

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบจำลองสงครามสัตว์ประหลาด
(Monster War Simulation System)



โดย

นางสาวอรรวิทย์ รัตนชน 44010595

นายอุดม อศิรพจน์ 44010619

นายเอกสิทธิ์ เกษมสานต์ 44010625

อาจารย์ที่ปรึกษา

อ.เกียรติคุณ เจียรนัยระนงกิจ

อ.เอนก กอธนสาร

เลขหมู่.....
เลขทะเบียน..... 61525
วัน,เดือน,ปี..... 18 ก.ค. 2549

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2547

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบจำลองสงครามสัตว์ประหลาด

Monster War Simulation System

คณะผู้จัดทำ นางสาวอรวรีย์ รัตนชน

นายอุดม อสิรพจน์

นาย เอกสิทธิ์ เกษมसानต์



.....อาจารย์ที่ปรึกษา

(อ.เกษรจิตต์ เจียรนัยชนะกิจ)

.....อาจารย์ที่ปรึกษา

(อ.เอนก กอชนสาร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจำลองสงครามสัตว์ประหลาด

นางสาวอรรวิชัย รัตนชน 44010595

นายอุดม อศิรพจน์ 44010619

นายเอกสิทธิ์ เกษมสานต์ 44010625

อ.เกียรติคุณ เจียรนัยชนะกิจ อาจารย์ที่ปรึกษา

อ.เอนก กอธนสาร อาจารย์ที่ปรึกษา

ปีการศึกษา 2547

บทคัดย่อ

ปัจจุบัน เกมออนไลน์กำลังได้รับความนิยมมากขึ้นเรื่อยๆ ในประเทศไทย ดังนั้นแนวทางในการพัฒนาเกมออนไลน์ ซึ่งต้องผสมผสานความรู้ด้านต่าง ๆ ของคอมพิวเตอร์ ไม่ว่าจะเป็น การเขียนโปรแกรม, การใช้อินเทอร์เน็ต, ระบบเน็ตเวิร์ค ไปจนถึงการเขียนโปรแกรมบนเว็บและระบบฐานข้อมูล จึงเป็นเรื่องที่น่าสนใจ เป็นอย่างยิ่งเพื่อเป็นแนวทางสำหรับการพัฒนาในอนาคต

ปัญญานิพนธ์ฉบับนี้เป็นอีกแนวทางหนึ่งในการพัฒนาเกมออนไลน์ ด้วยการนำเอาเทคโนโลยีของแอปพลิเคชันบนโทรศัพท์มือถือเข้ามาร่วมกับโปรแกรมบนเครื่องคอมพิวเตอร์เพื่อเป็นอีกหนึ่งทางเลือกใหม่ของเกมออนไลน์ อีกทั้งยังช่วยเพิ่มความสะดวกสบายให้กับผู้เล่นอีกด้วย

โดยเกมนี้ผู้เล่นจะได้เลี้ยงสัตว์ประหลาดของตัวเองและสามารถนำไปประลองกัน ในเครือข่ายได้ นอกจากนี้ผู้เล่นยังสามารถนำสัตว์ประหลาดของไปตัวเองไปเลี้ยงบนมือถือได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Monster War Simulation System

Onwaree Rattanathon 44010595

Udom Asiraphot 44010619

Ekasit kasamesan 44010625

A.Kiatkul Jiaranaitanakit Advisor

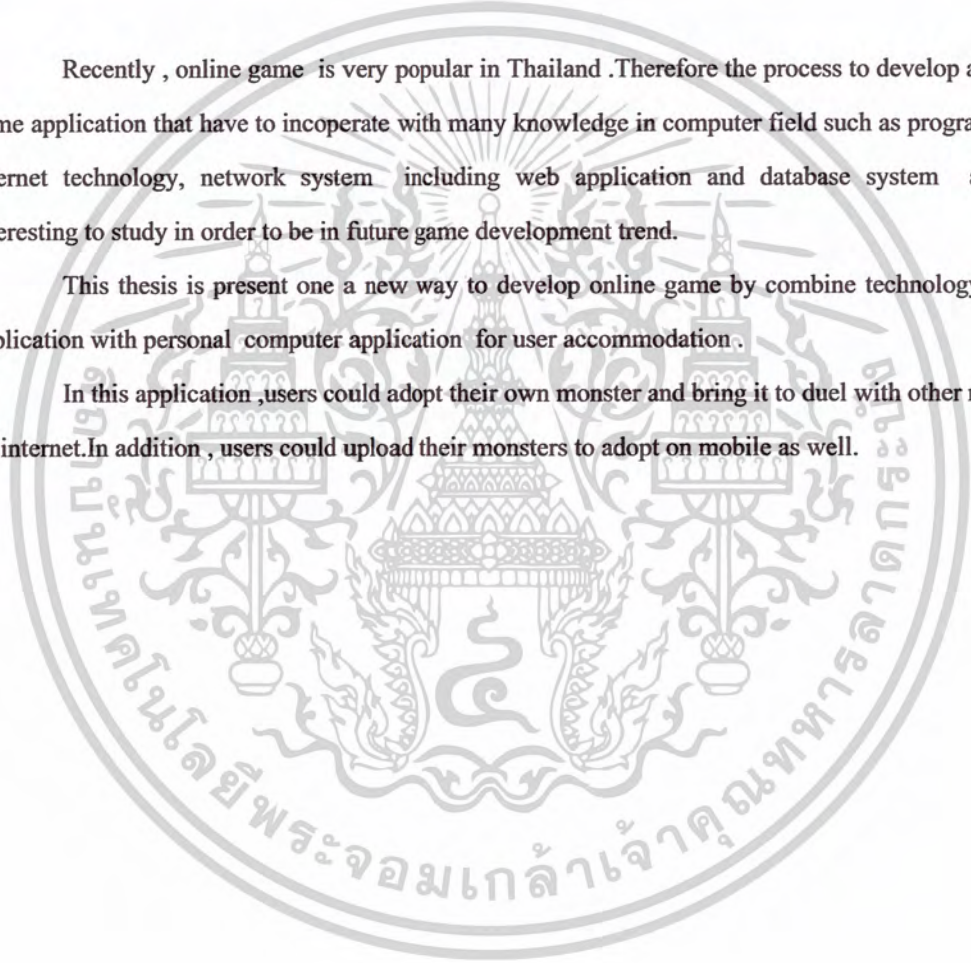
A.Anake Ghortanasan Advisor

ABSTRACT

Recently , online game is very popular in Thailand .Therefore the process to develop an online game application that have to incorporate with many knowledge in computer field such as programming , internet technology, network system including web application and database system are very interesting to study in order to be in future game development trend.

This thesis is present one a new way to develop online game by combine technology mobile application with personal computer application for user accommodation .

In this application ,users could adopt their own monster and bring it to duel with other monster on internet.In addition , users could upload their monsters to adopt on mobile as well.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	1
1.4 วิธีในการดำเนินงาน	1
บทที่ 2 ทฤษฎี	2
2.1 เทคโนโลยีที่ใช้ทางด้านคอมพิวเตอร์	2
2.1.1 DirectX	2
2.1.2 TCP/IP	5
2.1.3 WINSOCK	8
2.2 เทคโนโลยีที่ใช้สำหรับแอปพลิเคชันบนมือถือ	16
2.2.1 Java Server Pages (JSP)	16
2.2.2 J2ME	21
2.2.3 หลักการพัฒนาแอปพลิเคชันบน MIDP ด้วย J2ME	24
2.2.4 MIDlets	26
2.2.5 การเชื่อมต่อ J2ME กับเครือข่าย	28
2.3 เทคโนโลยีอื่นๆ	32
2.3.1 MySQL	32
2.3.2 JDBC	37
บทที่ 3 การดำเนินงานและการออกแบบระบบ	43
3.1 การสำรวจและวิเคราะห์ระบบเกมออนไลน์	43
3.1.1 Gunbound	43
3.1.2 Ragnarok Online	46
3.1.3 Survival project	49
3.2 โครงสร้างและการทำงานของระบบ	52
3.2.1 การออกแบบระบบฐานข้อมูล	51
3.2.2 การออกแบบระบบ	57
3.2.2.1 Use Case Diagram แสดงความสัมพันธ์ของผู้ใช้กับระบบ	57
3.2.2.2 Activity Diagram	58
3.2.2.3 class diagram	61
บทที่ 4 ผลการทดลอง	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ลักษณะตัวเกม	65
4.1.1 ต่อสู้ออนไลน์	65
4.1.2 เลี้ยงมอนเตอร์บนมือถือ	66
บทที่ 5 บทวิจารณ์และสรุป	72
5.1 บทวิจารณ์และสรุป	72
5.2 ปัญหาที่เกิดขึ้นในการพัฒนาและแนวทางในการแก้ปัญหา	72
5.3 ข้อจำกัดของระบบ	72
5.4 แนวทางการพัฒนา	72
บรรณานุกรม	73
ภาคผนวก	74



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ปัจจุบันตลาดเกมออนไลน์ในเมืองไทยค่อนข้างได้รับความนิยมมากขึ้นอย่างเห็นได้ชัด ดังนั้น เราจึงต้องมาพูดถึงเรื่องแนวทางในการพัฒนาเกมออนไลน์กัน ซึ่งต้องผสมผสานความรู้ด้านต่าง ๆ ของคอมพิวเตอร์ ไม่ว่าจะเป็น การเขียนโปรแกรม, การใช้อินเทอร์เน็ต, ระบบเน็ตเวิร์ค ไปจนถึงการเขียนโปรแกรมบนเว็บและระบบฐานข้อมูล ดังนั้น นอกจากคนไทยเราจะเป็นคอเกมออนไลน์แล้ว การสร้างเกมออนไลน์จึงเป็นโปรเจกต์หนึ่งที่น่าท้าทายนักเขียนโปรแกรมเกมของไทยด้วยเหมือนกัน

1.2 วัตถุประสงค์ของงานวิจัย

เพื่อฝึกฝนทักษะกระบวนการคิดและการนำเทคโนโลยีมาประยุกต์ใช้ในการสร้างเกม โดยสร้างเป็นเกมออนไลน์ ซึ่งเป็นเกมที่ใช้เทคโนโลยีให้กว้างขวางกว่าเกมแนวอื่น รวมถึงเพื่อศึกษาแนวทางในการพัฒนา application , การสื่อสารแบบไร้สายบน โทรศัพท์มือถือ

1.3 ขอบเขตของงานวิจัย

ขอบเขตของระบบจำลองสงครามสัตว์ประหลาด แบ่งได้เป็น 4 ส่วน

1. ส่วนของการพัฒนา application client ของตัวเกมบนคอมพิวเตอร์ ในระบบ 2D Graphic
2. ส่วนของการพัฒนา application client ของตัวเกมบนมือถือ ซึ่งอาศัยการ download / upload ข้อมูลผ่าน web server และเล่นด้วยระบบ offline
3. ส่วนของการพัฒนา ระบบ network แบบ client – server และ ระบบ database ที่ใช้ในการเล่นเกมแบบ online
4. ส่วนของ web server ประกอบด้วย web community บนเครื่อง PC เพื่อใช้ในการให้บริการลงทะเบียน ข่าวสารสำหรับสมาชิกและ web server เพื่อช่วยในการ ให้บริการ ในการติดต่อกับฐานข้อมูลสำหรับ โทรศัพท์มือถือ

โดยทั้ง4ส่วนนั้นจะเกี่ยวเนื่องกัน ในทางการส่งข้อมูลและการเล่นเกม

1.4 วิธีในการดำเนินงาน

งานวิจัยนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่างๆที่เกี่ยวข้องกับงานวิจัย โดยศึกษา tool ต่างๆ และนำมาประยุกต์ใช้กับงานวิจัย

บทที่ 2 ทฤษฎี

2.1 เทคโนโลยีที่ใช้ทางด้านคอมพิวเตอร์

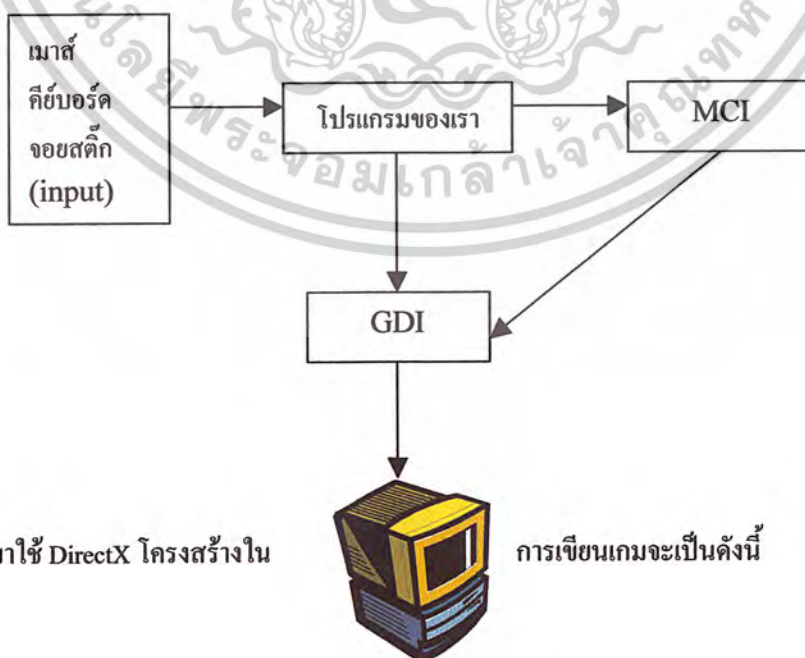
2.1.1 DirectX

DirectX พัฒนาขึ้นมาภายใต้เทคโนโลยีที่เรียกว่า COM (Component Object Model) ของไมโครซอฟท์ และในตัว DirectX นั้นก็แบ่งออกเป็นอีกหลายชุดคำสั่ง หรือคลาสย่อยๆ อีกหลายคลาส เพื่อใช้ในการทำงานต่างๆกัน

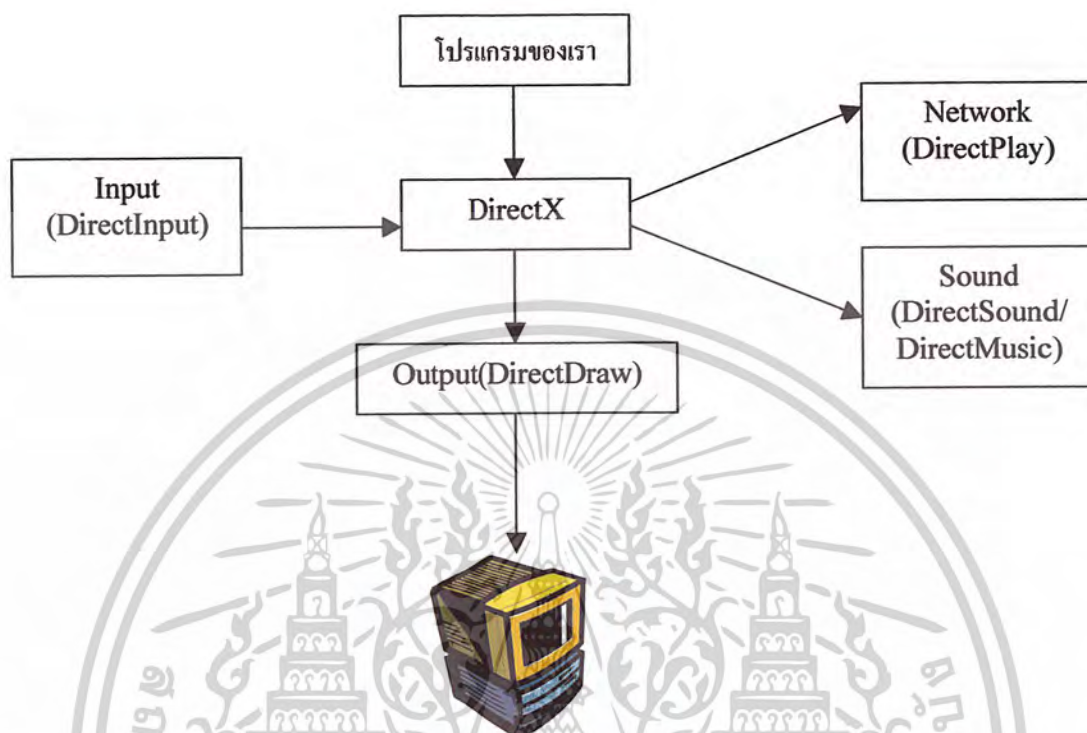
DirectX ได้รวมเอาส่วนประกอบที่ทำงานคล้ายกันเข้าไว้ด้วยกันดังนี้

- DirectX Graphics ประกอบด้วย DirectDraw และ Direct3D ใช้ในการสร้างภาพ
- DirectX Audio ประกอบด้วย DirectSound และ DirectMusic
- DirectInput ใช้ในการรับค่าจากเมาส์ คีย์บอร์ด และจอยสติ๊ก
- DirectPlay ใช้ในการเล่นแบบหลายๆ คนผ่านระบบเครือข่าย
- DirectShow ใช้ในการแสดงผลด้านมัลติมีเดีย เช่น Mpeg หรือ AVI เป็นต้น
- DirectSetup ใช้ในการติดตั้งโปรแกรม

วิธีการเขียนเกมแบบต่างๆ ไปถ้าไม่ใช่ DirectX เราจะใช้ชุดคำสั่ง GDI (Graphics Device Interface) และชุดคำสั่งมัลติมีเดีย (MCI) จาก Win32 ในการเล่นไฟล์เสียง .wav หรือการแสดงผลภาพออกทางหน้าต่าง ดังรูปต่อไปนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จากรูปข้างต้น DirectX จะเป็นตัวกลางผ่านไปสู่ระบบฮาร์ดแวร์ แทนที่เราจะเรียกใช้ชุดคำสั่งด้านฮาร์ดแวร์โดยตรง ที่เป็นเช่นนี้ก็เพราะว่า ชุดคำสั่งใน DirectX จะอำนวยความสะดวกให้ในการแสดงผล และการรับข้อมูล ซึ่งเหมาะที่จะใช้ในเกมมากกว่านั่นเอง ไม่จำเป็นต้องดูแลการแสดงรูป และการเล่นแอนิเมชันด้วยตัวเอง ซึ่งถ้าท่านเขียนเกมที่มีตัวละครเดินไปเดินมา เราจะต้องนำมันมาเรียงกันในตำแหน่ง x และ y ที่ต้องการ และสลับเฟรมไปเรื่อยๆ ที่เราเรียกกันว่า สไปรท์ (Sprite) นั่นเอง แต่ DirectX ได้เตรียมชุดคำสั่ง DirectDraw เอาไว้จัดการกับการแสดงผล และรูปภาพ ซึ่งจะกระทำอยู่บนพื้นผิว (Surface) นำรูปที่ต้องการมาวางไว้บน Surface จากนั้นก็สลับ Surface นี้เท่านั้น ก็จะได้เป็นภาพเคลื่อนไหวออกมา

ถ้าจะยกตัวอย่างการจัดการด้านเสียงเอฟเฟกต์และดนตรี เช่น การสั่งให้เล่นเสียง .WAV ถ้าเราใช้ชุดคำสั่ง MCI ธรรมดา เราอาจจะไม่สามารถควบคุมความถี่ของเสียง ความดังของเสียงได้โดยตรง แต่ถ้าใช้ DirectSound เราจะสามารถควบคุมได้ง่ายกว่า เช่น ให้ออกลำโพงไหน หรือว่าใช้ระดับเสียงเท่าใด เป็นต้น

2.1.1.1 Surface

ส่วนประกอบแรกที่เราจะต้องมาทำความรู้จัก นั่นก็คือ DirectDraw ในการแสดงภาพต่างๆ เพราะว่าการสร้างโปรแกรม สิ่งแรกที่จะต้องแสดงบนจอเมื่อถือก็คือ ภาพต่างๆ เช่น ภาพฉากหลัง, ภาพตัวละคร, ภาพอาวุธ หรือข้อความแสดงข้อมูลภายในเกมต่างๆ

การแสดงผลของ DirectDraw เราจะใช้สิ่งที่เรียกว่า “พื้นผิว” หรือ Surface ในการแสดงผล เราจะวาดลงไปตรงๆที่จอภาพเหมือนกับการเขียนโปรแกรมกราฟฟิกเลยไม่ได้ เพราะว่า DirectDraw ได้เตรียม Surface ไว้เป็นที่รองรับในการแสดงผลอยู่แล้ว Surface เปรียบเสมือนกับกระดาษที่จะใช้ในการวาด ถ้าเราเขียนเกมโดยใช้ DirectX แต่ไม่มีการฉายในการวาดภาพ ก็คงจะไม่ใช้เกม ดังนั้นในหนึ่งโปรแกรม จะต้องมียกน้อย 1 พื้นผิว เราเรียกพื้นผิวแบบนี้ว่า “Primary Surface” หรือ “พื้นผิวหลัก”

