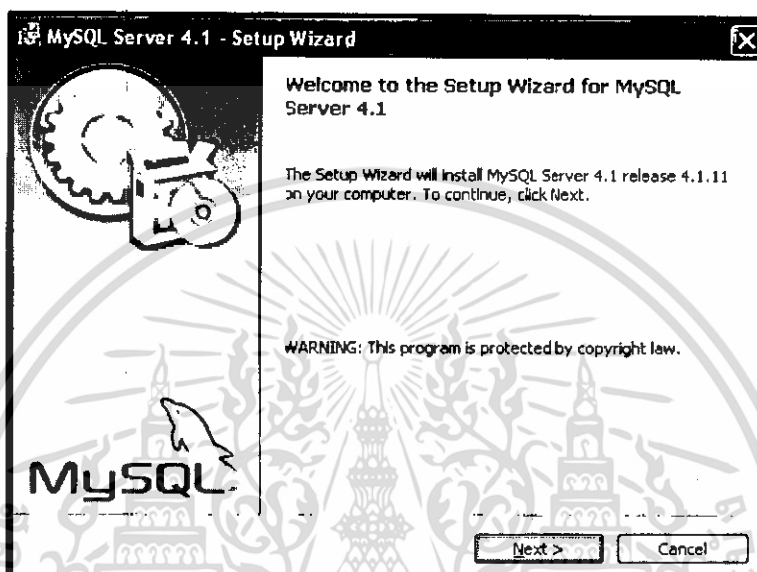


## ภาคผนวก ก

### คู่มือการติดตั้ง

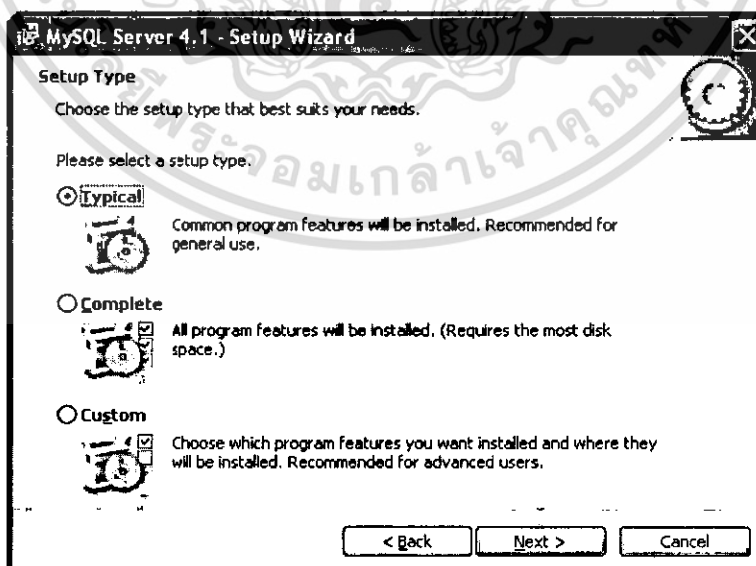
#### 1. การติดตั้ง MySQL Server 4.1

##### 1.1 ปรากฏหน้าจอ Welcome to the Setup Wizard for MySQL Server 4.1 คลิกปุ่ม Next



รูปที่ ก.1.1 แสดงหน้าเริ่มต้นการลงโปรแกรม

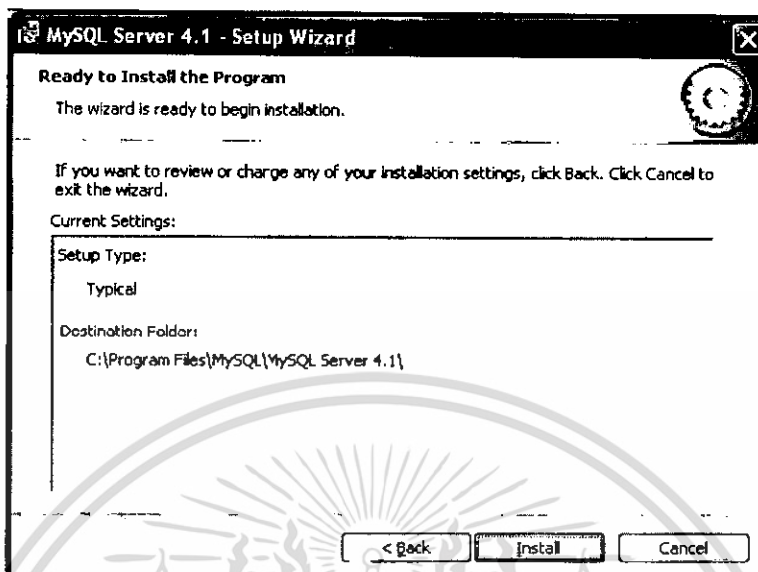
##### 1.2 ปรากฏหน้าจอ Setup Type เลือก Typical จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.2 แสดง Setup Type

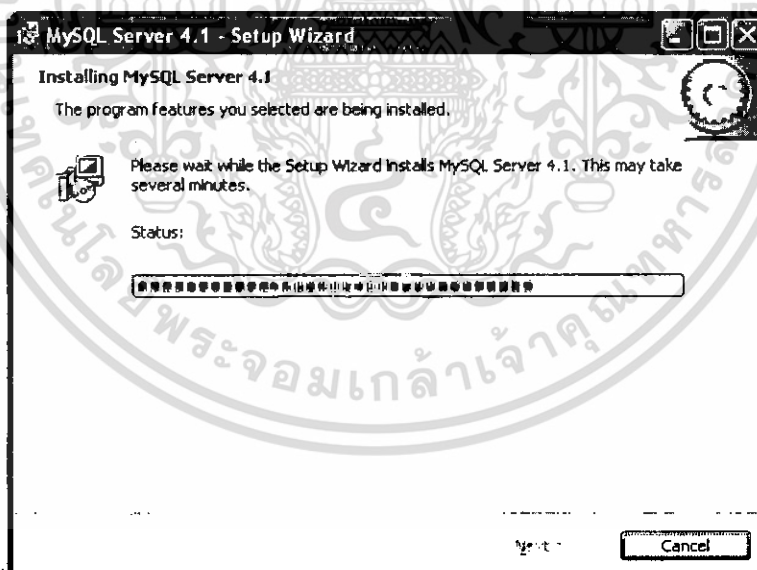
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3 ปรากฏหน้าจอ Ready to Install the Program จากนั้นคลิกปุ่ม Install