จากจาก Surface จะช่วยในการแสดงผลภาพฉากหรือตัวละครในเกมแล้ว เราก็จะใช้ Surface ในการแสดงผลภาพเคลื่อนไหวของเกมอีกด้วย โดยสิ่งที่ได้กล่าวไปแล้วว่า โปรแกรมหนึ่ง โปรแกรม อย่างน้อยจะต้องมี 1 พื้นผิวหลัก หรือ Primary Surface ซึ่งในความเป็นจริง ถ้าเรามีเพียงแค่ Primary Surface การแสดงผลภาพเคลื่อนไหวอาจจะไม่สมบูรณ์ เราจะต้องอาศัยพื้นผิวที่เรียกว่า “Back Surface” หรือ “พื้นผิวรอง” ซึ่งจะใช้เก็บภาพเฟรมที่เคลื่อนไหวต่อไป และภาพเคลื่อนไหวจะเกิดขึ้นได้ก็ต่อเมื่อมีการสลับพื้นผิวระหว่าง Black Surface และ Primary Surface นี้

ถ้าเรามีรูปภาพอยู่ 2 รูป และต้องการให้เรียงกันเป็นภาพเคลื่อนไหว เราก็ให้นำเอาภาพแรกใส่ไว้ใน Primary Surface และนำภาพที่สองมาใส่ไว้ใน Back Surface จากนั้นเราจะทำการ “พลิก” (Flip) ซึ่งการพลิกนี้ จะเป็นการทำให้ภาพที่อยู่ใน Black Surface มาปรากฏอยู่ใน Primary Surface ส่วนภาพเดิมที่อยู่ใน Primary Surface นั้นจะไปอยู่ใน Black Surface แทน เมื่อเราสั่งพลิกอีกครั้งหนึ่ง ภาพทั้งสองก็จะกลับสู่ที่เดิมของมัน และถ้าเราสั่งพลิกอีก มันก็จะสลับกันอีกไปเรื่อยๆ

ในการแสดงผลภาพเคลื่อนไหว ไม่จำเป็นว่าจะต้องใช้ทั้ง Primary Surface และ Black Surface เสมอไป เราใช้แค่ Primary Surface เพียง Surface เดียวก็ทำได้ ยกตัวอย่างเช่น ถ้าเรามีภาพที่จะทำการเคลื่อนไหวมากกว่า 2 ภาพ เราสามารถสร้าง Surface แบบชั่วคราว เพื่อใช้เก็บแต่ละภาพๆ ได้ โดยไม่จำเป็นต้องใช้ Back Surface และ ไม่จำเป็นจะต้องพลิกก็ได้

แต่ในการสร้างภาพเคลื่อนไหวในเกมจริงๆนั้น ตัวละครไม่ได้มีแค่หนึ่งตัว ดังนั้นเราจึงต้องใช้ Back Surface เข้ามาช่วย ภาพที่ได้ก็จะมีควมสมบูรณ์มากขึ้น โดยวิธีก็คือ เราจะต้องวาดภาพแต่ละภาพลงใน Back Surface ให้เรียบร้อย จากนั้นจึงทำการพลิกเพื่อนำภาพใน Back ขึ้นไปสู่ Primary Surface จากนั้นก็วาดภาพใหม่ (ในเฟรมต่อไป) ลงใน Back Surface และก็ทำการพลิกอีก เพื่อให้เป็นภาพเคลื่อนไหวนั่นเอง

2.1.2 TCP/IP

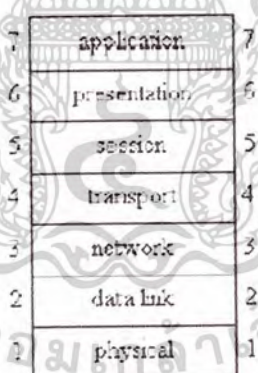
TCP/IP เป็นโพรโทคอลที่ใช้กันแพร่หลายที่สุด โดยเฉพาะเมื่อถูกนำไปใช้กับเครือข่ายอินเทอร์เน็ต ปัจจุบัน TCP/IP ได้ถูกใช้งานอย่างแพร่หลายออกไปในเครือข่ายทุกระดับ ไม่เฉพาะแต่ในอินเทอร์เน็ตอีกต่อไป

OSI Model มาตรฐานอ้างอิงในการสื่อสารข้อมูล

เมื่อคอมพิวเตอร์เครื่องหนึ่งมีความต้องการรับส่งข้อมูลกับคอมพิวเตอร์อีกเครื่องอื่น ๆ การเชื่อมต่อระหว่างคอมพิวเตอร์เข้าด้วยกันเป็นระบบเครือข่ายจึงเกิดขึ้น ปัญหาที่เกิดขึ้นคือการเชื่อมต่อระหว่างคอมพิวเตอร์ที่ต่างระบบกันเป็นสิ่งที่ทำได้ยาก เนื่องจากขาดมาตรฐานส่วนกลางที่จำเป็นต้องใช้ในการรับส่งข้อมูล คอมพิวเตอร์ในชุดนี้เรียกว่า ระบบปิด (Closed System)

ปัญหานี้ทำให้หน่วยงานมาตรฐานสากล คือ International Standards Organization หรือ ISO ทำการกำหนดโครงสร้างทั้งหมดที่จำเป็นต้องใช้ในการสื่อสารข้อมูลจากคอมพิวเตอร์ระบบหนึ่งไปยังคอมพิวเตอร์อีกระบบหนึ่งได้ ซึ่งจะเรียกการทำงานในลักษณะนี้ว่า ระบบเปิด (Open System) เราเรียกโครงสร้างการรับส่งข้อมูลนี้ว่า Open System Interconnection หรือ OSI Model

OSI Model กำหนดให้การสื่อสารข้อมูลจากระบบคอมพิวเตอร์หนึ่งไปยังอีกระบบหนึ่ง แบ่งออกเป็น 7 ชั้นย่อยๆ ซึ่งคอมพิวเตอร์ทั้งสองระบบจะมีชั้นตอนทั้ง 7 ชั้นนี้เหมือนกันทั้ง 2 ฝ่าย เราเรียกชื่อเต็มของการสื่อสารข้อมูลนี้ว่า OSI 7-Layer Reference Model



รูป 2-1 OSI Model Layers

Layer ที่ 7 Application Layer

เป็นชั้นที่อยู่บนสุดของขบวนการรับส่งข้อมูล ทำหน้าที่ติดต่อกับผู้ใช้ โดยจะรับคำสั่งต่างๆจากผู้ใช้งานและส่งให้คอมพิวเตอร์แปลความหมาย และทำงานตามคำสั่งที่ได้รับในระดับโปรแกรมประยุกต์

Layer ที่ 6 Presentation Layer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นชั้นที่ทำหน้าที่ตกลงกับคอมพิวเตอร์อีกด้านหนึ่งในชั้นเดียวกันว่า การรับส่งข้อมูลในระดับโปรแกรมประยุกต์จะมีขั้นตอนและข้อบังคับอย่างไร

Layer ที่ 5 Session Layer

ทำหน้าที่ควบคุมจังหวะในการรับส่งข้อมูลของคอมพิวเตอร์ทั้งสองด้าน ที่รับส่งแลกเปลี่ยนข้อมูลกันให้มีความสอดคล้องกัน (Synchronization) และกำหนดวิธีที่ใช้รับส่งข้อมูล

Layer ที่ 4 Transport Layer

ทำหน้าที่ต่อการรับส่งข้อมูลระดับสูงของ Layer ที่ 5 (ซึ่งมองข้อมูลอยู่ในรูปที่เรียกว่า dialog หรือประโยคของข้อมูลที่โต้ตอบกัน) มาเป็นข้อมูลที่รับส่งในระดับฮาร์ดแวร์

Layer ที่ 3 Network Layer

ทำหน้าที่เชื่อมต่อคอมพิวเตอร์ของด้านรับและด้านส่งเข้าหากันผ่านระบบเครือข่าย พร้อมทั้งเลือกหรือกำหนดเส้นทางที่จะใช้ในการรับส่งข้อมูลระหว่างกัน และส่งผ่านข้อมูลที่ได้รับไปยังอุปกรณ์ในเครือข่ายต่างๆ จนกระทั่งถึงปลายทาง

Layer ที่ 2 Datalink Layer

เป็นชั้นที่ทำหน้าที่เชื่อมต่อการรับส่งข้อมูลในระดับฮาร์ดแวร์ ข้อมูลที่อยู่ใน layer ที่ 2 นี้จะอยู่ในรูปของ Frame คือกลุ่มของข้อมูลที่มีรูปร่างตามข้อบังคับของฮาร์ดแวร์ที่ใช้ในการรับส่งข้อมูล

Layer ที่ 1 Physical Layer

เป็นชั้นล่างสุดของขั้นตอนในการรับส่งข้อมูลของ OSI Model ซึ่งเป็นชั้นเดียวที่มีการเชื่อมต่อกันทางกายภาพระหว่างคอมพิวเตอร์สองระบบที่ทำกรรับส่งจะข้อมูลกัน ซึ่งใน Layer นี้จะกำหนดคุณสมบัติทางกายภาพของฮาร์ดแวร์ที่ใช้เชื่อมต่อกันระหว่างคอมพิวเตอร์ทั้งสองระบบ

สำหรับ TCP/IP ก็มีการแบ่งโปรโตคอลสื่อสารออกเป็นชั้นๆเช่นกัน โดยมีจำนวนชั้นอยู่ 4 ชั้น ทำให้สามารถเปรียบเทียบกับ OSI Model ได้โดยง่าย จะเรียกลำดับชั้นของ TCP ว่า TCP/IP Stack ซึ่งแต่ละชั้นที่ชื่อเรียกกันแตกต่างกันดังนี้

- ชั้นบนเรียกว่า Process Layer จะเป็น Application protocol ที่ทำหน้าที่เชื่อมต่อกับผู้ใช้และให้บริการต่างๆ เช่น FTP, Telnet, SNMP ฯลฯ
- ชั้นถัดมาเรียกว่า Host-to-Host Layer จะเป็น TCP หรือ UDP ที่ทำหน้าที่คล้ายกับ Layer ที่ 4 ของ OSI Model คือควบคุมการรับส่งข้อมูลจากปลายทางด้านส่งถึงปลายทางด้านรับข้อมูล และตัดข้อมูลออกเป็นส่วนย่อยให้เหมาะสมกับเครือข่ายที่ใช้รับส่งข้อมูล รวมทั้งประกอบข้อมูลส่วนย่อยๆนี้เข้าด้วยกันเมื่อถึงปลายทาง
- ชั้นถัดลงมาคือ Internetwork Layer ได้แก่ส่วนของโปรโตคอล IP ซึ่งทำหน้าที่คล้ายกับ Layer ที่ 3 ของ OSI Model คือเชื่อมต่อคอมพิวเตอร์เข้ากับระบบเครือข่ายที่อยู่ชั้นล่างลงไปและทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่เลือกเส้นทางการรับส่งข้อมูลผ่านอุปกรณ์เครือข่ายต่างๆ ในชั้นนี้จะจัดการกับกลุ่มข้อมูลในลักษณะที่เรียกว่า Frame ในรูปแบบของ TCP/IP นั่นเอง

- ชั้นสุดท้ายที่อยู่ล่างสุดคือ เรียกว่า Network Interface คือชั้นที่ควบคุมฮาร์ดแวร์การรับส่งข้อมูลผ่านเครือข่าย ซึ่งเทียบได้กับ Layer ที่ 1 และ 2 ของ OSI Model

2.1.2.1 โพรโทคอล UDP

ใน Host-to-Host Layer นอกจากจะมีโปรโตคอล TCP ทำงานแล้ว ยังมีโปรโตคอล UDP (User Datagram Protocol) ที่มีคุณสมบัติแตกต่างกันอยู่ด้วย ในการรับส่งข้อมูลผ่านโปรโตคอล UDP จะเป็นแบบที่ทั้งสองด้านไม่จำเป็นต้องอาศัยการสร้างการเชื่อมต่อกัน (Connectionless) ระหว่างเครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องที่ขอใช้บริการ โดยไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโปรโตคอล TCP และไม่มีการตรวจสอบความถูกต้องครบถ้วนในการส่งข้อมูลนั้นๆ ด้วย เนื่องจากโปรโตคอล UDP ไม่มีสัญญาณสอบทานข้อมูล (acknowledgement) ในการส่งข้อมูลแต่ละครั้งและไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดในการส่งข้อมูล เมื่อเป็นเช่นนี้แอปพลิเคชันหรือโปรเซสใดที่ต้องการทำงานของโปรโตคอล UDP ในการส่งผ่านข้อมูลก็อาจจะต้องสร้างขบวนการตรวจสอบข้อมูลขึ้นมาเอง

เราจะเห็นได้ว่าโปรโตคอลชั้นบนขึ้นไป ที่ใช้การส่งผ่านข้อมูลโดยโปรโตคอล UDP เช่น โปรโตคอล SNMP (ใช้ควบคุมและจัดการอุปกรณ์ภายในเครือข่าย), หรือโปรโตคอล DHCP (ใช้ส่งข้อมูลพารามิเตอร์ของเครือข่ายให้กับเครื่องลูกข่ายได้ใช้งาน) การส่งข้อมูลเหล่านั้นไม่ต้องรับทราบหรือตรวจสอบว่าข้อมูลไปถึงปลายทางอย่างถูกต้องหรือไม่ แต่กลไกการตรวจสอบข้อมูลที่มีการรับส่งจะไปทำงานในขั้นตอนของโปรโตคอลที่สูงกว่าแทน

ตัวอย่างขั้นตอนกลไกการทำงานโดยใช้โปรโตคอล UDP มีดังนี้

1. ในชั้นของ Process layer เมื่อโปรแกรมควบคุมอุปกรณ์เครือข่ายเช่น โปรแกรม Network management ต้องการส่งข้อมูลไปยังอุปกรณ์ที่ต้องการแอปพลิเคชันนั้นจะติดต่อผ่านโปรโตคอล SNMP ในชั้น Process layer
2. โปรโตคอล SNMP จะติดต่อกับโปรโตคอล UDP ในชั้นถัดไป เพื่อขอติดต่อผ่าน port ที่กำหนด
3. โปรโตคอล SNMP เตรียมข้อมูลที่จะส่งรวมทั้งที่อยู่ปลายทาง
4. โปรโตคอล SNMP ส่งผ่านข้อมูลให้โปรโตคอล UDP ที่อยู่ในชั้น Host-to-Host layer
5. โปรโตคอล UDP ทำหน้าที่ผนึกข้อมูลหรือ datagram นั้น ไปให้กับโปรโตคอล IP ในชั้นถัดลงไป เพื่อส่งข้อมูลออกจากเครื่อง

ซึ่งจะเห็นว่านี่กลไกการส่งข้อมูลที่แตกต่างจากโปรโตคอล TCP ซึ่งจะต้องมีการติดต่อกันก่อนและทั้งสองฝ่ายรับทราบการส่งข้อมูลของช่องทางส่งข้อมูลนั้น

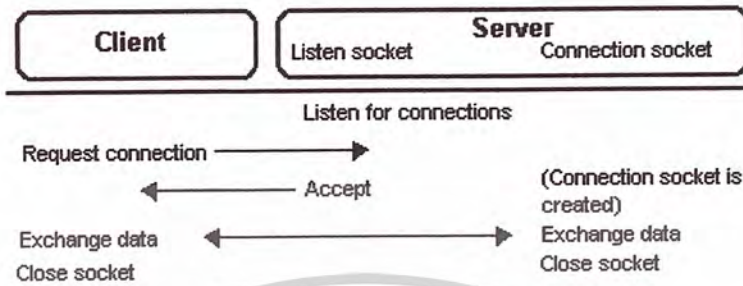
2.1.3 WINSOCK

Winsock คือ API ที่สามารถทำให้เราสร้างและใช้ sockets ได้ การเชื่อมต่อทาง sockets ส่วนใหญ่แล้วจะใช้ผ่านทางอินเทอร์เน็ตหรือแลน การเชื่อมต่อแบบนี้เป็นการเชื่อมต่อแบบ 2 ทางซึ่งหมายความว่าทั้งสองฝ่ายสามารถที่จะทั้งรับและส่งข้อมูลหากันได้ เครื่องคอมพิวเตอร์แต่ละเครื่องที่อยู่บนอินเทอร์เน็ตหรือแลนจะมี IP address ซึ่ง IP address นี้ประกอบไปด้วย 4 ไบต์ ซึ่งคั่นด้วยจุด(.) ตัวอย่างของ IP address เช่น "129.240.3.5" ตัวเลข 4 ตัวแต่ละตัวนั้นเป็น ไบต์ ฉะนั้นมันจึงมีค่าได้ตั้งแต่ 0 ถึง 255 นั่นหมายความว่ามันมีค่าถึง $256 * 4 = 4,294,967,296$ ทำให้มี address ที่ต่างกัน ได้มากกว่า 4 พันล้านค่า ซึ่งเป็นที่เพียงพอสำหรับการใช้งานในปัจจุบัน socket สามารถสร้างจากคอมพิวเตอร์ที่มี address 37.143.125.23 ไปยังเครื่องคอมพิวเตอร์ที่มี address 64.253.241.72 แต่จะมีปัญหาที่คือหากเครื่องคอมพิวเตอร์ต้องการที่จะใช้มากกว่า 1 socket ในเวลาเดียวกัน ดังนั้นเรา address ที่มากขึ้นเพื่อที่จะระบุแต่ละ socket ลงไป address ที่เพิ่มมานี้เราเรียกมันว่าพอร์ตเราใช้พอร์ตอยู่ตลอดเวลาในขณะที่เราใช้อินเทอร์เน็ต ตัวอย่างก็คือ โพรโทคอลพื้นฐาน HTTP ใช้ พอร์ต 80 นั่นก็หมายความว่าหากเราพิมพ์ <http://www.intertainment.8m.com> ในบราวเซอร์ของเรา บราวเซอร์นั้นจะทำการหาว่า server นั้นมี IP address เป็นอะไร และจะทำการเชื่อมต่อไปยัง port 80 พอร์ต 1 – 1000 นั้นมักจะใช้กันมากใน โพรโทคอลพื้นฐานทั่วไป โพรโทคอลก็เปรียบเสมือนกับภาษาที่ใช้ในการสื่อสารกัน อย่างเช่น HTTP นั้นใช้ในการส่งข้อมูลทางเว็บเพจ, FTP ใช้เพื่อส่งไฟล์ข้อมูล อย่างไรก็ตาม winsock สามารถทำได้ทุกอย่างไม่ว่าจะ สร้าง socket, เชื่อมต่อ, จบการเชื่อมต่อ, ปิด, หรือ listen การ listening ก็คือการที่โปรแกรมเตรียมรอรับการเชื่อมต่อจากคอมพิวเตอร์อื่นๆอยู่ที่พอร์ตใดพอร์ตหนึ่ง โปรแกรมที่ listen จากพอร์ตเราเรียกมันว่า daemons ดังตัวอย่าง HTTP daemons จะ listen อยู่ที่พอร์ต 80 และให้ index.html หรือหน้าเพจใดๆก็ตามในการเชื่อมต่อ

การเชื่อมต่อ Client – Server

เครื่องคอมพิวเตอร์ที่เร็วกับอินเทอร์เน็ตหรือแลนที่เร็ว ทำการสร้างเซิร์ฟเวอร์พิเศษ ทุกๆอย่างจะต้องผ่านเซิร์ฟเวอร์ และบางทีเซิร์ฟเวอร์อาจจะต้องทำงานให้กับไคลเอนท์ทั้งหมดด้วย (ซึ่งถ้าหากเกมตัวเกมมีเพียงหน้าจอใหญ่เพียงจอเดียวก็เท่ากับว่าสัดส่วนของการทำงานออกไปเลย) เซิร์ฟเวอร์อาจจะต้องหรือไม่ต้องคำนวณการทำงานก็ได้ ขึ้นอยู่กับโปรแกรมเกมว่ากำหนดให้มีการทำงานอย่างไร สิ่งที่เป็นความสามารถเด่นของเซิร์ฟเวอร์ก็คือมันสามารถรับผู้เล่น ได้อย่างไม่จำกัดและเซิร์ฟเวอร์สามารถทำงานในขณะที่ไคลเอนท์ทั้งหมดเพียงแต่แสดงสิ่งที่เซิร์ฟเวอร์ส่งมาที่ตนเองเท่านั้น และอีกอย่างหนึ่งคือเซิร์ฟเวอร์ไม่จำเป็นต้องทำงานบนเครื่องคอมพิวเตอร์เครื่องใดเครื่องหนึ่ง แต่สามารถทำงานได้ในเซิร์ฟเวอร์แบบผู้เล่นหลายคนซึ่งมีชื่อโดเมน ดังนั้นเราจึงไม่ต้องพิมพ์ IP address หรือเปลี่ยนมันเลย ไคลเอนท์ – เซิร์ฟเวอร์ ของเกมมักจะมี ล็อบบี้ ซึ่งเราสามารถจะเจอผู้เล่นคนอื่นที่รออยู่หรือกำลังพูดคุยกันอยู่ได้