รูปที่ ก.1.3 แสดง Ready to Install the Program

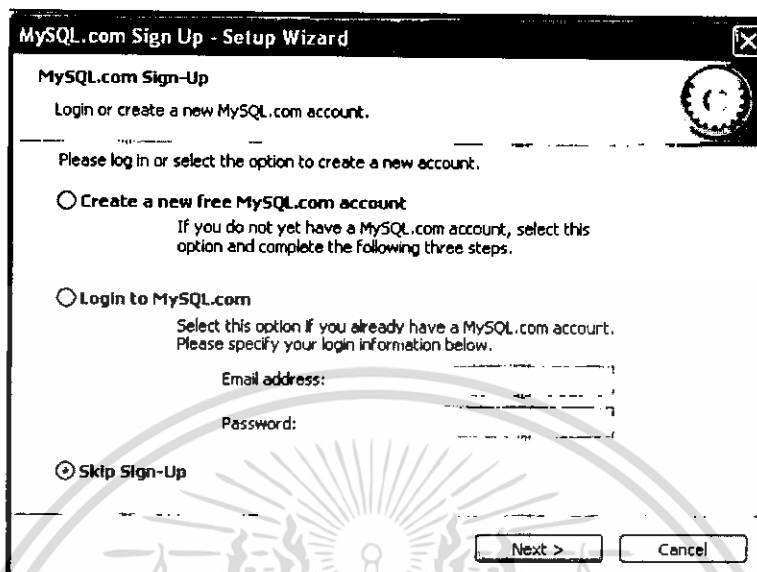
### 1.4 ปรากฏหน้าจอ Installing MySQL SQL Server



รูปที่ ก.1.4 แสดง Installing MySQL Server 4.1

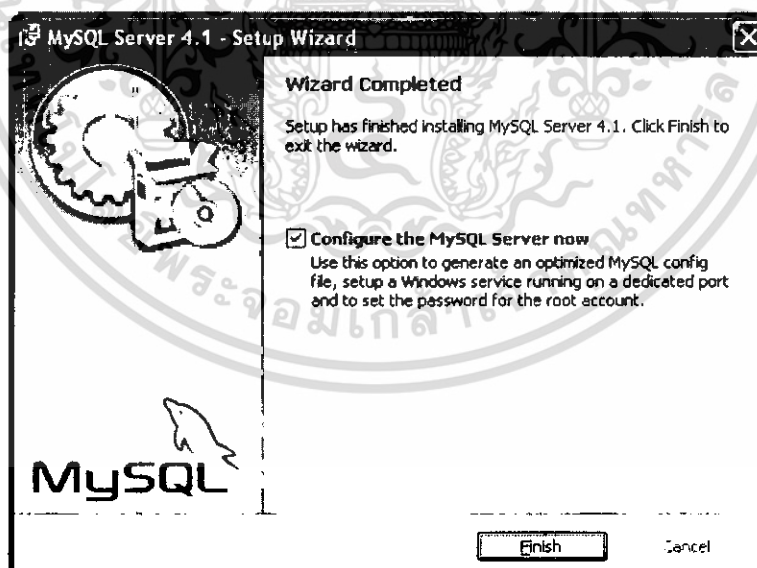
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ปรากฏหน้าจอ MySQL.com Sign-Up เลือก Skip Sign-Up จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.5 แสดง MySQL.com Sign-up

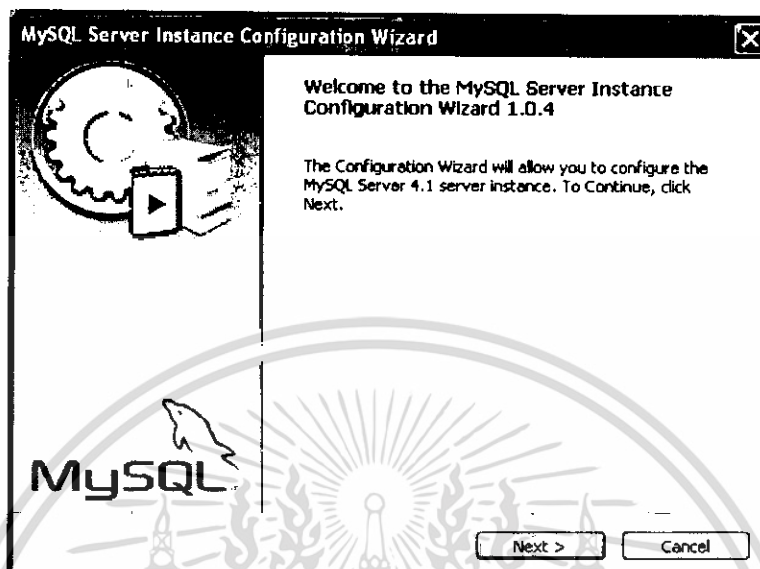
1.6 ปรากฏหน้าจอ Wizard Completed เลือกทำการ Configure the MySQL Server now จากนั้นคลิกปุ่ม Finish