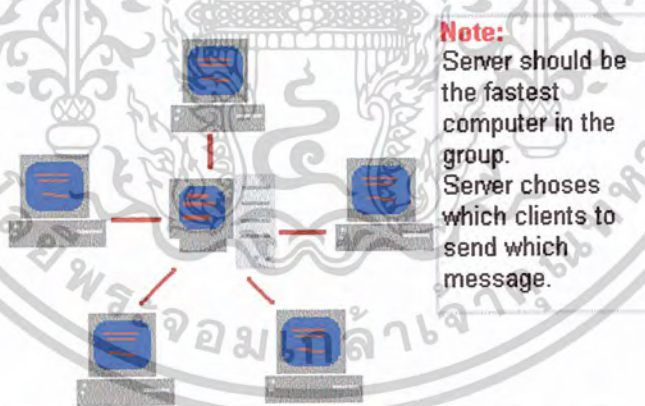
รูปนี้คือแผนผังของ ไคล์แอน – เซิร์ฟเวอร์



รูปที่ 2-2 แผนผังของ ไคล์แอน – เซิร์ฟเวอร์

ในแผนภาพมี socket อยู่ 3 socket หนึ่งก็คือ socket ของไคล์แอน ซึ่งก็คือ socket ที่เครื่องคอมพิวเตอร์ไคล์แอนเป็นผู้ใช้ เครื่องเซิร์ฟเวอร์ใช้ socket 2 socket แต่ในตอนแรก มันจะใช้เพียง socket เดียวเท่านั้น ซึ่งนั่นก็คือ socket ที่ไว้ใช้ในการ listen ซึ่งมันจะคอยlisten จากพอร์ตที่ไคล์แอนรู้จัก เมื่อพอร์ตที่ใช้ listen ขอมรับการเชื่อมต่อ มันจะทำการสร้าง socket ที่ใช้ในการเชื่อมต่อเพิ่มอีก 1 socket ที่มันทำเช่นนี้ก็เพราะว่าหาก ไคลแอนใช้พอร์ตที่ใช้ในการ listen ในการสื่อสารส่งข้อมูลกับเซิร์ฟเวอร์แล้ว คนอื่นก็ไม่สามารถที่จะติดต่อกับเซิร์ฟเวอร์ได้

ภาพแสดงแผนผังของคอมพิวเตอร์หลายๆเครื่องเล่นเกมไคล์แอน – เซิร์ฟเวอร์



รูปที่ 2- 3 แผนผังของคอมพิวเตอร์หลายๆเครื่องเล่นเกมไคล์แอน – เซิร์ฟเวอร์

ปัญหาที่เกิดขึ้นกับอินเทอร์เน็ตและแลนก็คือมันมักจะช้าเกินไป นั่นหมายถึงว่ามันมีความสำคัญมากกับการกำหนดขนาดของแพ็คเกจข้อมูล และความเร็วในการส่งข้อมูล ซึ่งขึ้นอยู่กับตัวเกมว่าเซิร์ฟเวอร์จะได้ทำงานเยอะหรือน้อย บางเกมอาจใช้สถาปัตยกรรมแบบผสม นั่นก็คือไคล์แอนไม่ต้องส่งข้อมูลผ่านเซิร์ฟเวอร์ก็ได้ ซึ่งมันทำให้การทำงานนั้นง่ายและไวขึ้น
จบเรื่องของทฤษฎี winsock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Winsock มีสองเวอร์ชัน คือเวอร์ชัน 1.1 และ เวอร์ชัน 2.1 แนะนำให้ใช้เวอร์ชัน 2 หากว่าไม่ต้องใช้ TCP/IP เราจะเขียนเกม โคล์แอน – เจฟเวอร์ เป่ายังจูบ ทดลอง โคล์แอนจะเป็น โปรแกรมแบบใช้ได้หลายๆเทรค และ ใช้ directX และ win32 app.

เมื่อเริ่มโปรแกรมเราจะต้อง #include เข้าไปที่ต้นของไฟล์ที่จะใช้ winsock และเราต้องทำการเพิ่ม ws2_32.lib เข้าไปในโปรเจกต์ด้วย

Data types :

sockaddr

sockaddr ใช้ในการกำหนดความแตกต่างของการเชื่อมต่อในแต่ละsocket ลักษณะข้อมูลของตัวมันนั้นมีไว้เพื่อใช้เพื่อเก็บข้อมูลเกี่ยวกับ socket (เบอร์ของพอร์ต, IP address, อื่นๆ) ซึ่งถูกตอบรับ โดยเซฟเวอร์ โดยจะมีแค่โปรแกรมของทางเซฟเวอร์เท่านั้นที่จะใช้ข้อมูลนี้

sockaddr_in

sockaddr_in ใช้ในการกำหนดความแตกต่างของการเชื่อมต่อในแต่ละsocket ซึ่งมันจะบรรจุส่วนของข้อมูลที่จะกำหนดความแตกต่างของแต่ละพอร์ต เวอร์ชันนี้ของsockaddr เป็น TCP ใช้มันให้มากที่สุดเท่าที่เป็นไปได้ ลักษณะนี้จะใช้เมื่อมีการสร้าง socket

```
struct sockaddr_in
{
    Short sin_family;
    u_short sin_port;
    struct in_addr sin_addr;
    char sin_zero[8];
}
```

WSAData

WSAData จะใช้เมื่อเราทำการ โหลดและกำหนดค่าเริ่มต้นให้กับ ws2_32.dll โดยทำได้โดยเรียกใช้ ฟังก์ชัน WSAStartup() ใช้ฟังก์ชันนี้เพื่อกำหนดให้คอมพิวเตอร์ใช้ winsock ให้ถูกเวอร์ชัน

SOCKET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOCKET เป็นข้อมูลที่ใช้ในการเก็บข้อมูล socket handles ซึ่งค่า handles นี้มีไว้เพื่อระบุ socket โดยที่ SOCKET นี้กำหนดเป็นค่า unsigned int

ขั้นตอนการทำงาน

```

โหลด ws2_32.dll ได้โดยใช้โค้ดดังนี้
WSADATA w;
int error = WSAStartup (0x0202, &w);

if (error)
{
    return;
}
If (w.wVersion !=0x0202)
{
    WSACleanup ();
    Return;
}

```

0x0202 ในโค้ดนั้นหมายถึงเวอร์ชัน 2.2 แต่หากว่าเป็นเวอร์ชัน 1.1 ไม่ได้หมายความว่ามันจะเปลี่ยนเป็น 0x0101 ไฟล์ของ WSAStartup () ใน โครงสร้างของ WSADATA และ โหลดไลบรารี WinSock2 ฟังก์ชัน WSACleanup () ทำหน้าที่ยกเลิกการ โหลด WinSock DLL

เมื่อต้องการจะสร้าง socket

```
SOCKET s = socket (AF_INET, SOCK_STREAM, 0);
```

เพียง code ก็สามารที่จะสร้าง socket ได้แล้ว แต่เราจะต้องทำการ bind มันกับพอร์ตในภายหลังด้วย เมื่อถึงเวลาที่ต้องการใช้มัน AF_INET เป็นค่าคงที่ ที่ประกาศอยู่ใน winsock2.h หากว่ามีฟังก์ชันที่ต้องการถึง ข้อมูลบางอย่างที่เกี่ยวกับ address family (หรือ int af) ให้เราใช้ AF_INET SOCK_STREAM เป็นค่าที่บอก Winsock ให้รู้ว่าเราต้องการใช้ stream socket (TCP/IP) แต่ถึงกระนั้นเราก็ยังสามารถที่จะใช้ datagram socket

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ เพียงแต่ว่าความน่าเชื่อถือในการส่งข้อมูลมีค่า เราจะปล่อยให้ parameter ตัวสุดท้ายเป็น 0 ซึ่งจะทำให้มันเลือก โพรโทคอลที่ถูกต้องให้กับเรา (ซึ่งน่าจะต้องเป็น TCP/IP)

เมื่อต้องการกำหนดพอร์ตให้กับ socket (เรียกว่าการ bind socket)

```

Sockaddr_in addr;

addr.sin_family = AF_INET;
addr.sin_port = htons(5001);
addr.sin_addr.s_addr = htonl(INADDR_ANY);
if (bind(s, (LPSOCKADDR)&addr, sizeof(addr)) ==
SOCKET_ERROR)
{
    WSACleanup();
    return;
}

```

addr จะอธิบายถึง socket โดยกำหนดความแตกต่างของแต่ละ port โดยที่ IP address นั้นเราจะกำหนดให้มันเป็น INADDR_ANY ซึ่งมันจะอนุญาตให้มันเป็น IP ADDRESS ใดๆก็ได้เพราะเราไม่สนใจเรื่องของ IP ADDRESS เพราะเราต้องการเพียงบอกให้ Winsock รู้ว่าพอร์ตไหนที่เราต้องการใช้ในการเชื่อมต่อ หากเราใช้พอร์ตเบอร์ 7134 (ซึ่งเป็นแบบสั้น) เราจะใช้ฟังก์ชัน htons () เราจะใช้ฟังก์ชัน htonl () กับ IP address แต่หากเราต้องการจำแนกแต่ละ IP address จริงๆละก็เราจะไม่ใช่ htonl () แต่จะใช้ inet_addr () ตัวอย่างเช่น inet_addr ("129.42.12.241") inet_addr จะทำการกระจาย string และเอาจุดออกจากนั้นจะทำการแปลงค่าให้กลายเป็นข้อมูลชนิด long

เมื่อต้องการใช้พอร์ตในการ listen

```

if (listen(s,5) == SOCKET_ERROR)
{
    WSACleanup();
    return;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงตอนนี้เมื่อมีไคลเอนต์ติดต่อเข้ามา มันก็จะทำการตอบรับการเชื่อมต่อนั้น สิ่งที่เป็นจุดพิเศษในcode ข้างต้นก็คือ listen (SOCKET s,int backlog) backlogคือจำนวนของไคลเอนต์ที่สามารถเชื่อมต่อเข้ามาได้ในขณะที่ socket กำลังถูกใช้งานอยู่ เช่นหากเรากำหนด backlog ให้มีค่าเท่ากับ 5 และมีไคลเอนต์พยายามติดต่อเข้ามา 7 คน ไคลเอนต์ 2 คนที่เหลือจะได้รับ ข้อมูลerror และต้องทำการเชื่อมต่อเข้ามาใหม่อีกครั้งในภายหลัง backlog มีค่าตั้งแต่ 2 ถึง 10 กำลังพอดี ขึ้นอยู่กับว่าจะมีคนใช้งานเซิร์ฟเวอร์มากขนาดไหน

เมื่อต้องการเชื่อมต่อกับ socket

```
sockaddr_in target;

target.sin_family = AF_INET;
target.sin_port = htons (5001);
target.sin_addr.s_addr = inet_addr ("52.123.72.251");

if (connect (s, target, sizeof(target)) == SOCKET_ERROR)
{
    WSACleanup;
    return;
}
```

code เพียงเท่านี้เราก็สามารถที่จะทำการร้องขอการเชื่อมต่อได้แล้ว target จะกำหนดเป็นค่าของ socket ที่เราต้องการจะทำการร้องขอการเชื่อมต่อไป ฟังก์ชัน connect () ต้องการค่า socket ที่ถูกต้อง, คำอธิบาย ของ socket เป้าหมาย,และขนาดของความยาวของคำอธิบายนั้น ฟังก์ชันนี้จะทำการส่งการขอการเชื่อมต่อและ จะทำการรอการตอบรับการร้องขอนั้น หรือ รายงานผล error ที่เกิดขึ้น

เมื่อเกิดการตอบรับการเชื่อมต่อ

```
#define MAX_CLIENTS 5;

int number_of_clients = 0;
SOCKET client[MAX_CLIENTS];
sockaddr client_sock[MAX_CLIENTS];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (number_of_client < MAX_CLIENTS)
{
    client[number_of_clients] =
        accept (s, client_sock[number_of_clients], &addr_size);
    if (client[number_of_clients] == INVALID_SOCKET)
    {
        WSACleanup ();
        return;
    }
    else
    {
        startThread (client[number_of_clients]);
        number_of_client++;
    }
}

```

MAX_CLIENTS ไม่จำเป็นมากนัก แต่ใช้เพื่อทำให้code ตัวอย่างนี้ดูเข้าใจได้ง่ายขึ้น number_of_client ใช้สำหรับนับว่ามีไคลเอนต์ติดต่อเข้ามากี่คนแล้ว client[MAX_CLIENTS]เป็นarray ของsocket ที่ใช้เพื่อสำหรับกำหนด handles ของ socket ที่กำลังติดต่อกับไคลเอนต์อยู่ client_sock[MAX_CLIENTS] เป็น array ของ sockaddr ซึ่งใช้ในการเก็บข้อมูลเกี่ยวกับชนิดของการเชื่อมต่อ,พอร์ตที่ใช้ และอื่นๆ โดยปรกติแล้วเราไม่ต้องไปยุ่งเกี่ยวกับ client_sock แต่ว่าฟังก์ชันจะต้องการมันในฐานะที่เป็น parameter ตัวหนึ่ง โดยสรุปแล้วการทำงานของมันเป็นคือ จะรองนกว่าจะมีไคลเอนต์ร้องขอการเชื่อมต่อเข้ามา มันก็จะคอยรับการเชื่อมต่อและเริ่มการทำงานของเทรดที่จะทำการสื่อสารกับไคลเอนต์

การส่งข้อมูล

```

char buffer[11];

printf (buffer, "Whatever...");

send (s, buffer, sizeof(buffer), 0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

parameter ตัวที่สองของฟังก์ชัน send () เป็นค่าคงที่ประเภท char FAR *buf และเป็นตัวชี้ไปที่ buffer ของข้อมูลที่เราต้องการจะส่ง parameter ตัวที่สามนั้นเป็นข้อมูลประเภท int ซึ่งทำหน้าที่บอกความยาวของ buffer ที่เราส่งไป parameter ตัวสุดท้ายนั้นคือ flag ซึ่งเราจะไม่ต้องใช้ กำหนดเป็น 0

การรับข้อมูล

```
char buffer[80];
recv (s, buffer, sizeof(buffer),0);
```

การรับข้อมูลก็มีรายละเอียดเหมือนกับการส่งข้อมูล เพียงแต่คราวนี้เราไม่ได้ส่ง buffer เปลี่ยนเป็นรับ buffer

การแยก IP address หรือ URL

```
u_long addr = inet_addr(Host);
if (addr == INADDR_NONE)
{
    hostent* HE = gethostbyname(Host);
    if (HE == 0)
    {
        WSACleanup ( );
        return;
    }
    addr = *((u_long*) HE->h_addr_list[0]);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

code นี้ทำการแยก const char *Host เป็น u_long addr ซึ่งเราสามารถใช้นั้นเชื่อมต่อกับคอมพิวเตอร์เครื่องอื่นได้

เมื่อต้องการจะปิด socket