รูปที่ ก.1.6 แสดง Wizard Completed

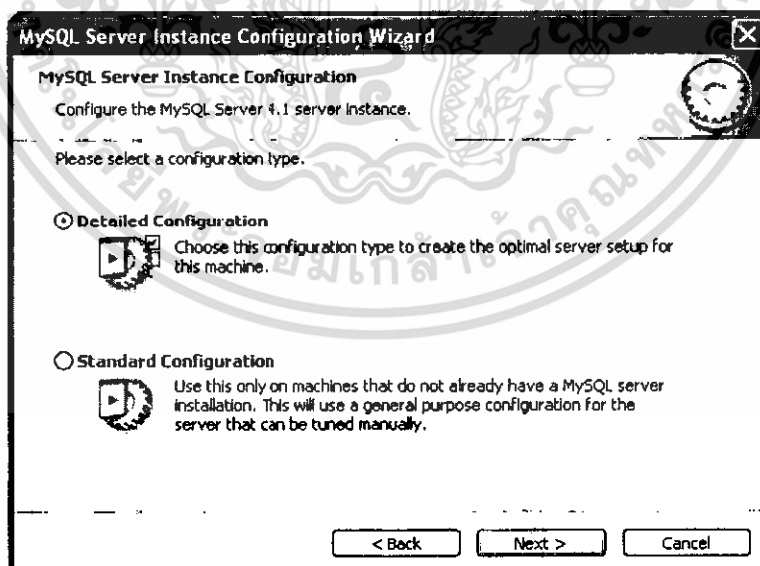
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 ปกรากฎหน้าจอ Welcome to the MySQL Server Instance Configuration Wizard 1.0.4  
จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.7 แสดง Welcome to the....

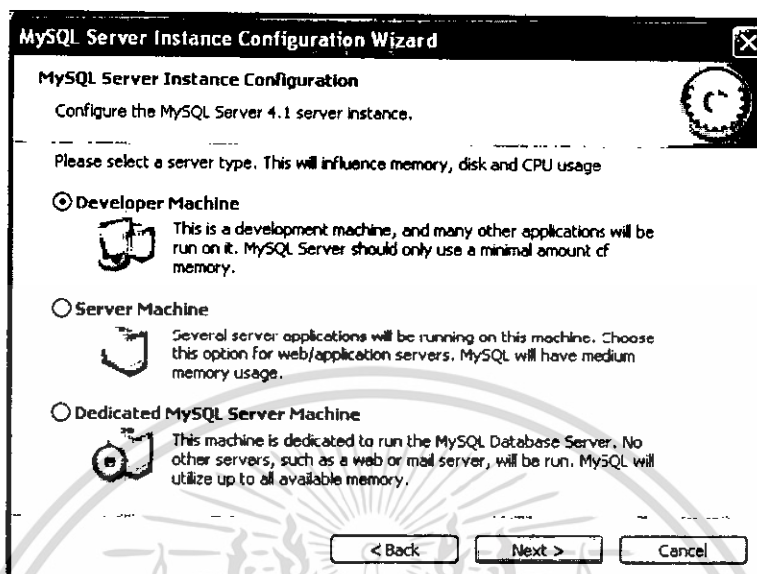
1.8 ปกรากฎหน้าจอ MySQL Server Instance Configuration เลือก Detailed Configuration  
จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.8 แสดง MySQL Server Instance Configuration 1

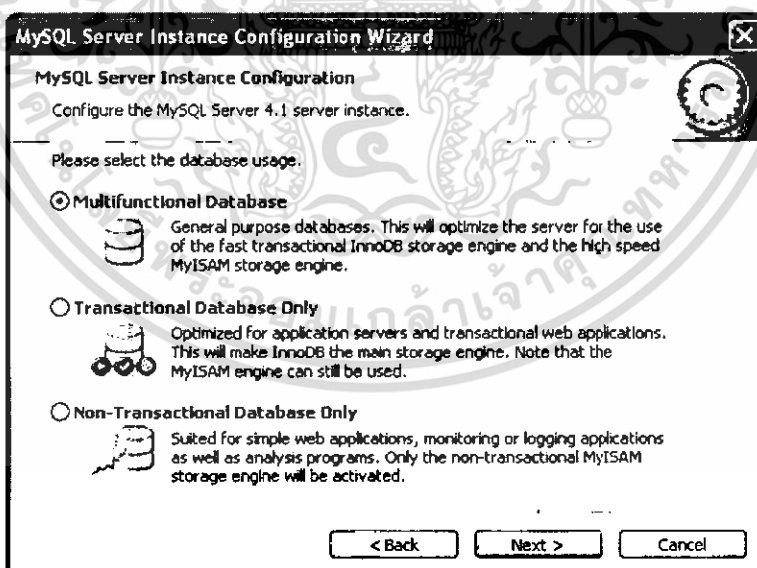
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.9 หน้า Please select a server type เลือก Developer Machine จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.9 แสดง MySQL Server Instance Configuration 2

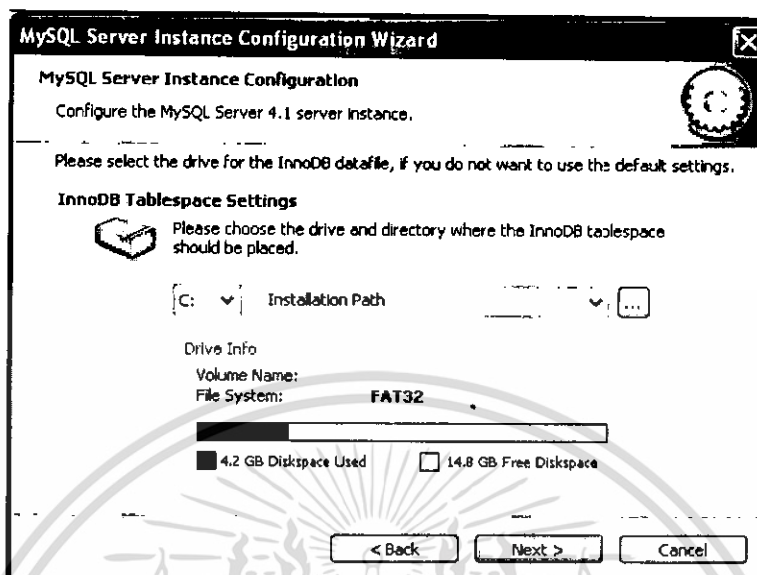
1.10 หน้า Please select the database usage. เลือก Multifunctional Database จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.10 แสดง MySQL Server Instance Configuration 3