```
shutdown (s, SD_SEND);
```

```
closesocket (s);
```

ฟังก์ชัน shutdown(SOCKET s, int how) ในตัวแปร how นั้นมีค่าที่เป็นไปได้ดังนี้

- SD_SEND หมายถึง socket ไม่สามารถส่งข้อมูลได้อีกต่อไป
- SD_RECEIVE หมายถึง socket ไม่สามารถรับข้อมูลได้อีกต่อไป
- SD_BOTH หมายถึง socket ไม่สามารถทั้งรับและส่งได้

2.2 เทคโนโลยีที่ใช้สำหรับแอปพลิเคชันบนมือถือ

2.2.1 Java Server Pages (JSP)

Java Server Pages (JSP) เป็น web-scripting เทคโนโลยีคล้ายกับ Netscape server-side JavaScript (SSJS) หรือ Microsoft Active Server Pages (ASP) แต่ผิดกันตรงที่หัวใจของ JSP คือ Java ซึ่งเป็นภาษาที่ค่อนข้างปลอดภัยที่ออฟเจ็ค (object-oriented style) ซึ่งช่วยทำให้ง่ายต่อการพัฒนาในโปรเจกใหญ่ ๆ ตลอดจนสามารถนำส่วนประกอบต่าง ๆ กลับมาใช้ได้อีก (software reusable) จุดเด่นที่สำคัญของ JSP คือสามารถทำงานได้โดยไม่ต้องขึ้นอยู่กับผู้ผลิตซอฟต์แวร์รายใดรายหนึ่งโดยเฉพาะ ซึ่งโดยทั่วไปเทคโนโลยีต่าง ๆ มักจะออกมาในลักษณะของผลิตภัณฑ์จากบริษัทผู้ผลิตแห่งใดแห่งหนึ่ง แต่ JSP ใช้ลักษณะของ specification ซึ่งกำหนดโดย Sun Microsystems ดังนั้นผู้ผลิตซอฟต์แวร์จึงสามารถอ้างอิง specification ที่กำหนดขึ้น ผลิต JSP Container (ตัวที่ใช้ในการรัน JSP) ขึ้นมาใช้กับแพลตฟอร์มใดก็ได้

JSP Containers

JSP Pages (ไฟล์ที่เขียนขึ้นโดยใช้ JSP script และลงท้ายด้วย .jsp) จะถูกรันโดย JSP Container ซึ่งมักจะเป็นส่วนประกอบที่อยู่ใน Webserver หรือ เป็นตัวเอทอนใน Application Server. โดยทั่วไป JSP Container จะเป็นตัวรับ request จาก client ส่งผ่านไปยัง JSP Page และส่งค่าที่ได้จากการประมวลผลโดย JSP Page กลับไปยัง client. JSP Container ที่ใช้กันอยู่มีมาจากหลายค่าย ยกตัวอย่างเช่น [GNU JSP](#), [Expresso](#), [Tomcat Jakarta](#), [Resin](#), [Weblogic](#) เป็นต้น JSP Container ที่เราจะใช้กันคือ Jasper ซึ่งเป็น JSP Container ที่อยู่ใน Tomcat

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Servlet Engine จากค่าย apache (www.apache.org) ซึ่งทาง Sun ใช้เป็นตัวอ้างอิงในวงการ JSP โดยในอนาคต Tomcat จะเป็นตัวที่ใช้ในการรัน Servlet แทน Jserv ซึ่งใช้กันอยู่ในปัจจุบัน

การติดตั้ง Tomcat

1. Java Development Kit

โปรแกรม Tomcat ถูกเขียนขึ้นโดยใช้ Java ซึ่งจะรันกับ Java Development Kit (JDK) เวอร์ชัน 1.1 และ 1.2 ดังนั้นก่อนจะรัน เราจะต้องติดตั้ง JDK1.xxx ตามแพลตฟอร์มที่เราใช้เสียก่อน สำหรับวินโดวส์และยูนิกซ์ให้ไปดาวน์โหลดที่ <http://java.sun.com/jdk/>. (บางครั้ง JDK อาจถูกเรียกว่า SDK) สำหรับ Linux ให้ไปที่ <http://www.blackdown.org/>. สมมุติว่าเราใช้วินโดวส์ และติดตั้ง JDK1.2.2 ไว้ที่ D:\jdk1.2.2 ให้เซต PATH=d:\jdk1.2.2\bin โดยใช้คำสั่งใน Dos (set PATH=d:\jdk1.2.2\bin;) หรือ คลิกขวาที่ My Computer แล้วเลือกที่ Environment แท็บ ที่เราต้องเซต PATH เพราะ Tomcat จะรันโดยใช้คำสั่ง java.exe ซึ่งอยู่ใน bin ไดรเรททอรีของ JDK

2. Jakarta Tomcat

ทาง apache ยังคงทำการพัฒนา Tomcat ให้มีประสิทธิภาพมากเพื่อใช้เป็นมาตรฐานในอนาคต ดังนั้นถ้าเข้าไปที่เว็บไซต์ในส่วนของดาวน์โหลด เราจะเห็นตัวเลือกในการดาวน์โหลดได้คร่าว ๆ เป็นสองแบบ คือ แบบ source files ที่ต้องมาคอมไพล์เองโดยใช้เครื่องมือช่วยที่เรียกว่า [Ant](#) ซึ่งเป็นส่วนที่นักพัฒนามิชอบดาวน์โหลดกัน กับแบบที่คอมไพล์แล้ว (binary) พร้อมทั้งจะรันซึ่งเหมาะสำหรับคนที่อยากนำ Tomcat มาใช้งานอย่างเดียว และเป็นอย่างที่เราจะใช้ ดังนั้น ให้ไปที่ <http://jakarta.apache.org/downloads/binindex.html> แล้วทำการติดตั้งสมมุติว่าเราได้ Tomcat ไว้ที่ D:\jakarta-tomcat ให้เปิด Command Prompt (MSDos) จาก Application Menu หรือใช้ start->run->type cmd แล้วไปที่ไดเรกทอรีที่เป็น bin ซึ่งในตัวอย่างก็คือ D:\jakarta-tomcat\bin แล้วรันโดยใช้คำสั่ง D:\jakarta-tomcat\bin> startup หลังจากนั้นให้เปิด browser เพื่อเช็คว่า Tomcat ทำงานถูกต้องหรือเปล่า โดยไปที่ <http://127.0.0.1:8080/examples/jsp/>

Note: <http://127.0.0.1:8080> คือ ให้ browser ส่ง request ไปที่เครื่องตัวเอง (localhost) ที่พอร์ต 8080 (Tomcat default port) ถ้าจะปิด Tomcat ก็ให้พิมพ์ D:\jakarta-tomcat\bin> shutdown ลงไปแทน

Web Applications

ก่อนที่เราจะเริ่มเขียน JSP สิ่งหนึ่งที่ควรจะทำคือทำความเข้าใจโครงสร้างของไดเรกทอรีต่าง ๆ สำหรับไฟล์ของเราใน Tomcat ในสมัยก่อนเราไม่ค่อยได้ให้ความสนใจกับการจัดเรียงไฟล์ต่าง ๆ ซักเท่าไร ข้อเสียที่เห็นได้ชัดคือการยากในการจัดเก็บ และการยากในการย้ายไฟล์จากเซฟเวอร์หนึ่ง ไปยังอีกเซฟเวอร์หนึ่ง เพราะโดยทั่วไปแต่ละ webserver จะมีลักษณะการจัดเก็บไฟล์ไม่เหมือนกัน แต่เมื่อ Java Servlet Specification เวอร์ชัน 2.2

61525

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกมา เซฟเวอร์ที่ไชรัน Servlet v2.2 จะถูกบังคับให้มีต้องการสนับสนุนการจัดเก็บไฟล์แบบหนึ่ง ที่เรียกว่า Web Application

web application คือกลุ่มของไครเรททอรี่และไฟล์ที่อาจจะประกอบด้วย html, jsp, servlet, javabean และอื่น ๆ ซึ่งอยู่รวมกันในลักษณะของระบบไฟล์ (file system) หรือถูกอัดอยู่ในไฟล์เดียวกันโดยจะเรียกว่า Web Archive (.war) ไฟล์ ซึ่งมีประโยชน์ในการโยกย้ายและติดตั้งจากเซฟเวอร์หนึ่งไปยังอีกเซฟเวอร์หนึ่ง Tomcat เป็น Servlet Engine ที่สนับสนุน Java Servlet API v.2.2 ดังนั้น ไฟล์ JSP ที่เราเขียน จะต้องถูกจัดอยู่ในรูปแบบ Web Application นี้ด้วย

ถ้าใครเคยคลุกคลีกับ webserver จะเห็นว่ามีการใช้ไครเรททอรี่ในการจัดกลุ่มของไฟล์ไว้ด้วยกันตามหน้าที่ ซึ่งทาง Tomcat ก็ใช้หลักการเดียวกันในการจัดการ web application หลักการก็คือ แต่ละ web application จะถูกจัดให้อยู่ในไครเรททอรี่ของตัวเอง ที่เรียกว่า context path ยกตัวอย่างเช่น ถ้า web application ของเราถูกแมปกับ context path ที่ชื่อ webapplication1 เวลาดูจาก URL ก็จะเห็นเป็น

<http://www.mycompany.com/webapplication1/> เป็นต้น

อย่างที่กล่าวมาข้างต้นว่า web application คือกลุ่มของไครเรททอรี่และไฟล์ที่ถูกจัดอยู่รวมกันเป็นชั้น ๆ ดังนั้น ถ้าเรามีไฟล์ที่ชื่อ helloworld.html :ซึ่งอยู่บนสุด (top-level) ของไครเรททอรี่ใน web application เราก็จะสามารถเรียกออกมาดูได้โดยใช้ URL เท่ากับ

<http://www.mycompany.com/webapplication1/helloworld.html>

ในกรณีที่ไฟล์ไม่ได้อยู่บนสุดของไครเรททอรี่ใน web application เราก็ยังสามารถเรียกออกมาดูได้โดยอ้างอิงไปที่ไครเรททอรี่ย่อยนั้น ยกตัวอย่างเช่น ถ้ากลุ่มไฟล์ JSP ถูกเก็บไว้ในไครเรททอรี่ถัดลงมาที่ชื่อ jsp เราก็สามารถที่จะเรียกไฟล์มาดูโดยใช้ URL เท่ากับ

<http://www.mycompany.com/webapplication1/jsp/xxx.jsp>

ส่วนประกอบของ Web Application

web application หนึ่ง ๆ อาจประกอบด้วย

- 1) Servlet
- 2) JavaServer Pages
- 3) Utility Classes
- 4) ไฟล์ปกติ เช่น HTML, images, sounds
- 5) JavaBean

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละอย่างจะถูกเก็บไว้ในไคลเรคทอรีต่อไปนี้

1. root directory หรือส่วนบนสุด : ไฟล์ *.html, *.jsp มักจะถูกเก็บไว้ส่วนบนสุดของ web application แต่อาจจะมีไคลเรคทอรีย่อยลงมาจาก root เพื่อเก็บ ไฟล์ images ก็ได้

2. WEB-INF/web.xml : ไฟล์ web.xml เป็นส่วนที่ใช้ในการบรรยายส่วนประกอบต่าง ๆ (Web Application Deployment Descriptor) รวมไปถึงการป้อนค่าเริ่มต้นให้กับโปรแกรมต่าง ๆ ที่บรรจุอยู่ใน web application นี้ ยกตัวอย่างเช่น เราสามารถป้อนค่าเริ่มต้นให้กับ Servlet ได้โดยใส่ค่าเข้าไปใน XML Tag ตรงส่วนที่เรียกว่า <init-param> เป็นต้น

3. WEB-INF/classes/* : เป็นส่วนที่ใช้ในการเก็บไฟล์ *.class ทั้งหมดที่ใช้ใน web application ซึ่งอาจจะเป็นทั้ง Servlet หรือ ไฟล์ class ธรรมดาที่ไม่ได้อยู่ในรูปของ .jar ก็ได้ ยกตัวอย่างเช่น ถ้าไฟล์ของเราอยู่ใน package ชื่อ com.mycompany.mypackage.MyServlet ไฟล์นี้ก็จะถูกเก็บอยู่ในไคลเรคทอรี WEB-INF/classes/com/mycompany/mypackage/MyServlet.class

4. WEB-INF/lib/*.jar : ไคลเรคทอรีนี้ใช้เก็บไฟล์ .jar ที่เราต้องการใช้สำหรับ web application ซึ่งอาจจะเป็นส่วนที่ได้มาจาก third party หรือเป็นพวก Database driver ก็ได้ ประโยชน์อย่างหนึ่งที่เราเห็นได้ในการเก็บไลบรารีต่าง ๆ ไว้ที่ไคลเรคทอรีนี้คือ การง่ายต่อการติดตั้ง เหตุผลคือ หลังจาก web application ของเราถูกใส่เข้าไปใน server แล้ว ตัว server จะโหลดไฟล์ที่อยู่ในไคลเรคทอรีนี้โดยอัตโนมัติ เราจึงไม่ต้องทำการเซ็ต CLASSPATH ให้ยุ่งยาก ตัวอย่างง่าย ๆ ของโครงสร้าง web application อาจเป็น

```
/index.html
/howto.jsp
/feedback.jsp
/images/banner.gif
/images/jumping.gif
/WEB-INF/web.xml
/WEB-INF/lib/jspbean.jar
/WEB-INF/classes/com/mycompany/servlets/MyServlet.class
/WEB-INF/classes/com/mycompany/util/MyUtil.class
```

Integration with Tomcat

การติดตั้ง web application สามารถทำได้ 3 วิธีคือ

1. นำ web application ที่อยู่ในรูปของกลุ่มไคลเรคทอรี (ยังไม่ได้ถูกบีบอัด) ไปใส่ไว้ที่ \$TOMCAT_HOME/webapps/. กรณีนี้ Tomcat จะจัด context path ให้ web application เองโดยอิงจากชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

subdirectory ที่บรรจุกลุ่มไคลเรทเทอร์ของ web application ไว้ ยกตัวอย่างเช่น เราใส่ web application ไว้ที่ \$TOMCAT_HOME/webapps/myapp/ ดังนั้น context path ที่จะใช้ติดต่อกับ web application นี้ก็คือ myapp วิธีนี้เป็นวิธีที่ง่ายที่สุด และเหมาะสำหรับการพัฒนาเพื่อนำไปใช้จริง

2. นำ web application archive (.war ไฟล์) ไปใส่ไว้ที่ \$TOMCAT_HOME/webapps/ วิธีนี้เหมาะสำหรับการติดตั้ง web application ใน production server ที่ใช้งานจริง โดย Tomcat จะแกะและขยายไฟล์ทั้งหมดที่อยู่ใน .war ไฟล์ออกมาแล้วทำการโหลดเข้าหน่วยความจำ

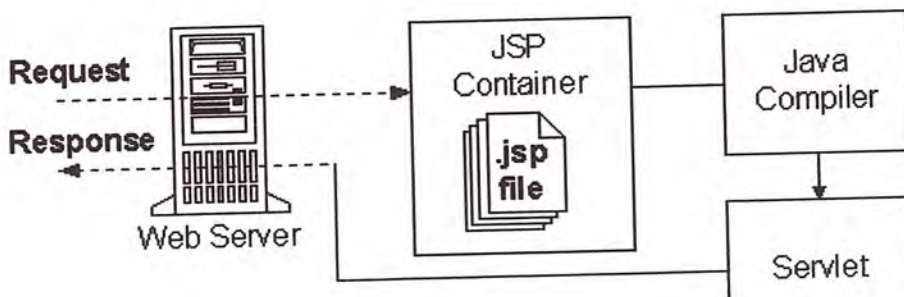
3. ในกรณีที่ไคลเรทเทอร์ที่เราพัฒนา web application ไม่ได้อยู่ที่ \$TOMCAT_HOME/webapps/ เราก็สามารถบอก Tomcat ให้โหลด web application นี้ได้โดยเข้าไปแก้ที่ \$TOMCAT_HOME/conf/server.xml ในส่วนของ <Context> ให้ชี้ไปที่ไคลเรทเทอร์ที่เราเก็บ web application ไว้ เช่น ถ้าเราเก็บ web application ไว้ที่ D:\MyWebApplication\ แล้วอยากให้มีชื่อ MyWebApp เป็น context path ในการชี้ไปที่ web application ของเรา เราก็แค่เอา