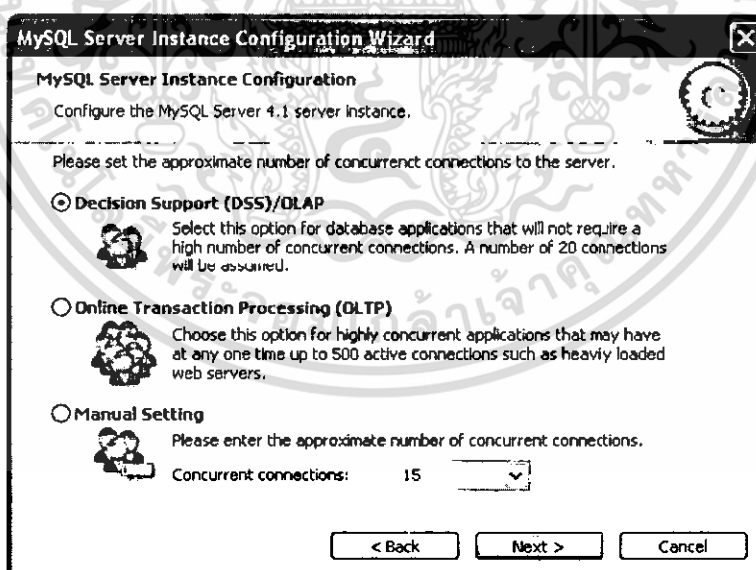
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.11 หน้า Please select the drive for the InnoDB datafile จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.11 แสดง MySQL Server Instance Configuration 4

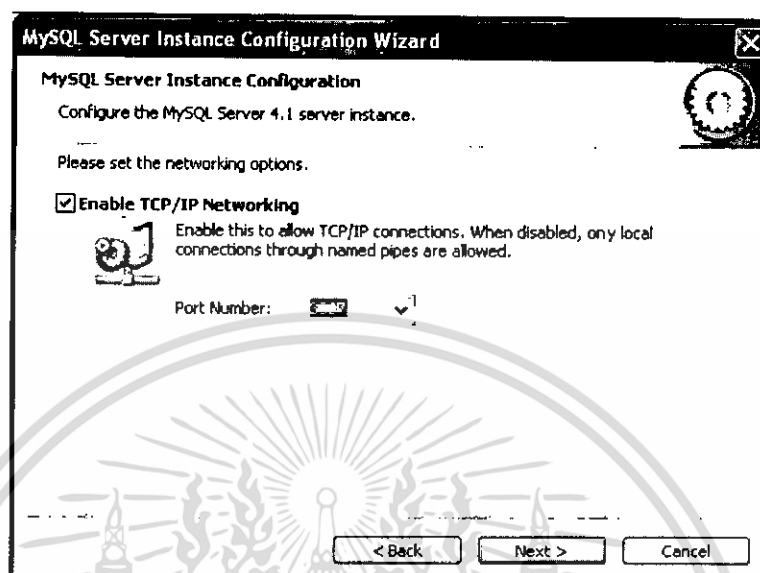
1.12 หน้า Please set the approximate number of concurrent connections to the server.  
เลือก Decision Support (DSS)/OLAP จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.12 แสดง MySQL Server Instance Configuration 5

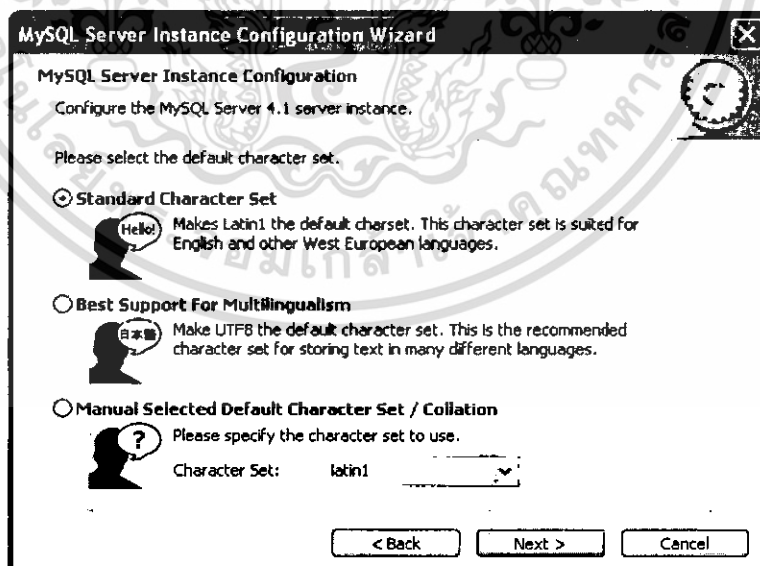
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.13 หน้า Please set the networking options. ทำการเลือก Enable TCP/IP Networking ให้ Port Number เป็น 3307 จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.13 แสดง MySQL Server Instance Configuration 6

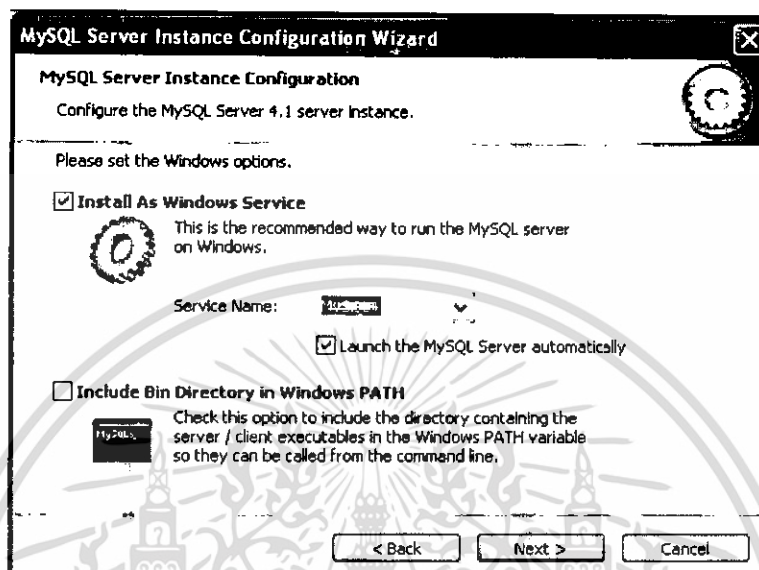
1.14 หน้า Please select the default character set. เลือก Standard Character Set จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.14 แสดง MySQL Server Instance Configuration 7

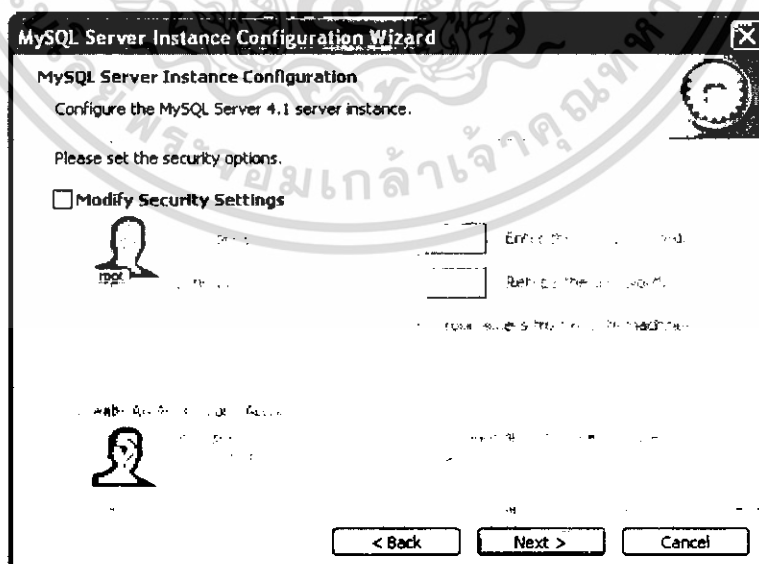
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.15 หน้า Please set the Windows options. เลือกทำการ Install As Windows Service ตั้งค่า Service Name เป็น MySQL4 เลือกทำ Launch the MySQL Server automatically จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.15 แสดง MySQL Server Instance Configuration 8

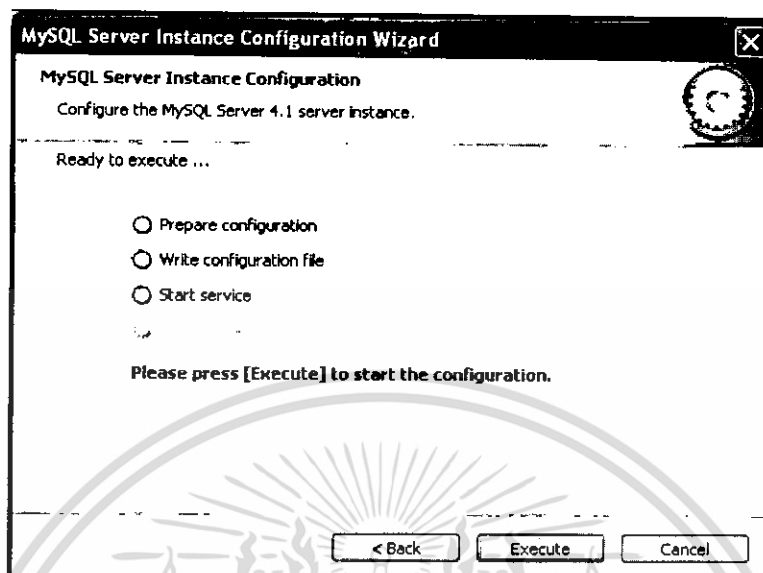
1.16 หน้า Please set the security options. ทำการ Disable ที่ Modify Security Settings จากนั้นคลิกปุ่ม Next



รูปที่ ก.1.16 แสดง MySQL Server Instance Configuration 9

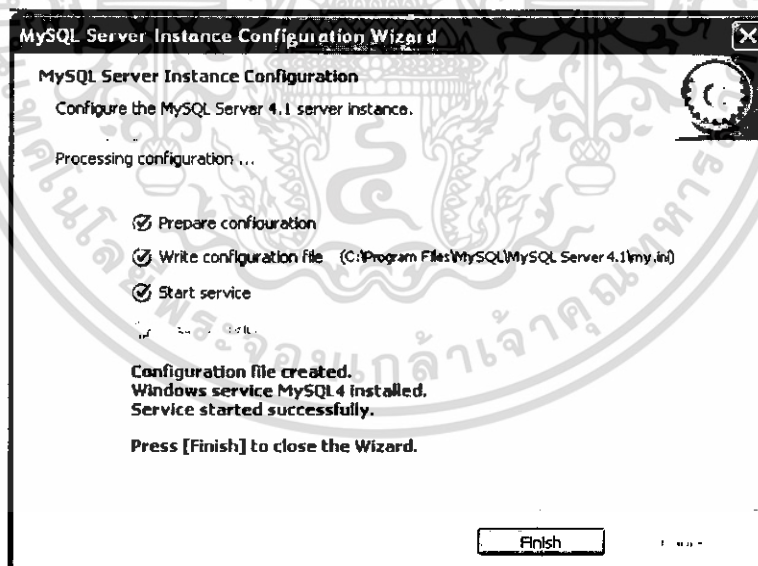
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.17 หน้า Ready to execute คลิกปุ่ม Execute



รูปที่ ก.1.17 แสดง MySQL Server Instance Configuration 10

## 1.18 หน้า Processing Configuration คลิกปุ่ม Finish

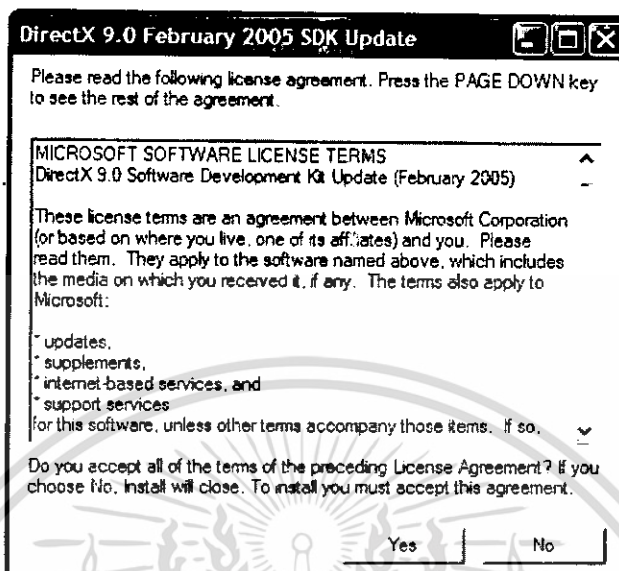


รูปที่ ก.1.18 แสดง MySQL Server Instance Configuration 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. การติดตั้ง DirectX SDK 9.0