```
<Context path="/MyWebApp" docBase="D:\MyWebApplication" debug="0" reloadable="true"/>
```

ไว้ที่บรรทัดถัดมาจาก Context path อันสุดท้ายที่มีอยู่แล้ว เป็นต้น (สมมติว่าเราติดตั้ง Tomcat ไว้ที่ ไคลเรทเทอร์ D:\jakarta-tomcat ดังนั้น \$TOMCAT_HOME จะเป็น D:\jakarta-tomcat)

การทำงานของ JSP

หลังจาก request จาก client ส่งมาถึงเซิร์ฟเวอร์, JSP Container จะทำการแปลงไฟล์ JSP ให้กลายเป็น Servlet source ไฟล์ (ในกรณีที่ JSP ไฟล์ดังกล่าวถูกเรียกเป็นครั้งแรก) ซึ่งไฟล์ Servlet source ที่ได้จะถูกคอมไพล์เป็น .class เพื่อใช้ในการประมวลผล request ของ client แล้วส่งกลับไปให้เซิร์ฟเวอร์ในรูปของ outputStream ซึ่งถูกส่งไปที่ client ในท้ายสุด หลังจากนั้น ถ้า JSP ไฟล์ดังกล่าวถูกเรียกอีก การแปลงไฟล์หรือคอมไพล์จะไม่เกิดขึ้น เพราะ JSP Container จะใช้ไฟล์ .class ที่เก็บไว้แล้วประมวลผลแทน ข้อสังเกตคือ การเรียกไฟล์ JSP ครั้งแรกจะรู้สึกช้า แต่ครั้งถัดไป ๆ จะเห็นได้ว่าเร็วขึ้นมาก เพราะได้มีการลดขั้นตอนต่างๆ ที่เสียเวลาไป ใน JSP Specification 1.1 ยังมีการแนะนำให้ JSP Engine รองรับการโหลดไฟล์ JSP ขึ้นมาคอมไพล์ใหม่แบบอัตโนมัติ (Reloading) ซึ่งมักจะเกิดขึ้นในกรณีที่ JSP ไฟล์ได้มีแก้ไข แล้วจะต้องมีการคอมไพล์ไฟล์ Servlet ที่กำลังใช้งานอยู่ใหม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-4 การใช้งาน jsp

จากข้อความที่อธิบายข้างต้น เราสามารถแบ่งระยะเวลาของไฟล์ JSP หนึ่งๆ ออกเป็นสองช่วงคือ

1. Translation Time คือช่วงเวลาที่ JSP ไฟล์ถูกแปลงให้กลายเป็น Servlet ไฟล์และถูกคอมไพล์ให้กลายเป็น .class ไฟล์ ซึ่งจะเกิดก่อนการรับ request จาก client ครั้งแรก
2. Client Request Time คือช่วงเวลาที่ .class ของ JSP ไฟล์ ทำการรับ request จากแต่ละ client แล้วทำการประมวลผล

2.2.2 J2ME

J2ME เป็นแพลตฟอร์มจาวาที่ออกแบบมาเพื่อใช้งานกับแอปพลิเคชัน ที่ทำงานบนอุปกรณ์ขนาดเล็ก เช่น โทรศัพท์เคลื่อนที่ พีดีเอ โทรศัพท์พร้อมจอภาพที่เชื่อมต่อกับอินเทอร์เน็ต โทรศัพท์ดิจิทัลขนาดเล็ก อุปกรณ์บันทึกและระบบนำทางในรถยนต์ สวิตช์ในระบบเครือข่าย คอมพิวเตอร์ของเครื่องอำนวยความสะดวกภายในบ้าน ฯลฯ

J2ME ได้นำโครงสร้างแบบโมดูลที่มีความยืดหยุ่นสูงเข้ามาใช้ เพื่อให้สามารถสนับสนุนการทำงานของอุปกรณ์หลากหลายประเภท จากรูปที่ 2.1 แสดงให้เห็นว่า J2ME กำหนดชั้นการทำงานของซอฟต์แวร์ไว้ 3 เลเยอร์ด้วยกัน โดยเลเยอร์ทั้งหมดจะอยู่เหนือชั้นระบบปฏิบัติการของอุปกรณ์ ดังนี้

เลเยอร์ Java Virtual Machine เป็นเลเยอร์ของ Java Virtual Machine ปรับแต่งให้เข้ากับระบบปฏิบัติการของอุปกรณ์ และรองรับแต่ละ configuration ของ J2ME จากรูปที่ 2.1 Virtual Machine ของ J2ME ได้แก่ CVM และ KVM

เลเยอร์ Configuration เป็นเลเยอร์ของคอนฟิกูเรชันของ J2ME ซึ่งกำหนดคลาสไลบรารี อุปกรณ์ใช้งานทั่วไปหรือกลุ่มอุปกรณ์ที่มีความต้องการหน่วยความจำและหน่วยประมวลผลใกล้เคียงกัน จากรูปที่ 2.1 Configuration ใน J2ME มี 2 ประเภทด้วยกัน คือ CDC (Connected Device Configuration) และ CLDC (Connected Limited Device Configuration)

- เลเยอร์ Profile เป็นเลเยอร์ที่สร้างเหนือเลเยอร์คอนฟิกูเรชัน โดยกำหนดคลาสไลบรารีเพื่อสนองตอบความต้องการของตลาดเฉพาะกลุ่ม จากรูปที่ 2.1 ตัวอย่างของ Profile ในเลเยอร์นี้ ได้แก่ PDAP, MIDP, Foundation Profile และ Personal Profile

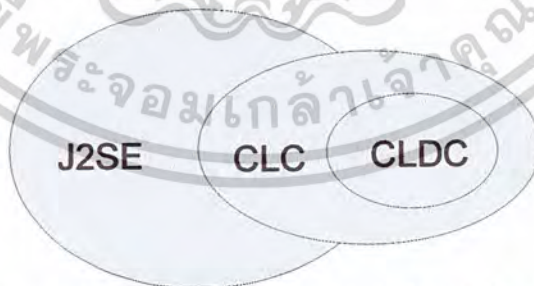
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนพิทุเรชั่นใน J2ME

คอนพิทุเรชั่นและโพรไฟล์เป็นองค์ประกอบหลักของ J2ME โดยมีจุดสำคัญ คือ เพื่อปรับแต่ง Virtual Machine และคลาสไลบรารีให้เหมาะสมกับอุปกรณ์แต่ละประเภท คอนพิทุเรชั่น คือ ชุดคำสั่งที่มีคุณสมบัติขั้นต่ำของ Java Virtual Machine และจาวาคลาสสำหรับอุปกรณ์แต่ละประเภท เป็นตัวแทนของอุปกรณ์ที่ลักษณะคล้ายคลึงกัน อาจกล่าวได้ว่า คอนพิทุเรชั่นเป็นตัวกำหนดคุณสมบัติหรือไลบรารีกำหนดคอนพิทุเรชั่นจะมีในอุปกรณ์ที่อยู่ในกลุ่มเดียวกัน

ปัจจุบัน คอนพิทุเรชั่นใน J2ME แบ่งเป็น 2 ประเภท คือ CDC (Connected Device Configuration) และ CLDC (Connected Limited Device Configuration) โดยมีเป้าหมายที่กลุ่มอุปกรณ์ 2 ประเภทซึ่งมีความต้องการหน่วยความจำและหน่วยประมวลผลพื้นฐานใกล้เคียงกัน

- CDC เป็นอุปกรณ์ใช้ร่วมกับอุปกรณ์อื่น ติดตั้งตายตัว และใช้เชื่อมต่อข้อมูล โดยปรกติมักมีส่วนติดต่อกับผู้ใช้หลากหลายแบบ มีหน่วยความจำประมาณ 2-16 เมกกะไบต์ ใช้หน่วยประมวลผลแบบ 32 บิต หรือมากกว่า เชื่อมต่อเครือข่ายที่มีแบนด์วิดธ์สูงอย่างต่อเนื่อง โดยอาศัยพอร์ต TCP/IP ตัวอย่างอุปกรณ์ประเภทนี้ ได้แก่ โทรศัพท์มือถือขนาดเล็ก อินเทอร์เน็ตทีวี โทรศัพท์พร้อมจอภาพที่เชื่อมต่อกับอินเทอร์เน็ต อุปกรณ์สื่อสารที่มีความซับซ้อนสูง อุปกรณ์บันทึกและระบบนำทางในรถยนต์
- CLDC เป็นอุปกรณ์ส่วนบุคคล พกพาได้ และใช้เชื่อมต่อข้อมูล โดยปรกติมักมีส่วนติดต่อกับผู้ใช้แบบง่ายๆ (เมื่อเปรียบเทียบกับระบบบนเครื่องคอมพิวเตอร์แบบตั้งโต๊ะ) มีหน่วยความจำประมาณ 128 กิโลไบต์ – 1 เมกกะไบต์ ใช้หน่วยประมวลผลแบบ 16 หรือ 32 บิต เชื่อมต่อกับเครือข่ายที่มีแบนด์วิดธ์ต่ำเป็นระยะเวลาสั้นๆ โดยไม่อาศัยพอร์ต TCP/IP ตัวอย่างอุปกรณ์ประเภทนี้ ได้แก่ โทรศัพท์เคลื่อนที่แบบไม่ซับซ้อนมากนัก เพจเจอร์รับส่งข้อความ เครื่องปาล์มไอเอสแบบพกพา



รูปที่ 2-5 แสดงความสัมพันธ์ระหว่าง J2SE และคลาสไลบรารีใน CDC และ CLDC

ในชั้นของ Configuration มีคลาส 2 ประเภทด้วยกัน คือ คลาสที่นำมาจาก J2SE และที่ออกแบบเฉพาะอุปกรณ์ขนาดเล็ก คลาสที่นำมาจาก J2SE จะมีคุณสมบัติอย่างเดียวกันกับคลาสใน J2SE หรือซัพคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ J2SE เช่น แพ็กเกจ java.lang java.io และ java.util จากรูปที่ 2-2 จะแสดงความสัมพันธ์ระหว่าง J2SE และ คลาสไลบรารีของ CDC และ CLDC จากภาพจะเห็นว่าคลาสใน CLDC ส่วนใหญ่ทำงานร่วมกับคลาส CDC ได้ดีเช่นเดียวกับกรณีของคลาสใน CLDC

ขณะเดียวกันคลาสที่ไม่ได้นำมาจาก J2SE และออกแบบเพื่อให้เฉพาะอุปกรณ์มักทำงานร่วมกับ J2SE ได้ไม่ค่อยดี ใน CLDC คลาสเหล่านี้จะอยู่ในกลุ่มกรอบการติดต่อสื่อสารทั่วไป (Generic Connection Framework) โดยระบุไว้ในแพ็กเกจ javax.microedition.io

J2ME สำหรับอุปกรณ์ไร้สาย

J2ME ได้ให้กำเนิดแอปพลิเคชันยุคใหม่บนอุปกรณ์ไร้สาย ช่วยให้เกมแบบหลายผู้เล่นที่ทำงานผ่าน อินเทอร์เน็ต การทำธุรกรรมทางการเงินบนโทรศัพท์มือถือ แอปพลิเคชันสำหรับองค์กรทั้งไคลเอ็นท์และ เซิร์ฟเวอร์ เกิดขึ้นได้บนโทรศัพท์เคลื่อนที่ และเพจเจอร์รับส่งข้อความ MIDP CLDC และ KVM ได้กลายมา เป็นรากฐานในการพัฒนาแอปพลิเคชันบนอุปกรณ์ไร้สายยุคใหม่

สามารถแบ่งโครงสร้างออกเป็นเลขอร์ต่างๆ จากล่างขึ้นบน ดังนี้

- เลขอร์ฮาร์ดแวร์ MID หมายถึง ตัวโทรศัพท์เคลื่อนที่ (เช่น Nokia6600) หรือเพจเจอร์รับส่งข้อความ (เช่น RIM รุ่น Blackberry950)
- เลขอร์ซอฟต์แวร์ของระบบที่ติดตั้งมากับอุปกรณ์ หมายถึง ระบบปฏิบัติการและไลบรารีของระบบที่บริษัทผู้ผลิตให้มา
- เลขอร์ KVM เป็นส่วนที่เตรียม runtime environment ไว้ให้แอปพลิเคชันบนอุปกรณ์ไร้สาย
- เลขอร์ CLDC เป็นส่วนที่เตรียม API หลักของจาวาให้แอปพลิเคชันบนอุปกรณ์ไร้สาย
- เลขอร์ MIDP (Mobile Information Device Profile) เป็นส่วนที่เตรียมไลบรารีสำหรับส่วนติดต่อกราฟฟิกกับผู้ใช้ (User Interface) และหน่วยเก็บข้อมูล persistent storage ระบบเครือข่าย และไทม์เมอร์

นอกจากคลาสไลบรารีสำหรับ MIDP อาจเตรียมคลาสไลบรารีเฉพาะอุปกรณ์ไว้ให้นักพัฒนา เพื่อถึงความสามารถของฟังก์ชันที่มีอยู่แล้วไปใช้งานให้เกิดประโยชน์สูงสุด อาทิ การโทรศัพท์ การแชร์ข้อมูลกับแอปพลิเคชันที่ติดตั้งมาในเครื่อง (เช่น ปฏิทิน สมุดจดที่อยู่) การตรวจสอบข้อมูลอุปกรณ์ที่ผู้ผลิตเตรียมไว้มาใช้งาน จะช่วยเพิ่มความสามารถแก่แอปพลิเคชันบนอุปกรณ์ไร้สาย แต่ไม่สามารถเคลื่อนย้ายไปสู่อุปกรณ์อื่นที่ใช้ MIDP ได้ เนื่องจากคลาสที่นำมาใช้ออกนอกเหนือขอบเขตของ MIDP

ความต้องการของระบบ

อุปกรณ์ไร้สายจะทำงานสนับสนุน J2ME ได้ดีเมื่อมีคุณสมบัติตรงตามข้อกำหนด หากต้องการให้ KVM ทำงานได้อย่างมีประสิทธิภาพ โดยใช้ไลบรารี CLDC จะต้องมีคุณสมบัติของระบบขั้นต่ำ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีหน่วยความจำขนาด 160 – 512 กิโลไบต์ สำหรับสร้างแพลตฟอร์มจาวา
- มีหน่วยประมวลผลแบบ 16 – 32 บิต ความเร็ว 25 เม็กกะเฮิรตซ์
- ใช้พลังงานน้อย โดยมากมักทำงานโดยใช้แบตเตอรี่
- เชื่อมต่อกับเครือข่ายได้ในช่วงสั้นๆ (อาศัยระบบไร้สายเป็นส่วนใหญ่) แบบควิควิคก่อนข้างจำกัด (ความเร็ว 9600 ไบต์ต่อวินาทีหรือน้อยกว่า)
- มีหน่วยความจำชั่วคราวขนาด 32 กิโลไบต์ สำหรับเก็บจาวา รันไทม์และหน่วยความจำของออบเจกต์

2.2.3 หลักการพัฒนาแอปพลิเคชันบน MIDP ด้วย J2ME

ไลบรารีของ CLDC

ไลบรารีของ CLDC เป็นไลบรารีของเครือข่ายและระบบชั้นสูงที่ไม่จำกัดเฉพาะอุปกรณ์ใดอุปกรณ์หนึ่ง ประกอบด้วยคลาส 2 ประเภท คือ คลาสที่เป็นซัพเซตของ J2SE และคลาสที่เกี่ยวกับกรอบการติดต่อสื่อสารทั่วไป (Generic Connection Framework) ของ CLDC

ซัพคลาสของ J2SE

คลาสประเภทแรกของ CLDC เป็นซัพคลาสของไลบรารี J2SE กำหนดไว้ในแพ็คเกจ `java.lang` `java.io` และ `java.util` โดยคลาสระบบและคลาสชนิดข้อมูลเข้ากันได้กับ J2SE และ J2EE และเพื่อให้เข้ากันได้และเคลื่อนย้ายข้ามแพลตฟอร์มได้โดยสะดวก คลาสประเภทนี้จะใช้ชื่อแพ็คเกจตามชื่อคลาสอย่างเดียวกับใน J2SE หรือตามชื่อซัพคลาสของคลาสใน J2SE ที่เกี่ยวเนื่องกัน โดยไม่มีการเพิ่มเมธอด `public` หรือ `protected` หรือฟิลด์ซึ่งไม่มีในคลาสที่เกี่ยวข้องกันใน J2SE ความหมายของคลาสและเมธอดจึงไม่เปลี่ยนแปลง

คลาสที่มีเฉพาะใน CLDC

คลาสประเภทที่ 2 ของ CLDC กำหนดไว้ในแพ็คเกจ `javax.microedition.io` คลาสเหล่านี้มีเฉพาะใน CLDC จึงไม่สามารถทำงานร่วมกับไลบรารีของ J2SE ได้ เป็นคลาสชั้นสูงเกี่ยวกับเรื่องทั่วไปของเครือข่าย กำหนดไว้ในแพ็คเกจ `java.io` และ `java.net` คลาสประเภทนี้จะเตรียมกรอบการติดต่อสื่อสารผ่านเครือข่ายให้กับอุปกรณ์ที่สนับสนุน J2ME โดยกรอบการสื่อสารนี้มักเรียกว่า Generic Connection Framework คลาสส่วนใหญ่ได้แก่อินเทอร์เฟซที่แทนการสื่อสารรูปแบบต่างๆ เช่น ซอกเกต (Socket) คาด้าแกรม (Datagram) ซีเรียล (Serial) และ HTTP ขึ้นอยู่กับว่าผู้ผลิตอุปกรณ์หรือผู้ให้บริการเครือข่าย จะติดตั้งอินเทอร์เฟซการเชื่อมต่อเหล่านี้เพียงบางตัวหรือครบทั้งหมด เมื่อพิจารณาจากความสามารถของอุปกรณ์และเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทของข้อมูลพื้นฐาน

CLDC สนับสนุนเฉพาะช่วงขนาดของชนิดข้อมูลที่นำมาจาก J2SE ซึ่งได้แก่ byte short int long char และ boolean แต่ไม่สนับสนุนข้อมูลประเภท float และ double (ยกเว้น CLDC 1.1 จะสนับสนุนข้อมูลประเภท float) เนื่องจากสาเหตุ 2 ประการ คือ อุปกรณ์เป้าหมายที่นำ CLDC ไปใช้ ส่วนใหญ่ไม่มีฮาร์ดแวร์รองรับเลขทศนิยม และหากใช้จะใช้ซอฟต์แวร์เข้าช่วยก็จะต้องเสียค่าใช้จ่ายสูงมาก

นอกจากนี้ CLDC ยังกำหนดคลาส type wrapper สำหรับข้อมูลทุกประเภท ได้แก่

- Java.lang.Boolean
- Java.lang.Byte
- Java.lang.Character
- Java.lang.Integer
- Java.lang.Long
- Java.lang.Short

คลาส type wrapper กำหนดไว้ในจาวา เนื่องจากจาวามีระบบย่อยหลายระบบซึ่งทำงานได้เฉพาะออบเจกต์เท่านั้น ในกรณีนี้ คุณสามารถสร้างออบเจกต์โดยใช้คลาส type wrapper ซึ่งเก็บชนิดของข้อมูลที่นำมาจาก J2SE เอาไว้

ไลบรารีของ MIDP

ขณะที่ไลบรารีของ CLDC ช่วยสร้างฟังก์ชันที่ไม่จำกัดเฉพาะอุปกรณ์บางชนิด ไลบรารีของ MIDP กลับตรงข้าม ได้แก่ การจัดการแอปพลิเคชันบนอุปกรณ์ ส่วนติดต่อกราฟฟิกกับผู้ใช้ทั้งแบบพื้นฐานและซับซ้อน พื้นที่เก็บข้อมูลแบบคงตัว (persistent storage) และความสามารถเพิ่มเติมบนเครือข่าย

คลาสโปรแกรมจัดการแอปพลิเคชัน

คลาสที่ติดต่อกับ โปรแกรมจัดการแอปพลิเคชันบนอุปกรณ์ถูกกำหนดไว้ในแพ็คเกจ java.microedition.midlet แอปพลิเคชันทั้งหลายที่เขียนเป็น MIDP จะต้องขยายคลาส MIDlet ที่อยู่ในแพ็คเกจออกไป และจะต้องนำเมธอดทั้ง 3 ซึ่งได้แก่ startApp() pauseApp() และ destroyApp() เข้ามาใช้

คลาสของส่วนติดต่อกราฟฟิกกับผู้ใช้ (GUI Class)

ชุดเครื่องมือ Abstract Windowing Toolkit ใน J2SE ออกแบบมาเพื่อใช้งานกับแอปพลิเคชันบนเครื่องคอมพิวเตอร์แบบตั้งโต๊ะและไม่สามารถนำมาใช้กับอุปกรณ์ไร้สายได้ เนื่องจากความต้องการทรัพยากรประเภทหน่วยความจำค่อนข้างมาก MIDP มีวิธีการที่ต่างกันในการกำหนดไลบรารีสำหรับแพ็คเกจส่วนติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟฟิกับผู้ใช้ ด้วยการใช้ API ชั้นสูงซึ่งเน้นความสามารถในการเคลื่อนย้ายข้ามอุปกรณ์ และ API ชั้นพื้นฐานซึ่งเน้นองค์ประกอบทางกราฟฟิเฉพาะอุปกรณ์ และ input event ทั่วไป คลาสที่เชื่อมต่อกับส่วนติดต่อกับกราฟฟิกับผู้ใช้ และ event-handling กำหนดไว้ในแพ็คเกจ java.microedition.lcdui Screen ถือเป็นซูเปอร์คลาสของคอมโพเนนต์ส่วนติดต่อกับผู้ใช้โดยใช้ API ชั้นสูง ประกอบไปด้วย Alert Form List Textbox ฯลฯ Canvas และ Graphic เป็นคลาสหลักของ API ระดับต่ำ (Low-level) แอปพลิเคชันเกมเป็นแอปพลิเคชันที่ใช้ API ระดับต่ำในการสร้างส่วนติดต่อกับผู้ใช้

คลาสของพื้นที่เก็บข้อมูลแบบคงตัว (Persistent Storage Class)

ในบางครั้งแอปพลิเคชันที่เขียนบน MID จำเป็นจะต้องเก็บข้อมูลไว้บนอุปกรณ์อย่างถาวร คลาสที่กำหนดไว้ในแพ็คเกจ java.microedition.rms ให้กลไกการเก็บข้อมูลถาวรที่เรียกว่า RecordStore โดยยอมให้แอปพลิเคชันเขียนลบและปรับปรุงบันทึกข้อมูลใหม่ในหน่วยเก็บข้อมูลถาวรของอุปกรณ์

คลาสของเครือข่าย (Network Class)

แม้ว่ากรอบการติดต่อสื่อสารทั่วไปที่กำหนดไว้ใน CLDC จะประกอบไปด้วยชุดของอินเทอร์เฟซการเชื่อมต่อเครือข่าย แต่ก็ไม่มีโพรโตคอลอยู่เบื้องหลังอินเทอร์เฟซการเชื่อมต่อจริงๆ หากแต่ปล่อยให้ทำหน้าที่ของ MIDP

ในบรรดาอินเทอร์เฟซการเชื่อมต่อเครือข่ายเหล่านี้ Httpconnection ถือเป็นอินเทอร์เฟซหลักที่ต้องมีใน MIDP เสมอ คลาสของอินเทอร์เฟซเหล่านี้ กำหนดไว้ในแพ็คเกจ java.microedition.io

2.2.4 MIDlets

MIDlet หมายถึง แอปพลิเคชันบน MIDP MIDlet มีส่วนคล้ายกับจาวาแอปเพล็ต แม้จะไม่มีเมธอด main() แต่ MIDlet ก็นำคลาส javax.microedition.midlet.MIDlet ตลอดจนเมธอดทั้ง 3 [startApp() pauseApp() destroyApp()] เข้ามาใช้ นอกจากนี้ MIDlet ยังกำหนด constructor แบบ public ที่ไม่มีอาร์กิวเมนต์ใดๆ อีกด้วย เราสามารถให้คำนิยามของคลาส javax.microedition.midlet.MIDlet ได้ดังนี้

```
public abstract คลาส MIDlet extends Object {
    protected MIDlet()
    protected abstract void startApp() throws MIDletStateChangeException
    protected abstract void pauseApp()
    protected abstract void destroyApp() (Boolean unconditional)
        throws MIDletStateChangeException
    public final String getAppProperty(String key)
```

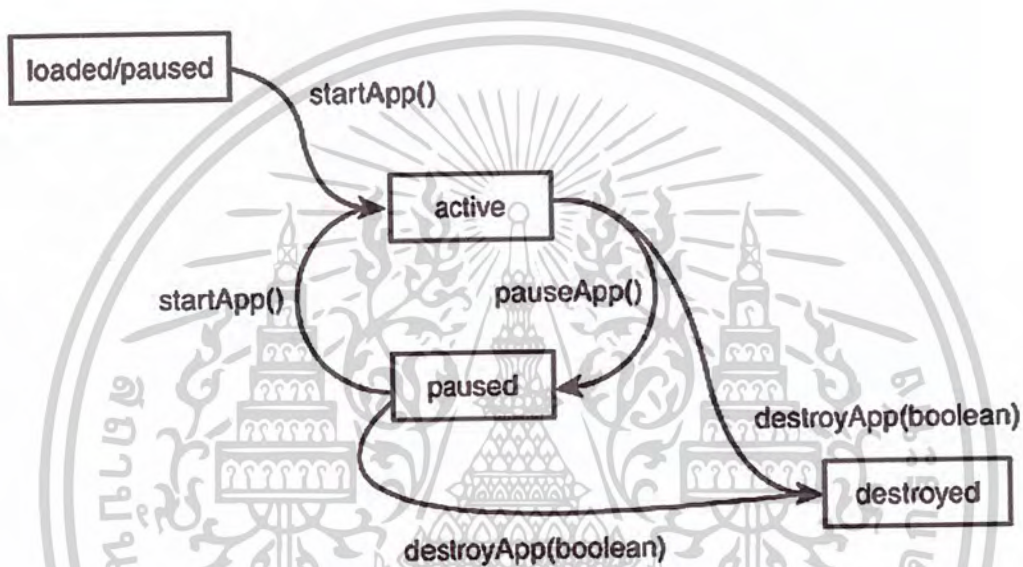
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public final void notifyDestroyed()
public final void notifyPaused()
public final String getAppProperty(String key)
public final void resumeRequest() }

```

คลาส MIDlet ระบุเมธอดที่สามารถเรียกใช้โดยซอฟต์แวร์จัดการแอปพลิเคชัน (AMS แอปพลิเคชัน Management System) เพื่อสั่งให้แอปพลิเคชัน MIDlet เริ่มต้นและหยุดทำงาน



รูปที่ 2-6 วงจรการทำงานของ MIDlet

การกระทำของ MIDlet ประกอบไปด้วย 4 สถานะ คือ บรรลุ กำลังทำงาน หยุดชั่วคราว และถูกทำลาย ออสซีซอฟต์แวร์จัดการแอปพลิเคชันเป็นตัวควบคุมการเปลี่ยนสถานะหนึ่งไปยังอีกสถานะหนึ่ง ควบคุมด้วยเมธอด startApp() pauseApp() และ destroyApp() ที่มาพร้อมกับ MIDlet รูปที่ 2-3 แสดงให้เห็นจุดเปลี่ยนระหว่างสถานะทั้ง 3 โดยการเรียกเมธอดข้างต้น

จากรูปเมื่อ MIDlet ถูกบรรจุลงในอุปกรณ์และ constructor ถูกเรียกแล้ว มันก็จะเข้าสู่สถานะบรรลุ ซึ่งต้องเกิดขึ้นทุกครั้งก่อนที่ซอฟต์แวร์จัดการแอปพลิเคชันจะเริ่มแอปพลิเคชันจะเรียกเมธอด startApp() ขึ้นมา และ MIDlet จะเข้าสู่สถานะกำลังทำงานเปิดรับทรัพยากรที่ต้องการและเริ่มต้นการทำงาน ในสถานะนี้ MIDlet จะทำงานและดึงทรัพยากรที่ต้องการไว้ใช้งาน

เมื่อซอฟต์แวร์จัดการแอปพลิเคชันไม่ต้องการให้ MIDlet ทำงานต่อไป ก็จะเรียกเมธอด pauseApp() จากนั้น MIDlet จะหยุดทำงานและเข้าสู่สถานะหยุดชั่วคราว คืนทรัพยากรที่ดึงมาใช้งานและเข้าสู่ภาวะไม่ทำงาน MIDlet สามารถกลับไปอยู่ที่สถานะกำลังทำงานได้ เมื่อซอฟต์แวร์จัดการแอปพลิเคชันเรียกเมธอด startApp() ขึ้นมา ท้ายที่สุดเมื่อซอฟต์แวร์จัดการแอปพลิเคชันไม่ต้องการเรียกใช้งาน MIDlet หรือต้องการ

เคลียร์หน่วยความจำเพื่อให้โปรแกรมอื่นได้ใช้งาน ก็จะส่งสัญญาณเตรียมทำลาย MIDlet ทิ้ง ด้วยการเรียกเมธอด `destroyApp()` และเข้าสู่สถานะถูกทำลายซึ่ง MIDlet จะปล่อยทรัพยากรทั้งหมด ทำการจัดเก็บข้อมูลถาวรต่างๆ และหยุดการทำงานทั้งหมด

หาก MIDlet อยู่ระหว่างสั่งกระทำการขั้นตอนสำคัญอยู่ ก็อาจร้องขอไม่ให้เข้าสู่สถานะถูกทำลายได้โดยเรียกใช้ `MIDletStateChangeException` อย่างไรก็ตามข้อควรระวังคือ ซอฟต์แวร์จัดการแอปพลิเคชันอาจปฏิเสธหรือยินยอมตามคำร้องขอนี้ก็ได้ ตัวแปรบูลีน `unconditional` ในลายเซ็นของเมธอด `destroyApp()` จะเป็นตัวกำกับว่าการร้องขอนี้สมควรหรือไม่ ถ้าตัวบ่งชี้มีค่าเท่ากับ `true` การร้องขอก็เป็นอันตกไป ตรงกันข้าม หากมีค่า `false` ก็จะได้รับการตอบรับและจะเรียกเมธอด `destroyApp()` ขึ้นมาใหม่ในคราวต่อไป

หาก MIDlet ต้องการเข้าสู่สถานะหยุดชั่วคราวหรือถูกทำลายด้วยตนเอง ก็สามารถทำได้โดยเรียกเมธอด `notifyPause()` หรือ `notifyDestroy()` ตามลำดับ โดยเมธอดทั้ง 2 จะแจ้งให้ซอฟต์แวร์จัดการแอปพลิเคชันทราบว่า MIDlet ได้เข้าสู่สถานะหยุดชั่วคราว / ถูกทำลายแล้ว ในกรณีนี้ซอฟต์แวร์จัดการแอปพลิเคชันจะไม่เรียกเมธอด `pauseApp()` หรือ `destroyApp()` อีก ทั้งนี้ MIDlet ควรจะล้างการใช้ทรัพยากรก่อนส่งสัญญาณให้ซอฟต์แวร์จัดการแอปพลิเคชันทราบ

MIDlet จะเข้าสู่สถานะหยุดทำงาน ได้ก็ต่อเมื่อผ่านสถานะกำลังทำงานมาก่อนแล้ว ในทางตรงกันข้าม หากต้องการเข้าสู่สถานะถูกทำลาย สามารถเข้าได้โดยตรงจากทั้งสถานะหยุดทำงานชั่วคราว และสถานะกำลังทำงาน นอกจากนี้ยังสามารถเข้าสู่สถานะหยุดการทำงาน ได้ในขณะที่กำลังทำงาน หรือเมื่อได้รับคำสั่งจากซอฟต์แวร์จัดการแอปพลิเคชัน

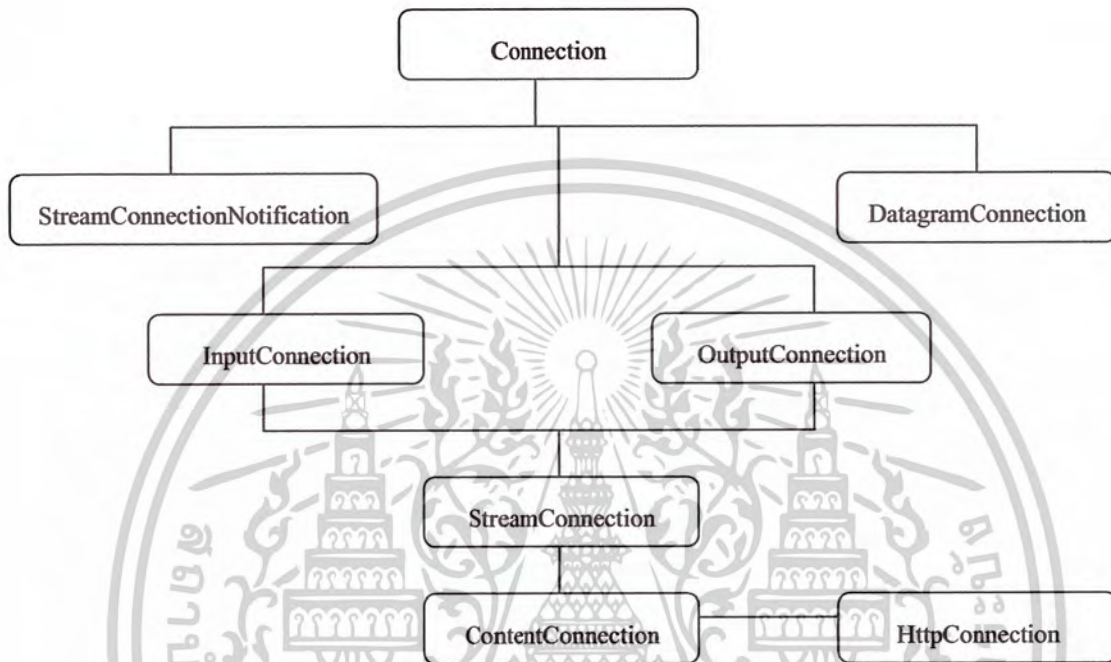
2.2.5 การเชื่อมต่อ J2ME กับเครือข่าย

การเชื่อมต่อเครือข่ายของ J2ME จะต้องมีความยืดหยุ่นสูงเพื่อรองรับอุปกรณ์หลากหลายประเภท ขณะเดียวกันก็ต้องสนับสนุนเฉพาะอุปกรณ์ด้วย โดย J2ME ได้ดึงเอาบางส่วนของแพ็คเกจ `java.io` และแพ็คเกจ `java.network` มาใช้และได้กำหนดขอบเขตการเชื่อมต่อกับระบบเครือข่ายขึ้นมาใหม่โดยเรียกว่า `Generic Connection Framework (GCF)`

Generic Connection Framework

`Generic Connection Framework (GCF)` คือโครงสร้างทั่วไปของการเชื่อมต่อเครือข่ายที่มีความยืดหยุ่นต่อความหลากหลายในรูปแบบการเชื่อมต่อ GCF ประกอบด้วย 1 คลาส คือ คลาส `Connector` และ 7 อินเตอร์เฟส นอกจากนี้ MIDP ยังได้กำหนดอินเตอร์เฟสเพิ่มอีก 1 อินเตอร์เฟส คือ `HttpConnection` ซึ่งทั้งหมดนี้ถูกบรรจุอยู่ในแพ็คเกจ `javax.microedition.io`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-14 อินเทอร์เน็ตใน GCF

อินเทอร์เน็ตแต่ละตัวมีหน้าที่ต่างกันดังนี้

- Connection การเชื่อมต่อแบบพื้นฐาน เช่น เปิดหรือปิดการเชื่อมต่อ
- InputConnection การเชื่อมต่อเพื่ออ่านข้อมูล
- OutputConnection การเชื่อมต่อเพื่อเขียนข้อมูล
- StreamConnection รวมการเชื่อมต่อทั้งเพื่ออ่านข้อมูลและเขียนข้อมูล ใช้ในการเชื่อมต่อแบบ Socket
- ContentConnection สืบทอดจากอินเทอร์เน็ต StreamConnection ใช้จัดการกับข้อมูลผ่านการเชื่อมต่อแบบ HTTP
- StreamConnection Notification รอให้การเชื่อมต่อกับเครือข่ายสำเร็จและคืนออบเจกต์ของอินเทอร์เน็ต StreamConnection
- DatagramConnection การเชื่อมต่อแบบ Datagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบคำสั่งที่ใช้สร้างการเชื่อมต่อ

การสร้างการเชื่อมต่อเครือข่ายทุกรูปแบบเริ่มต้นจากเมธอดในคลาส Connection

```
Connection.open(String connection_string)
```

Connection_string มีรูปแบบการคล้ายคลึงกับรูปแบบของ URL ที่ใช้อยู่ทั่วไปคือ

{ โพรโทคอล } :: { ปลายทาง } ([{ พารามิเตอร์ }])

- โพรโทคอลคือ รูปแบบการเชื่อมต่อซึ่งสามารถกำหนดได้หลายแบบดังนี้

ค่า	ประเภทการเชื่อมต่อ
file	ไฟล์ IO
comm	การเชื่อมต่อผ่านพอร์ตอนุกรม
socket	การเชื่อมต่อแบบ Socket
datagram	การเชื่อมต่อกับเซิร์ฟเวอร์
http	การเชื่อมต่อกับเซิร์ฟเวอร์

- ปลายทาง อาจจะเป็นชื่อของ host หรือชื่อไฟล์ก็ได้ เป็นต้น
- พารามิเตอร์ คือ ข้อมูลเพิ่มเติมในการเชื่อมต่อ เป็นตัวเลือกที่ไม่ต้องใส่ก็ได้