2.1 ทำการดับเบิลคลิกที่ไฟล์ dxsdk.exe จะมีหน้าต่างปรากฏขึ้นมาดังนี้ ตอบ yes



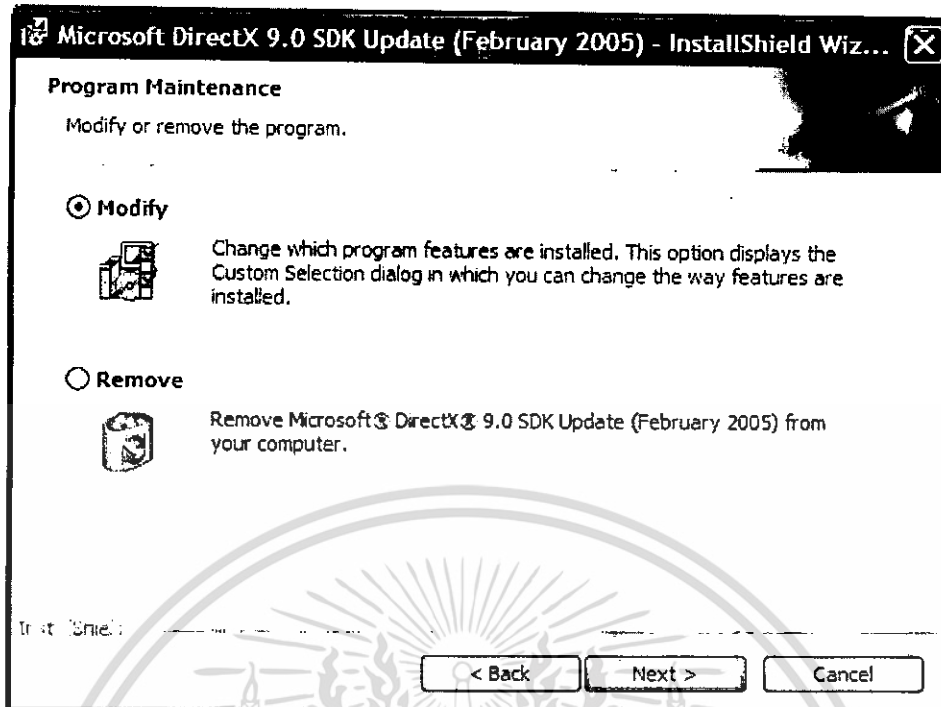
รูปที่ ก.2.1 หน้าต่างเงื่อนไขในการติดตั้งDirectx

2.2 หลังจากนั้นจะมีหน้าต่างสำหรับการติดตั้งปรากฏขึ้น กด Next



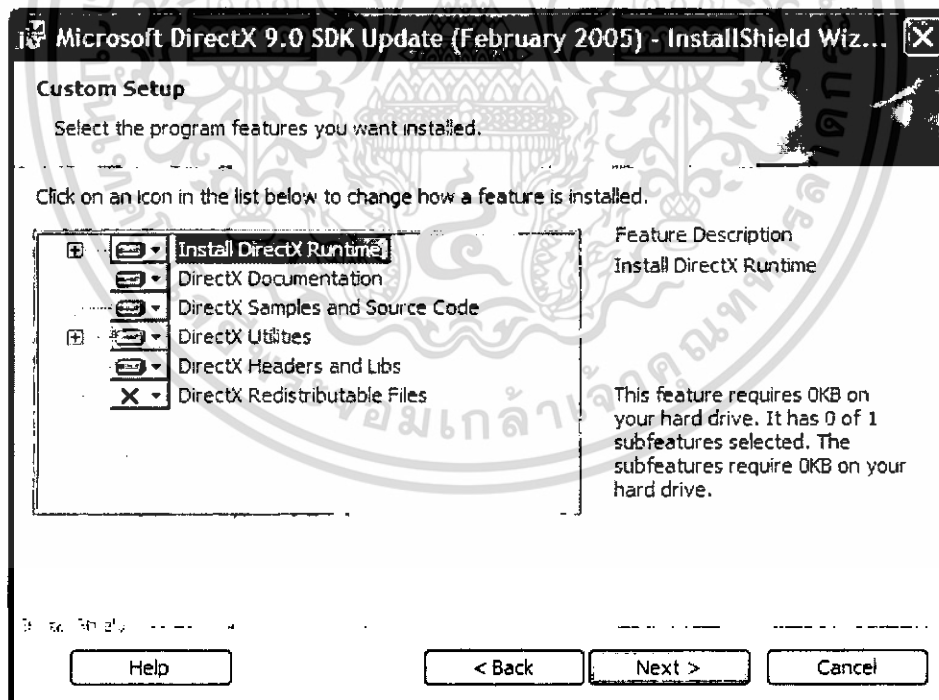
รูปที่ ก.2.2 หน้าต่างต้องรับก่อนการติดตั้งDirectx