ตัวอย่างด้านล่างแสดงการใช้เมธอด open() เพื่อสร้างการเชื่อมต่อแบบต่างๆ โดยการกำหนดโปรโตคอลที่ต่างกันออกไปดังนี้

```
//การเชื่อมต่อแบบ http
Connection.open("http://www.monster.com");

//การเชื่อมต่อแบบ sockets
Connection.open("socket://localhost:80");

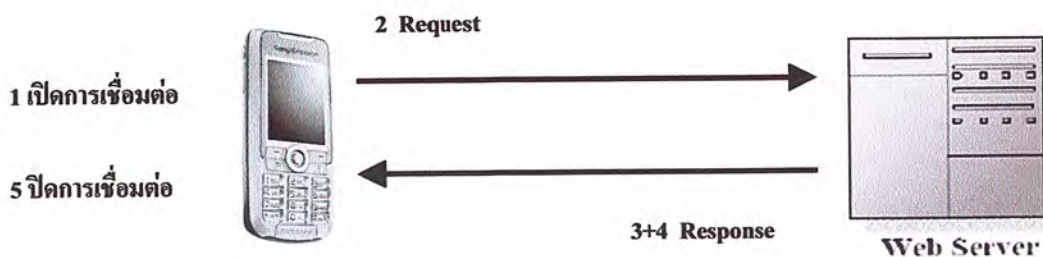
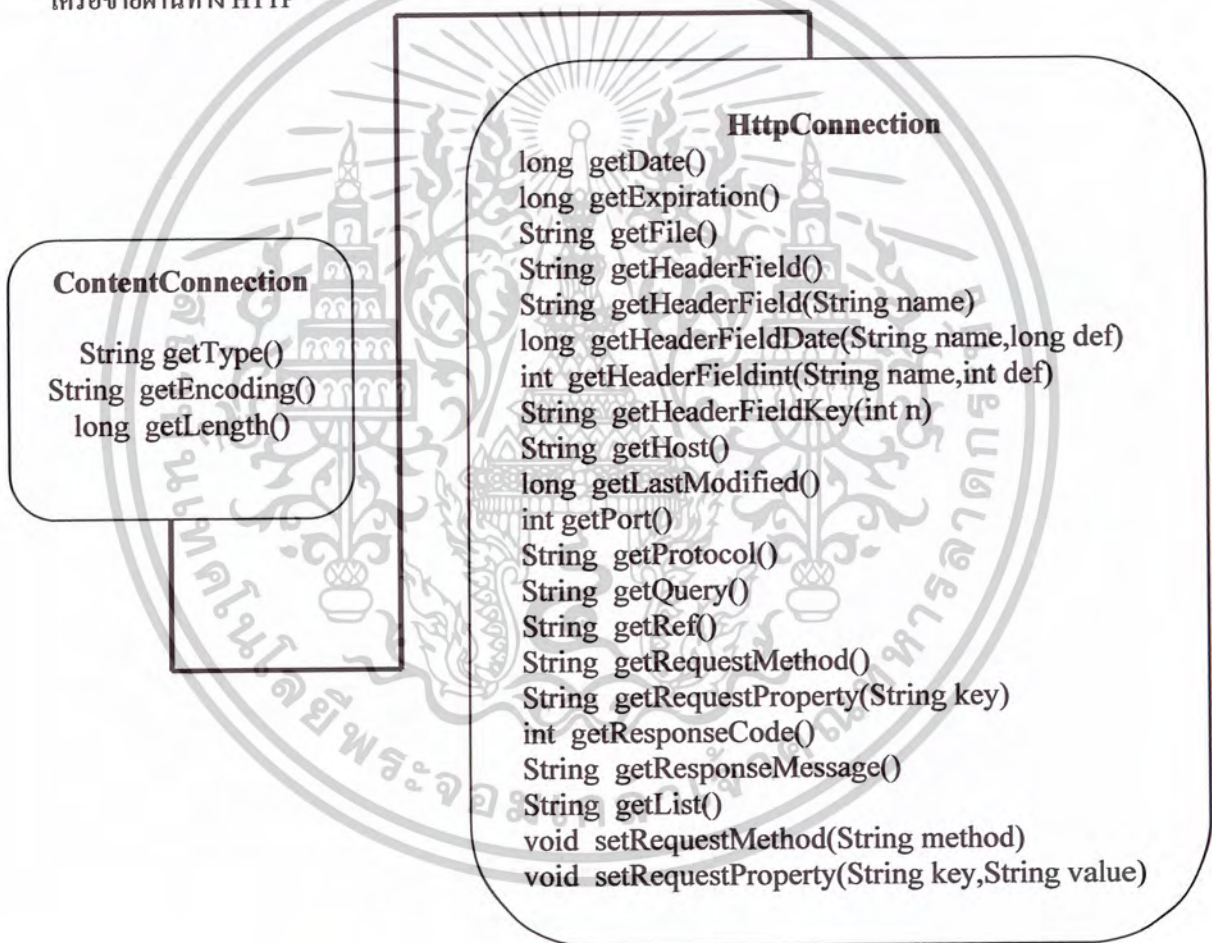
//การเชื่อมต่อแบบ Datagrams
Connection.open("datagram://http://www.monster.com:9000");

//การเชื่อมต่อกับ File:
Connection.open("file://foo.dat");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการเชื่อมต่อเครือข่ายแบบ HTTP

การเชื่อมต่อเครือข่ายแบบ HTTP ใช้ในการเชื่อมต่อระหว่างโทรศัพท์มือถือและเว็บเซิร์ฟเวอร์ โดยมีการกำหนดให้อินเตอร์เฟส `HttpConnection` เป็นมาตรฐานของอุปกรณ์ประเภท MIDP เพื่อรองรับการเชื่อมต่อเครือข่ายผ่านทาง HTTP



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อแบบ HTTP

1. เปิดการเชื่อมต่อกับเว็บเซิร์ฟเวอร์ (สถานะ HttpConnection เป็น Setup)
2. โทรศัพท์มือถือส่งการร้องขอ (Request) ไปที่เว็บเซิร์ฟเวอร์โดยใช้วิธี GET, HEAD หรือ POST (สถานะของ HttpConnection เป็น Connected)

วิธีการร้องขอ	รายละเอียด
GET	ใช้เรียกข้อมูลจากเว็บเซิร์ฟเวอร์
POST	ใช้ส่งข้อมูลกลับไปยังเว็บเซิร์ฟเวอร์
HEAD	ใช้เรียกดูเฉพาะส่วนหัว (header) ของข้อมูลจากเว็บเซิร์ฟเวอร์

3. เว็บเซิร์ฟเวอร์ตอบรับการร้องขอ (สถานะของ Httpconnection เป็น Connected)
4. เว็บเซิร์ฟเวอร์จัดการกับคำร้องขอและส่งผลการร้องขอกลับไปที่โทรศัพท์มือถือ (สถานะของ HttpConnection เป็น Connected)
5. ปิดการเชื่อมต่อ (สถานะของ HttpConnection เป็น Closed)

2.3 เทคโนโลยีอื่นๆ

2.3.1 MySQL

มายเอสคิวแอล(MySQL) จัดเป็นระบบฐานข้อมูลเชิงสัมพันธ์ (RDBMS: Relational Database Management System) ตัวหนึ่ง ซึ่งเป็นที่นิยมใช้กันมากในปัจจุบันตัวหนึ่ง สาเหตุก็เพราะ มายเอสคิวแอล เป็นฟรีแวร์ทางด้านฐานข้อมูลที่มีประสิทธิภาพสูงและมีความสามารถในการด้านของความเร็ว การรองรับจำนวนผู้ใช้ และขนาดของข้อมูลจำนวนมาก ทั้งยังสนับสนุนการทำงานบนระบบปฏิบัติการที่หลากหลาย ไม่ว่าจะเป็น ยูนิกซ์(Unix), โอเอสทู(OS/2), แมค โอเอส(Mac OS), ลินุกซ์(Linux) หรือวินโดวส์(Windows) ก็ตาม นอกจากนี้ มายเอสคิวแอล ยังสามารถใช้งานร่วมกับ เว็บเดเวลลอปเมนต์แพลตฟอร์ม(Web Development Platform) ทั้งหลาย ไม่ว่าจะเป็น ซี(C), ซีพลัสพลัส(C++), จาวา(Java), เพิร์ล(Perl), พีเอชพี(PHP), เอเอสพี(ASP) หรือ เจเอสพี(JSP) จึงเป็นสาเหตุให้ มายเอสคิวแอล ได้รับความนิยมมากในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติเด่นของ มายเอสคิวแอล

- มายเอสคิวแอล เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ซึ่งจะทำการเก็บ ข้อมูลแยกเป็นตารางแทนที่จะเก็บข้อมูลไว้รวม ๆ กันเป็นก้อนใหญ่ไว้ในที่หนึ่ง ซึ่งสิ่งนี้ได้เพิ่ม ความเร็วและความยืดหยุ่นในการใช้งานฐานข้อมูล ตารางเหล่านี้จะเชื่อมกัน โดยการกำหนดความ สัมพันธ์ให้แต่ละตาราง ซึ่งจะทำให้สามารถรวมข้อมูลจากหลาย ๆ ตารางได้
- มายเอสคิวแอล ใช้ภาษา SQL (Structured Query Language) เป็นพื้นฐานในการกระทำต่าง ๆ กับฐานข้อมูล ซึ่งภาษา SQL นี้เป็นภาษามาตรฐานในการติดต่อกับฐานข้อมูลอยู่แล้ว ทำให้ผู้ใช้ สามารถเรียนรู้การใช้งาน มายเอสคิวแอล ได้อย่างง่ายดายและรวดเร็ว

มายเอสคิวแอล เป็นซอร์ซแบบเปิด กล่าวคือ ใคร ๆ ก็ตามต่างก็มีสิทธิ์ใช้ มายเอสคิวแอล ได้โดยไม่ต้องเสีย ค่าใช้จ่ายใด ๆ ทั้งสิ้น อีกทั้งผู้ใช้สามารถเรียนรู้การทำงานของ มายเอสคิวแอล ได้ จากซอร์ซ โค้ด และสามารถทำการแก้ไขซอร์ซโค้ดนั้นเพื่อให้ มายเอสคิวแอล มีความเหมาะสมกับความต้องการ ของตนได้

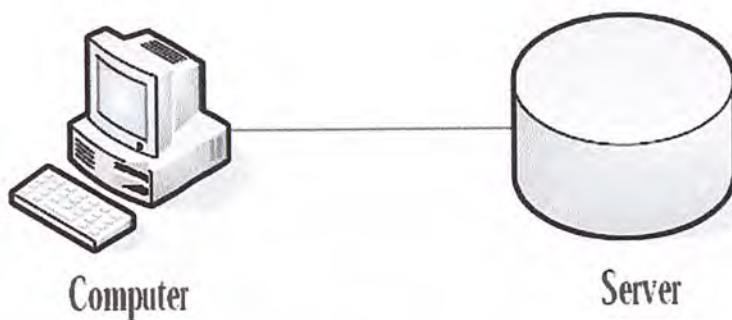
- มายเอสคิวแอล มีความเร็วสูง ในการเข้าถึงฐานข้อมูล
- สามารถใช้ มายเอสคิวแอล ได้ในหลาย ๆ ระบบปฏิบัติการ เช่น ลินุกซ์, ยูนิกซ์, วินโดวส์
- มายเอสคิวแอล ง่ายต่อการเรียนรู้และใช้งาน

สถาปัตยกรรมของมายเอสคิวแอล

สถาปัตยกรรม หรือ โครงสร้างภายในของ มายเอสคิวแอล ก็คือ การออกแบบการทำงานในลักษณะของ ไคลเอนต์/เซิร์ฟเวอร์(Client/Server) ซึ่งประกอบด้วยส่วนหลักๆ 2 ส่วนคือ ส่วนของผู้ให้บริการ (เซิร์ฟเวอร์) และ ส่วนผู้ใช้บริการ(ไคลเอนต์) โดยแต่ละส่วนก็จะมีโปรแกรมการทำงานตามหน้าที่ของตน ส่วนผู้ให้บริการ หรือ เซิร์ฟเวอร์ จะเป็นส่วนที่ทำหน้าที่บริหารจัดการระบบฐานข้อมูลในที่นี้ก็คือ ตัว มายเอสคิวแอลเซิร์ฟเวอร์ นั่นเองและเป็นที่ยึดเก็บข้อมูลทั้งหมด ข้อมูลที่เก็บไว้มีทั้งข้อมูลที่ใช้ในการทำงานกับระบบฐานข้อมูล และข้อมูลที่เกิดจากที่ผู้ใช้แต่ละคนสร้างขึ้นมา ส่วนผู้ให้บริการ หรือ ไคลเอนต์ ก็คือผู้ใช้นั่นเอง โดยโปรแกรม สำหรับใช้งานในส่วนนี้ได้แก่ มายเอสคิวแอล ไคลเอนต์, แอคเซส (Access), Web development Platform ต่าง เช่น Java, Perl, PHP, ASP, JSP Computer Server

รูปภาพ 5-1 สถาปัตยกรรมของ MySQL 5.1.4.3 หลักการทำงานของ มายเอสคิวแอล 1. ที่ฝั่งของ เซิร์ฟเวอร์ จะมีโปรแกรมหรือระบบสำหรับจัดการฐานข้อมูลทำงานรออยู่ เพื่อเตรียมหรือรอคอยการร้องขอการใช้บริการ จาก ไคลเอนต์ 2. เมื่อมีการร้องขอการใช้บริการเข้ามา เซิร์ฟเวอร์ จะทำการตรวจสอบตามวิธีของตน เช่น อาจจะมีการให้ผู้ให้บริการระบุชื่อและรหัสผ่าน และสำหรับ มายเอสคิวแอล สามารถกำหนดได้ว่าจะอนุญาตหรือ ปฏิเสธ ไคลเอนต์ ใดๆ ในระบบที่เข้าใช้บริการอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-7 สถาปัตยกรรมของ MySQL

หลักการทำงานของ มายเอสคิวแอล

1.ที่ฝั่งของ เซิร์ฟเวอร์ จะมีโปรแกรมหรือระบบสำหรับจัดการฐานข้อมูลทำงานรออยู่ เพื่อเตรียมหรือรอคอยการร้องขอการใช้บริการจาก ไคลเอนต์

2.เมื่อมีการร้องขอการใช้บริการเข้ามา เซิร์ฟเวอร์ จะทำการตรวจสอบตามวิธีของตน เช่น อาจจะมีการให้ผู้ใช้บริการระบุชื่อและรหัสผ่าน และสำหรับ มายเอสคิวแอล สามารถกำหนดได้ว่าจะอนุญาตหรือปฏิเสธ ไคลเอนต์ ใดๆ ในระบบที่เข้าใช้บริการอีกด้วย

3.ถ้าผ่านการตรวจสอบ เซิร์ฟเวอร์ ก็จะอนุมัติการ ให้บริการแก่ ไคลเอนต์ ที่ร้องขอใช้บริการนั้นต่อไป และถ้าในกรณีที่ ไม่ได้รับการอนุมัติเซิร์ฟเวอร์ก็จะส่งข่าวสารความผิดพลาดแจ้งกลับไปให้ ไคลเอนต์ ที่ร้องขอ บริการนั้น เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น ไคลเอนต์ หรือ เซิร์ฟเวอร์ อาจอยู่บนเครื่องเดียวกัน หรือแยกเครื่องกันก็ได้ ทั้งนี้ขึ้นอยู่กับลักษณะการทำงาน หรือการกำหนดของผู้บริหารระบบ ตามปกติถ้าเป็นการทำงานในลักษณะ เว็บพื้นฐาน(Web-based) มีการใช้ฐานข้อมูลขนาดไม่ใหญ่นัก ตัว มายเอสคิวแอล เซิร์ฟเวอร์ และ ไคลเอนต์ มักทำงานอยู่บนเครื่องเดียวกัน โดยเครื่องคอมพิวเตอร์ดังกล่าวจะต้องมีทรัพยากรเพื่อการทำงาน (เช่น เนื้อที่ฮาร์ดดิสก์, หน่วยความจำหลัก เป็นต้น) มากพอสมควร แต่สำหรับการทำงานจริง (Real-world Application) ก็มักจะแยก ไคลเอนต์ และ เซิร์ฟเวอร์ ออกเป็นคนละเครื่องกัน เพราะสามารถรองรับการทำงาน ได้ดีกว่ามากกว่าดั่งนั้น ผู้บริหารระบบ หรือผู้กำหนดนโยบายสำหรับทำงานเครือข่าย จะต้องคำนึงถึงเรื่องที่เกี่ยวข้องเหล่านี้ให้ดี เพื่อที่จะให้ระบบมีการทำงานรองรับการ ให้บริการผู้ใช้อย่างมีประสิทธิภาพและข้อมูลมีความปลอดภัยมากที่สุด

วิธีการเชื่อมต่อจาก ไคลเอนต์ เข้าสู่ เซิร์ฟเวอร์

จะกล่าวถึงลักษณะวิธีการเชื่อมต่อ 2 แบบ คือ แบบเนทีฟ(Native) และแบบผ่านตัวกลาง ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.แบบ Native เป็นที่นิยมกันมากในกรณีที่ ระบบปฏิบัติการของ มายเอสคิวแอล เซิร์ฟเวอร์ เป็น ยูนิคต์ เป็นวิธีการเชื่อมต่อที่มีวิธีการทำงานเร็วที่สุด เพราะทำงานกันภายใน โดยลักษณะการทำงานประเภทนี้ได้แก่ การใช้งาน มายเอสคิวแอล ร่วมกับ Web Development Platform ทั้งหมด ที่ต้องมีการ Modify ภายใน เช่น PHP

2.แบบผ่านตัวกลาง ในที่นี้จะกล่าวถึงรูปแบบที่นิยมใช้งานมากที่สุดก็คือ ODBC (Open DataBase Connectivity) ซึ่งส่วนใหญ่ใช้กับ เซิร์ฟเวอร์ ที่เป็น Windows Platform เป็นระบบปฏิบัติการ การทำงาน ประเภทนี้อาจมีการทำงานที่ช้ากว่าแบบ Native เพราะการทำงานในแต่ละครั้งระหว่าง ไคลเอนต์ และ เซิร์ฟเวอร์ ต้องผ่านตัวกลางก่อน แต่ ODBC ก็ถือว่ามีข้อได้เปรียบในเรื่องฐานผู้ใช้ Windows Platform มากกว่า และด้วย ODBC ทำให้สามารถใช้เครื่องมือไคลเอนต์ เคเวลอปเม้นท์ (Client Development Tools) ขอดนิชมได้มากกว่า เช่น Access, VB, ASP, JSP เพื่อเชื่อมต่อเข้าหา มายเอสคิวแอล เซิร์ฟเวอร์ ได้

เปรียบเทียบการทำงานระหว่างแบบเนทีฟ และแบบผ่านตัวกลาง จะเห็นได้ว่าแบบผ่านตัวกลางจะใช้ขั้นตอนมากกว่าแบบ เนทีฟ ถึง 2 เท่า แต่มีข้อดีข้อด้อยแตกต่างกันสรุปได้ดังนี้

แบบ Native	แบบผ่านตัวกลาง
1. มีการทำงานรวดเร็วกว่า เพราะสื่อสารกันภายใน	1. มีการทำงานช้ากว่าเพราะมีการทำงานเพิ่มขึ้นอีก 1 ขั้นตอน
2. ไคลเอนต์ ที่จะมาเชื่อมต่อต้องมีการฝังส่วนของ โปรแกรมของ เซิร์ฟเวอร์ บางส่วนไว้สำหรับการใช้งาน หมายถึงเราต้องปรับปรุง ไคลเอนต์ เพิ่มเติม	2. ไม่ต้องปรับปรุง ไคลเอนต์ เพิ่มเติม เพียงแต่สนับสนุน ODBC ก็สมารถทำงานได้แล้ว
3. ส่วนใหญ่มักไม่มีข้อจำกัดในการใช้งาน	3. มีข้อจำกัดขึ้นอยู่กับตัวกลางหรือ ODBC ที่ใช้
4. Server Platform ส่วนใหญ่มักเป็น ยูนิคต์	4. ส่วนใหญ่เป็น Windows Platform
5. ไคลเอนต์ ส่วนใหญ่ใช้งานในลักษณะ Web-based เช่น Java Perl PHP เป็นต้น	5. รองรับทั้ง Web-based หรือ Client Development Tools อื่นๆ เช่น Access, VB, ASP, JSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง เปรียบเทียบการทำงานระหว่างแบบเนทีฟ และแบบผ่านตัวกลาง

ความสามารถของ มายเอสคิวแอล

ความสามารถของ มายเอสคิวแอล โดยทั่วไปจะครอบคลุมความต้องการของผู้ใช้เพียงพอ แต่ถ้านำไปเปรียบเทียบกับฐานข้อมูลที่ขายกันในท้องตลาดแล้ว (Commercial) อาจแตกต่างกันมากปกติผลิตภัณฑ์ที่เป็น Commercial เหล่านั้น มักจะมีความสามารถต่างๆ ที่เกินความจำเป็นของผู้ใช้อยู่เสมอ สิ่งที่เกิดขึ้นเหล่านี้ถือเป็นการสูญเสียของผลิตภัณฑ์ เพราะทำขึ้นมาแต่ไม่ได้ถูกนำไปใช้งาน หรือใช้แต่ไม่เต็มความสามารถ นอกจากนี้ก็อาจทำให้ผลิตภัณฑ์มีขนาดใหญ่ขึ้น อีกทั้งราคาก็สูงตามไปด้วย ซึ่งสำหรับ มายเอสคิวแอล แล้ว จะมีความสามารถที่ครอบคลุมความต้องการของผู้ใช้ ไม่มีอะไรที่เกินความจำเป็น ทั้งนี้อาจสรุปความสามารถเด่นๆ ได้ดังนี้

- มายเอสคิวแอล จัดเป็นระบบฐานข้อมูลประเภท SQL-based ผู้ใช้หรือผู้พัฒนาสามารถใช้คำสั่ง SQL ในการสั่ง หรือใช้งานกับ มายเอสคิวแอล เซิร์ฟเวอร์ ได้โดยไม่ต้องศึกษาเพิ่มเติมแต่อย่างใด ซึ่งความสามารถนี้ถือเป็นแนวโน้มในการจัดการฐานข้อมูลในปัจจุบัน
- สนับสนุนการใช้งานสำหรับตัวประมวลผลกลาง (CPU: Central Processing Unit) หลายตัว
- การทำงานแบบมัลติเทร็ด (Multi-threaded) ใช้ Kernel Threads
- สนับสนุน API เพื่อใช้งานกับ Development Platform ต่างๆ มากมาย ไม่ว่าจะเป็น C, C++, Eiffel, Java, Perl, PHP, Python, หรือ Tcl และนอกจากนี้ยังสามารถใช้งานได้กับ ODBC (Open DataBase Connectivity) ซึ่งทำให้สามารถใช้งานกับเครื่องมืออื่นบน Windows Platform ได้เช่น Access เป็นต้น
- มายเอสคิวแอล สามารถทำงานได้บนระบบปฏิบัติการหลายตัวหลายค่าย ไม่ว่าจะเป็น AIX, BSD/OS, DEC Unix, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2, SGI Irix, Solaris, SunOS, SCO OpenServer, SCO Unixware, Tru64 Unix, Windows Platform ทำให้ผู้ใช้สามารถทำการย้ายหรือปรับขนาดของระบบขึ้นไปได้ในกรณีที่ต้องการขยายขนาดของข้อมูล หรือมีความต้องการทรัพยากรเพิ่มมากขึ้น
- ประเภทของข้อมูลที่ใช้ได้ ใน มายเอสคิวแอล ได้แก่ ตัวเลข(ทั้งแบบคิดและไม่คิดเครื่องหมาย) ขนาด 1, 2, 3, 4 และ 8 ไบท์ FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET และ ENUM