2.3 ต่อมาจะเป็นหน้าต่างตัวเลือกให้เลือก Install หรือ Modify ในที่นี้เป็น Modify



รูปที่ ก.2.3 หน้าต่างให้เลือก Function ที่ต้องการทำกับDirectx

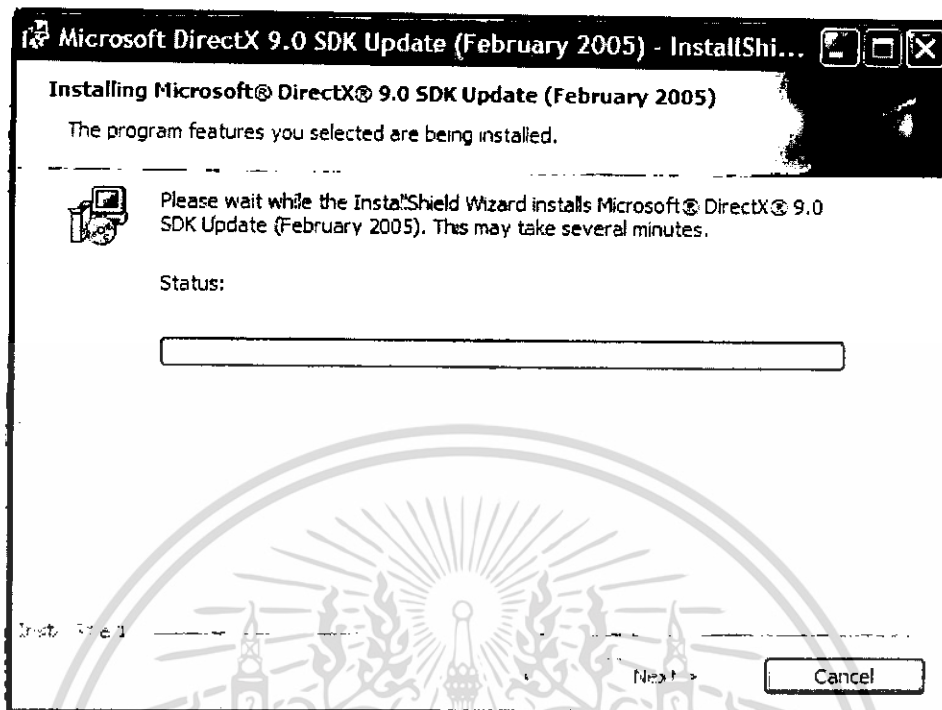
2.4 ทำการเลือกที่จะติดตั้งตัวใดบ้างให้เลือกดังต่อไปนี้แล้วตอบ Next



รูปที่ ก.2.4 หน้าต่างเลือก Feature ที่ต้องการติดตั้งของDirectx

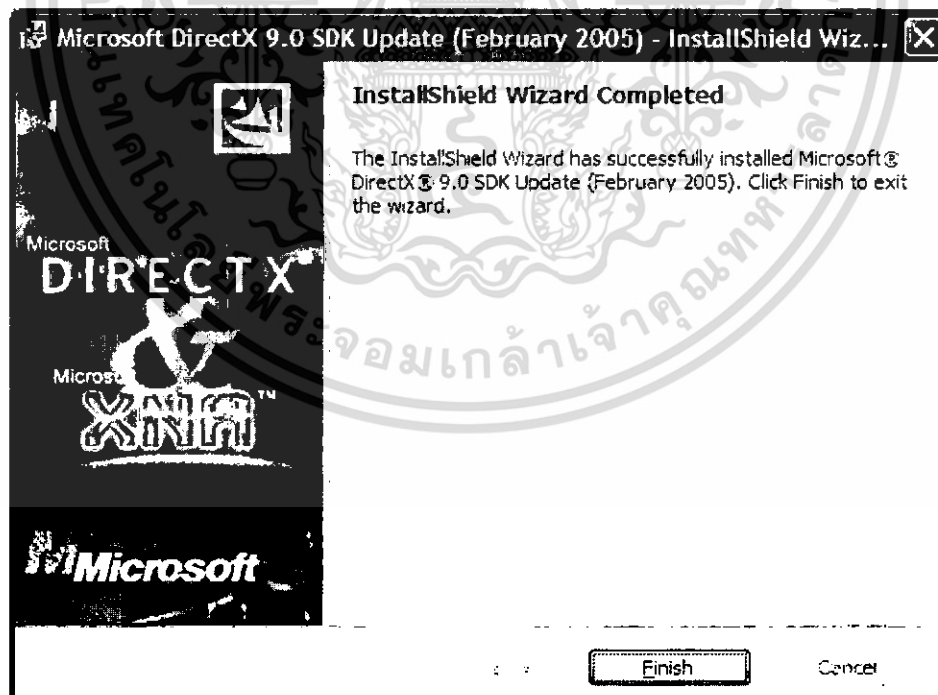
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 หลังจากนี้โปรแกรมจะทำการติดตั้งตัวเองลงเครื่อง



รูปที่ ก.2.5 หน้าต่างแสดงการติดตั้งDirectx

2.6 เสร็จสิ้นการติดตั้ง



รูปที่ ก.2.6 หน้าต่างเงื่อนไขในการติดตั้งDirectx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 หลังจากนั้นให้เข้าไปที่ Folder ที่ทำการติดตั้ง direct X แล้วเข้าไปยัง Folder ดังนี้ Samples\C++\DirectShow\BaseClasses เข้าสู่ Project แล้ว compiler แล้วตรวจสอบใน Folder Debug จะมีไฟล์ชื่อ Strmbasd.lib นำไฟล์นี้ไปใส่ใน Folder Lib ของ Direct X

2.8 การติดตั้งเสร็จสิ้นสามารถใช้งานได้

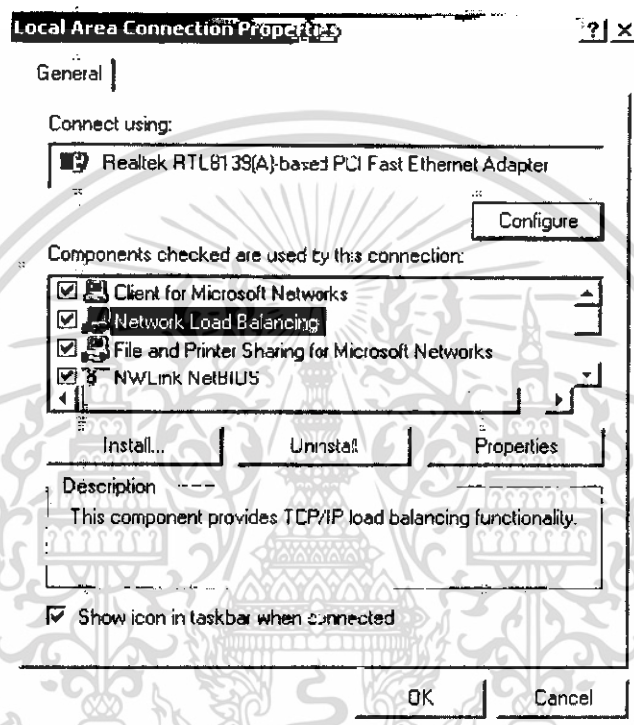


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข.

## การตั้งค่า Network Load Balancing (Window 2003 Server)

1. ที่หน้าต่าง Local Area Connection Properties ให้ทำการ เลือกที่ Network Load Balancing > Properties



รูปที่ ข.1 แสดง Local Area Connection Properties

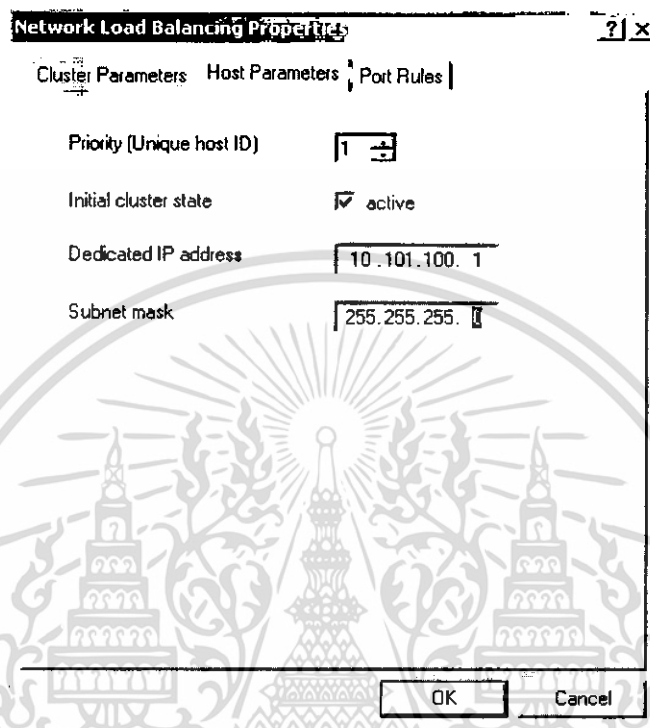
2. เมื่อทำการคลิกที่ Network Load Balancing > Properties จะเข้าสู่หน้าจอการทำงาน ให้เลือกที่ Cluster Parameters แล้วตั้งค่า Primary IP Address: 10.101.100.10 Subnet mask: 255.255.255.0 ซึ่งเป็น Subnet mask ใน class C (Primary IP Address คือ ไอพีกลาง ซึ่งเป็นไอพีที่รวมกลุ่มเซิร์ฟเวอร์ให้เป็นกลุ่มเดียวกัน)

The screenshot shows the 'Network Load Balancing Properties' dialog box with the 'Cluster Parameters' tab selected. The fields are as follows:

Field	Value
Primary IP address	10.101.100.10
Subnet mask	255.255.255.0
Full Internet name	cluster.domain.com
Network address	10.101.100.0
Multicast support	<input checked="" type="checkbox"/> enabled
Remote password	
Confirm password	
Remote control	<input checked="" type="checkbox"/> enabled
Help	Help
Buttons	OK, Cancel

รูปที่ ข.2 แสดง Network Load Balancing Properties

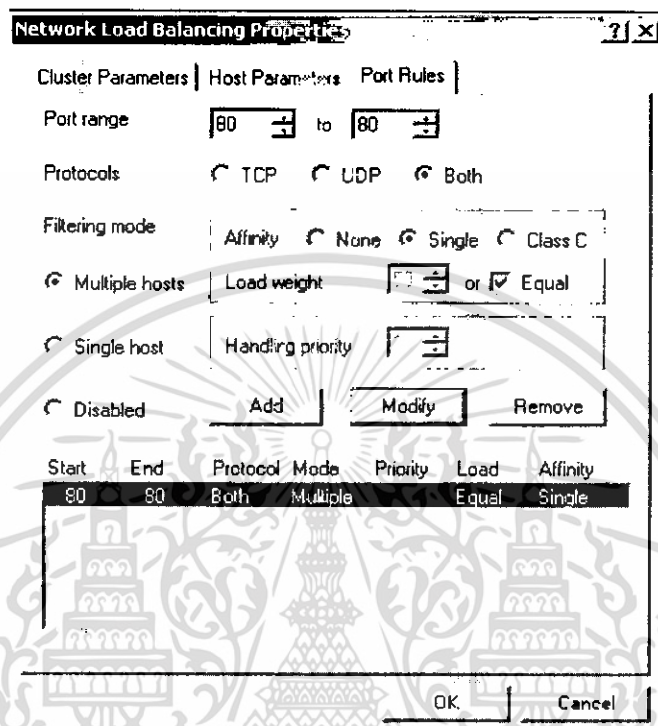
3. ทำการ Host Parameters ตั้งค่า Priority (Unique host ID) ให้เป็น 1 ตั้งค่า Dedicated IP address (ไอพีของเครื่องที่มีอยู่เดิม) ในที่นี้คือ 10.101.100.1 ค่า Subnet mask เป็น 255.255.255.0 (ค่า Unique host ID ของเครื่อง Server ในกลุ่มต้องไม่เหมือนกัน)



รูปที่ ข.3 แสดง Network Load Balancing Properties 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการเลือก Port Rules จากนั้นตั้งค่าของ Port range ให้เป็น พอร์ต 80 to 80 (เป็นพอร์ตที่ใช้สำหรับติดต่อ หากไม่แน่ใจตั้งเป็น 0 to 65532 ได้) เลือก Protocols เป็น Both คลิกเลือกที่ Filtering mode เป็น Single เมื่อเสร็จสิ้นการตั้งค่าต่างๆ แล้วกด OK



รูปที่ ข.4 แสดง NetWork Load Balancing Properties 3