- สนับสนุน GROUP BY และ ORDER BY clauses และ Group Functions ได้แก่ COUNT(), COUNT(DISTINCT), AVG(), STD(), SUM(), MAX(), และ MIN() - สนับสนุน Left OUTER JOIN และ Right OUTER JOIN - การกำหนดสิทธิและรหัสผ่าน ให้มีความปลอดภัย ความยืดหยุ่นสูง สามารถกำหนดเครื่องและ/หรือผู้ใช้ ในการเข้าถึงข้อมูลได้ มีการเข้ารหัสข้อมูล (Encryption) สำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสผ่านของผู้ใช้ด้วย ทำให้ผู้ใช้นั้นเฝ้าข้อมูลจะมีความปลอดภัย ไม่มีใครทำการเข้าถึงข้อมูลได้ หากไม่ได้รับอนุญาต

- สามารถทำดัชนี(Index) ได้สูงสุดถึง 32 ดัชนีในแต่ละตารางข้อมูล โดยที่ในแต่ละดัชนีสามารถใช้ฟิลด์ได้ตั้งแต่ 1-16 ฟิลด์
- สามารถรองรับข้อมูลขนาดใหญ่ เช่น ข้อมูลระดับล้านระเบียน ซึ่งในปัจจุบัน มายเอสคิวแอล สามารถรองรับข้อมูลได้ในระดับ 60,000 ตารางข้อมูล และ 5 ล้านระเบียน
- สนับสนุนรูปแบบของภาษา (Character Set) หลากชนิด เช่น ISO-8859-1 (latin1), big5, ujis และอื่นๆ ทำให้สามารถทำการจัดเรียงข้อมูล (Sort) หรือกำหนดการแสดงผลข้อผิดพลาด (Error Message) ได้ตามรูปแบบภาษาที่ต้องการ
- เครื่องที่ทำหน้าที่เป็นผู้ให้บริการ สามารถเชื่อมเข้าสู่ มายเอสคิวแอล เซิร์ฟเวอร์ โดยการใช้ที่ซีพี/ไอที ซ็อกเก็ต(TCP/IP Socket), ยูนิกซ์ ซ็อกเก็ต (Unix Socket) หรือ เนมไพป์(Name Pipes(NT))

2.3.2 JDBC

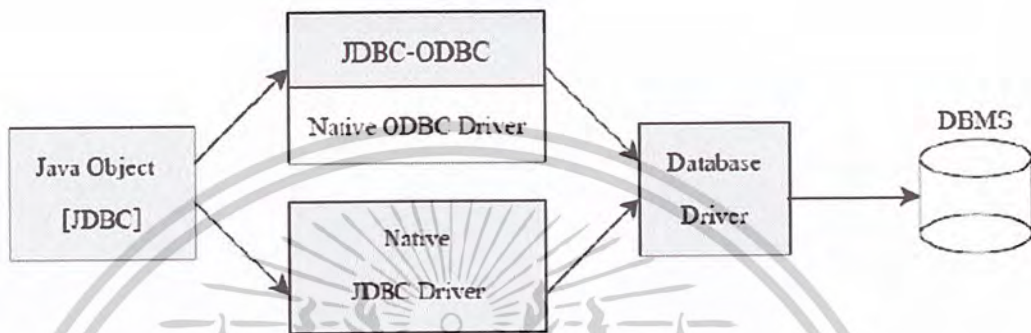
JDBC (Java Database Connectivity) ถูกพัฒนาโดย JavaSoft Department ของบริษัท Sun Microsystems ซึ่งก็คือฟังก์ชันมาตรฐานหรือ Java Application Programming Interface (API) สำหรับการเชื่อมต่อกับระบบฐานข้อมูล, นักพัฒนาสามารถใช้ JDBC API และยังประกอบด้วย packages อื่นๆด้วย ซึ่งนำเสนอในรูปแบบฟังก์ชันพิเศษหรือ higher level API เพื่อเขียน โปรแกรมอิสระในการกำหนด DBMS หรือ database connectivity mechanism ในการเริ่มใช้งาน, โดยทั่วไปการใช้ SQL database ในการติดต่อ (interface) กับ framework เพื่อที่จะจัดมาตรฐานในการติดต่อในส่วนบนสุดของชนิดต่างๆ ของ database connectivity modules ซึ่งก็คือมาตรฐานของ ANSI SQL-2 Entry level database เพราะว่า Relational databases เกือบจะทั้งหมดในปัจจุบันใช้มาตรฐานของ SQL-2 Entry level JDBC สร้างระดับการเชื่อมต่อเพื่อสื่อสารกับฐานข้อมูลในรูปแบบที่คล้ายคลึงกับ ODBC Database Connectivity ของบริษัทไมโครซอฟต์ ซึ่งในปัจจุบันได้ถือว่าเป็นมาตรฐานของการเชื่อมต่อกับฐานข้อมูลสำหรับเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) และระบบเครือข่ายท้องถิ่น (LAN) แต่ตามหลักการทำงานของทั้ง JDBC และ ODBC ตั้งอยู่บนมาตรฐานเดียวกันคือ X/Open SQL Call-level Interface ของระบบ X-Windows และ JDBC driver ต้องเข้ากันได้กับระดับมาตรฐานในการเข้าถึง SQL (ANSI SQL Entry Level Standard) และต้องผ่าน Conformance test ซึ่ง JavaSoft เป็นผู้กำหนดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการการทำงานของ JDBC

มีหน้าที่หลัก 3 อย่าง คือ

1. สร้างจุดเชื่อมต่อกับฐานข้อมูลที่ต้องการเชื่อมต่อด้วย
2. ส่งคำสั่ง SQL (SQL Statement) ไปยังฐานข้อมูลนั้น
3. รับผลจากคำสั่งที่ส่งไปในข้อ 2



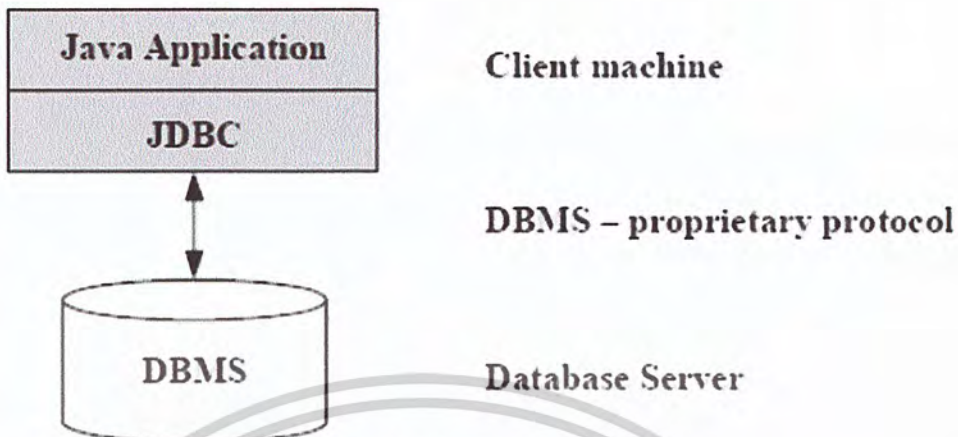
รูปที่ 2-8 การทำงานของ JDBC

จากภาพ จาวาออบเจกต์ (Java Object) จะร้องขอ ไปยัง JDBC Class ซึ่งจะเป็นตัวเรียก ไปยัง JDBC Driver หรือ JDBC-ODBC Bridge ที่กำหนด จากนั้นเน็ตเวิร์กไดรเวอร์ (Native Driver) จะเรียกตัวไดรเวอร์ของฐานข้อมูลจริง ๆ เพื่อเชื่อมต่อ ไปยังฐานข้อมูลอย่างสมบูรณ์

รูปแบบการเชื่อมต่อฐานข้อมูลของ JDBC

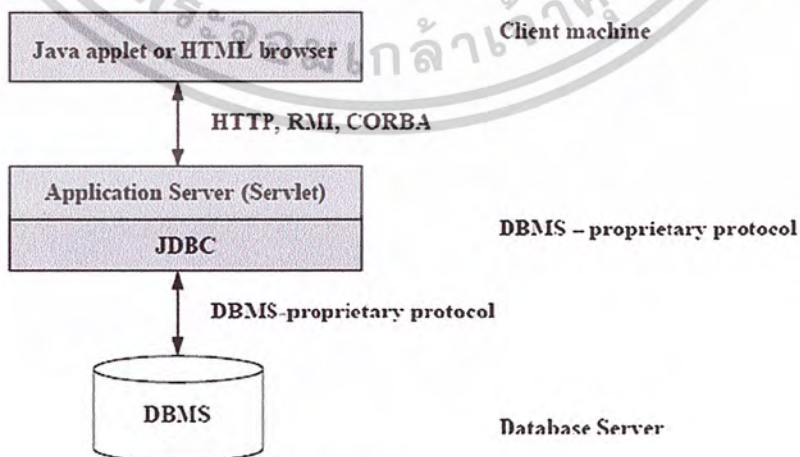
JDBC API สนับสนุนรูปแบบการเชื่อมต่อฐานข้อมูลทั้งแบบ Two-Tiers model และ Three-Tiers model • Two – Tiers model จาวาแอปเพล็ต (Java applet) หรือจาวาแอปพลิเคชัน (Application) จะติดต่อกับฐานข้อมูลโดยตรง จึงมีความจำเป็นที่โปรแกรมจาวาต้องการ JDBC ไดรเวอร์พิเศษที่สามารถสื่อสารกับระบบจัดการฐานข้อมูลชนิดนั้นได้ รูปแบบการเชื่อมต่อแสดงดังรูปที่ คำสั่งในการค้นคืนข้อมูลในรูปของภาษา SQL (Structured Query Language) จะถูกส่งจากผู้ไปสู่ฐานข้อมูล หลังจากนั้นผลจากการประมวลผลของระบบจัดการฐานข้อมูล ก็จะถูกส่งกลับมาสู่ผู้ใช้งานข้อมูลดังกล่าวนี้ส่วนมากจะติด ตั้งอยู่ต่างเครื่องคอมพิวเตอร์ โดยผู้ใช้งานสามารถเชื่อมต่อผ่านระบบเน็ตเวิร์ก (Network) รูปแบบ Two – Tiers นี้ใช้หลักการการทำงานเช่นเดียวกับรูปแบบไคลเอนต์/เซิร์ฟเวอร์ที่เรารู้จักกันดีโดยที่เครื่องคอมพิวเตอร์ของผู้ใช้ คือ ไคลเอนต์และเครื่องคอมพิวเตอร์ ที่ให้บริการฐานข้อมูลคือเซิร์ฟเวอร์ เครื่องข่ายคอมพิวเตอร์ที่นิยมใช้รูปแบบ Two-Tiers มักเป็นเครือข่ายอินทราเน็ต (Intranet) สำหรับดำเนินธุรกรรมภายในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-9 Two - Tiers Model

- Three - Tiers Model คำสั่งคั่นต่างๆจากผู้ใช้งานจะถูกส่งไปให้กับ Middle Tier หรือส่วนกลางของการบริการเสียก่อน หลังจากนั้น Middle Tier จะแปลงคำสั่งเหล่านี้ให้เป็นภาษา SQL เพื่อส่งไปให้กับระบบจัดการฐานข้อมูลเพื่อทำการประมวลผล ข้อมูลผลลัพธ์ที่ได้ก็จะถูกส่งกลับคืนไปให้กับ Middle Tier และส่งต่อไปให้ผู้ใช้งานในที่สุด หลักการทำงานเช่นนี้มักจะพบในเครือข่ายอินเทอร์เน็ตซึ่งประกอบด้วยเครื่องคอมพิวเตอร์หลากหลายชนิด และเว็บเซิร์ฟเวอร์ ซึ่งทำหน้าที่เป็น Middle Tier ก็จะเป็นตัวกลางในการจัดการให้คอมพิวเตอร์ทั้งไคลเอ็นต์และเซิร์ฟเวอร์ฐานข้อมูลสามารถพูดคุยกันได้ การปรับเปลี่ยนระบบคอมพิวเตอร์ของไคลเอ็นต์และเซิร์ฟเวอร์ หรือแม้กระทั่งการเปลี่ยนฐานข้อมูลตัวใหม่จะไม่มีผลกระทบต่อซึ่งกันและกันเกิดขึ้น

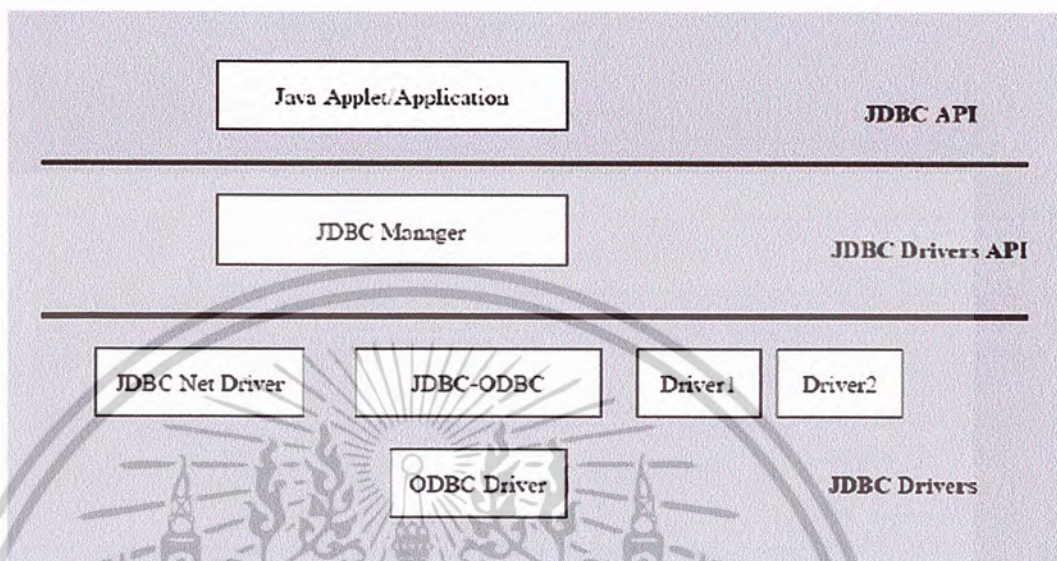


รูปที่ 2 -10 Three - Tiers Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ JDBC

โครงสร้างการเชื่อมต่อภายใน JDBC ประกอบด้วย 3 ระดับหลักคือ JDBC API, JDBC Driver API และ JDBC Driver



รูปที่ 2-11 ระดับการเชื่อมต่อของ JDBC API

JDBC API ประกอบด้วยส่วนติดต่อ (Interface) 2 ส่วนที่สำคัญ ส่วนแรกคือ JDBC API เป็นระดับของฟังก์ชัน API ที่อำนวยความสะดวกให้แก่ผู้เขียน โปรแกรมประยุกต์(application writers) ส่วนที่สองคือ JDBC driver API ในระดับ lower-level สำหรับผู้เขียน ไดรเวอร์ (driver writers)

รูปแบบของ JDBC ไดรเวอร์ (JDBC drivers)

JDBC ไดรเวอร์หรือส่วนที่อยู่เบื้องหลังการทำงานของฟังก์ชัน API ต่างๆของ JDBC โดยเทคโนโลยี JDBC drivers ถูกแบ่งออกเป็น 4 ประเภทด้วยกันคือ

ประเภทที่ 1: JDBC-ODBC Bridge plus ODBC Driver

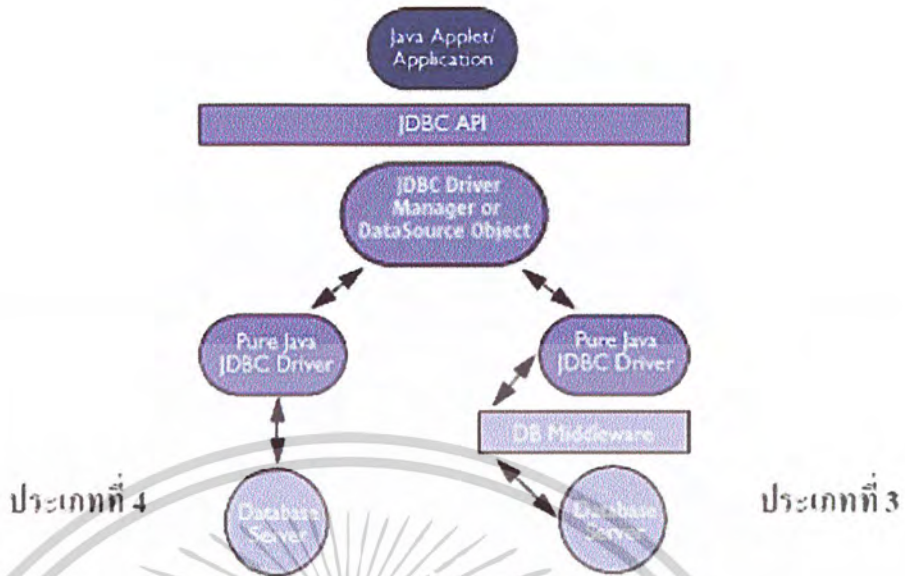
ประเภทที่ 2: Native-API (partly Java) Driver

ประเภทที่ 3: Pure Java Driver สำหรับ Database Middleware

ประเภทที่ 4 : Direct-to-Database Pure Java Driver แอปพลิเคชันและแอปเพล็ตสามารถเข้าถึง

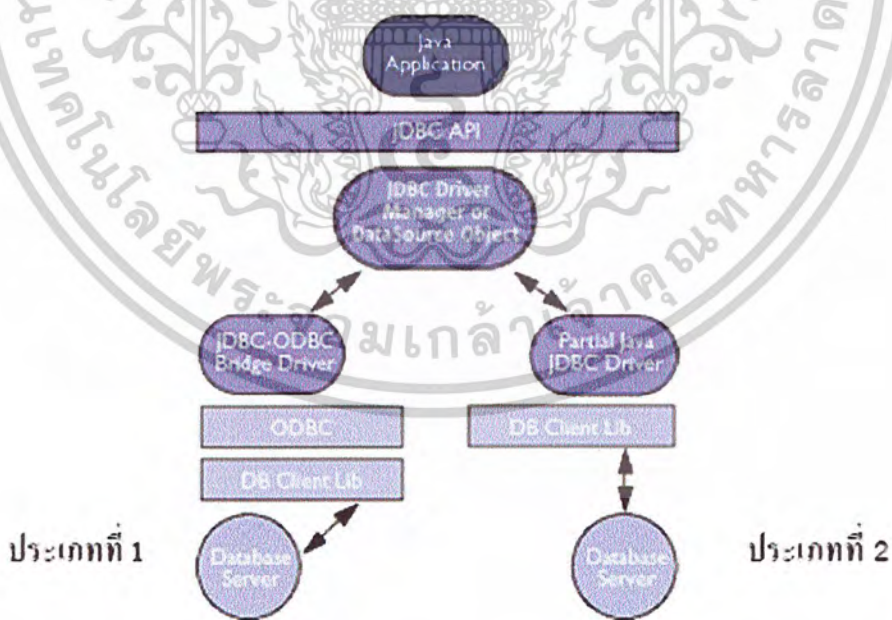
ฐาน ข้อมูลผ่าน JDBC API โดยใช้เทคโนโลยี Pure Java JDBC driver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-12 JDBC driver ประเภทที่ 3 และ 4 ที่ใช้ pure Java JDBC driver

JDBC Connectivity โดยใช้ ODBC ไดรเวอร์ (ODBC drivers) และ libraries ของฐานข้อมูลในเครื่องไคลเ็นต์ที่มีอยู่แล้ว



รูปที่ 2-13 JDBC driver ประเภทที่ 1 และ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภท 1 (Type1: JDBC-ODBC Bridge)

มักเรียกกันสั้นๆว่า JDBC-ODBC บริดจ์ หมายถึง JDBC ที่ทำงานอยู่บนชั้นการสื่อสารมิดเดิลแวร์ที่ชื่อ ODBC อีกทีหนึ่ง ถูกใช้สำหรับการศึกษาวิธีการเขียนโปรแกรมจาวา เพื่อติดต่อกับ JDBC เหมาะกับระบบงานที่มี ODBC ในการทำงานอยู่แล้ว

- สำหรับระบบงานที่ทั้งหมดทำงานอยู่บนพื้นฐานของแพลตฟอร์มวินโดวส์
- ไม่เหมาะกับการนำไปใช้งานบนระบบงานที่มีขนาดใหญ่ๆเนื่องจากทำให้เกิดความช้าในการทำงานและประสิทธิภาพในการทำงานที่ไม่ดี
- มีข้อมูลในส่วนโอเวอร์เฮด(Overhead) สูง เนื่องจากต้องมีส่วนในการติดต่อกันระหว่าง JDBC และ ODBC เพิ่มเติม
- ไม่สนับสนุนความสามารถทั้งหมดของมาตรฐาน JDBC เนื่องจากข้อจำกัดของ ODBC ที่มีรูปแบบการทำงานน้อยกว่า JDBC

ประเภท 2 (Type2: Native-API Driver)

สำหรับ JDBC ประเภทนี้ ตัวไดรเวอร์ (มิดเดิลแวร์ของแต่ละผลิตภัณฑ์มักจะใช้คำว่าไดรเวอร์แทนมิดเดิลแวร์ตัวนั้นๆ) จะถูกโหลดไว้ที่ไคลเอนต์หรือ Tier ส่วนติดต่อกับผู้ใช้

- มีประสิทธิภาพในการดีกว่าประเภท 1 เมื่อเปรียบเทียบกัน
- คำสั่งในการติดต่อกับเซิร์ฟเวอร์ จะเป็นคำสั่งที่ดีที่สุดสำหรับเซิร์ฟเวอร์นั้นๆ โดยเฉพาะ ทำการทำงานโดยรวมดีกว่า ข้อดีของการนำไปใช้งาน
- ผู้ใช้โปรแกรมในส่วนไคลเอนต์ยังต้องการ ไดรเวอร์ สำหรับเซิร์ฟเวอร์โดยเฉพาะ
- เมื่อมีการเปลี่ยนแปลงเซิร์ฟเวอร์เป็นผลิตภัณฑ์ตัวอื่น โปรแกรมในส่วนไคลเอนต์ต้องมีการเปลี่ยนแปลงและคอมไพล์ใหม่เสมอ

ประเภท 3 (Type 3: Pure Java Driver)

ต้องมีแอปพลิเคชันเซิร์ฟเวอร์ทำงานร่วมอยู่ในระบบ

- เป็นระบบที่มีประสิทธิภาพในการทำงานที่ดีกว่าประเภทที่ 1 และ 2 เมื่อเปรียบเทียบกัน
- เหมาะสำหรับองค์กรที่มีเซิร์ฟเวอร์ทางด้านฐานข้อมูลที่หลากหลายรูปแบบ
- การทำงานของไคลเอนต์ไม่จำเป็นต้องติดตั้ง JDBC ไดรเวอร์ไว้ในทุกตัว
- การติดตั้งและการดูแลระบบ ไคลเอนต์ทำได้ง่าย และสะดวก
- ยังต้องการ ไดรเวอร์สำหรับแต่ละผลิตภัณฑ์ เพื่อติดตั้งไว้ที่แอปพลิเคชันเซิร์ฟเวอร์

ประเภท 4 (Type 4: Direct-to-DB)

ลักษณะการทำงานของ JDBC ในประเภทที่ 4 มีการทำงานโดยการส่งคำสั่ง SQL จากไคลเอนต์ไปที่เซิร์ฟเวอร์ฐานข้อมูล โดยตรงผ่านระบบเครือข่าย

- มีประสิทธิภาพในการทำงานดีที่สุด เมื่อเทียบกับทุกประเภทที่กล่าวมา
- มีความยุ่งยากในการพัฒนาโปรแกรมเพื่อทำงานในฝั่งไคลเอนต์ เพราะผู้พัฒนาต้องเรียนรู้การทำงานของเซิร์ฟเวอร์แต่ละตัวก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การดำเนินงานและการออกแบบระบบ

3.1 การสำรวจและวิจัยระบบเกมออนไลน์

3.1.1 Gunbound

Gunbound เป็นเกมแนว Shooting RPG ซึ่งมีเครื่องชนิดมากมายหลายชนิดที่มีประสิทธิภาพและนำไปประยุกต์ใช้กับกลยุทธ์ต่างๆ ในการต่อสู้ได้เป็นอย่างดี นอกจากนี้ยังมีปรากฏการณ์ธรรมชาติที่เรียกว่า “Moondisk” ที่สามารถเปลี่ยนแปลงพลังของแต่ละแผนที่ได้อย่างเหลือเชื่อ ทั้งยังมีมก่อกวนค่าดีเลย์กระสุนของเครื่องชนิดและไอเท็มช่วยเหลือที่สามารถกำหนดอันดับในการยิงของผู้เล่นได้ด้วย ซึ่งจะนำไปสู่การปรับเปลี่ยนกลยุทธ์เฉพาะหน้าและทำให้เกมมีสีสันมากขึ้น เพราะช่วงจังหวะในการเล่นจะเปลี่ยนแปลงไปตลอดเวลา หากท่านไม่เลือกใช้อาวุธอย่างเหมาะสมแล้ว ท่านอาจกลายเป็นผู้แพ้ได้อย่างง่ายดาย

ในแต่ละเทิร์น ท่านสามารถเลือกใช้ตัวช่วยได้มากมาย ไม่ว่าจะเลือกยิง หรือจุดพื้นที่เพื่อซ่อนตัว ความหลากหลายและซับซ้อนเหล่านี้ ทำให้ผู้เล่นรู้สึกราวกับเป็นเกมที่มีชีวิต

ผู้เล่นสามารถเลือกใช้เครื่องชนิดได้ตามที่ต้องการ หรือจะเลือกสวมใส่เครื่องแต่งกายตามแบบที่ชอบ ทั้งนี้เครื่องแต่งกายที่สวมใส่จะเพิ่มความสามารถของเครื่องชนิดที่ผู้เล่นขับเคลื่อนด้วย จากนั้นท่านจะเข้าสู่สนามการต่อสู้ หากท่านชนะในการแข่งขัน ท่านจะได้รับ โบนัสจากการชนะ ซึ่งสามารถนำทองที่ได้รับไปซื้อเครื่องแต่งกายที่ต้องการได้

Gunbound เกมเอนจิน

- ใช้เอนจินที่สร้างขึ้นเพื่อเกมแนว Shooting RPG เท่านั้น ซึ่งสนับสนุนทั้งภาพ 2 มิติ / 3 มิติ
- สนับสนุนสี 16 Bit ด้วยความละเอียดของหน้าจอ 800*600 ฉากหลังเป็นภาพ โพลีกอน 3 มิติ
- สนับสนุนการแสดงผล 3 มิติ (การแสดงผลแบบการ์ตูนตามเวลาจริง, แสงสีหลายทิศทาง, สภาพแวดล้อมเปลี่ยนแปลง, mopping)
- สนับสนุนการแสดงผล 2 มิติ (เลื่อนหน้าจอได้อย่างอิสระ, รูปแบบการแสดงผลภาพ 2 มิติหลากหลาย, ฟังก์ชัน Light & Spinning)

ระบบความต้องการขั้นต่ำของเครื่อง

ระบบความต้องการขั้นต่ำ

CPU : Pentium 300A

Memory : 64 MB

Video Card : i740(สนับสนุนการแสดงผล 3มิติ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Video Memory : มากกว่า 8 MB, สนับสนุน DirectX8.0

O/S : Window 98

Internet Explorer รุ่น 4.0 หรือสูงกว่า

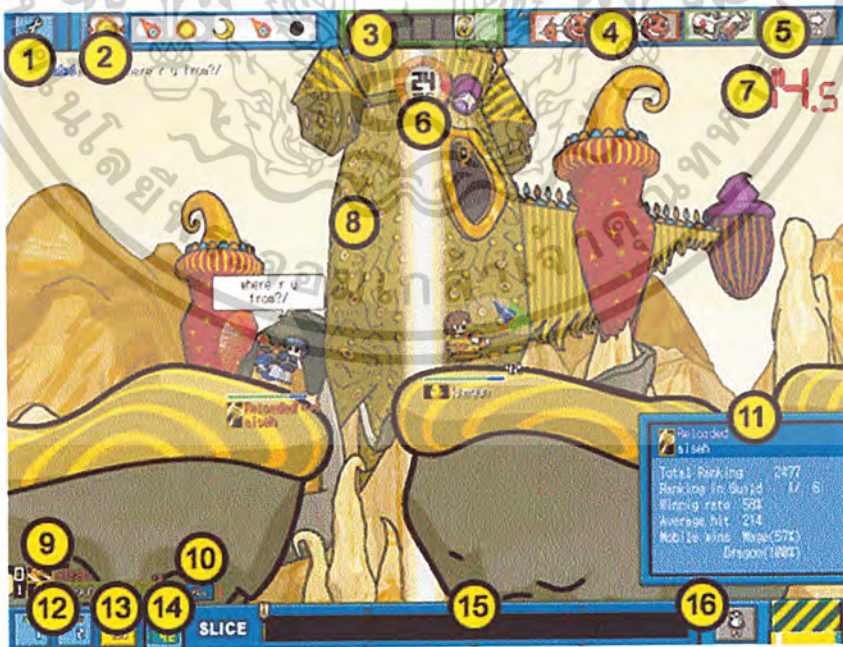
กฎกติกาและวิธีการเล่น

ผู้เล่นสามารถทำการสมัครเพื่อขอ ID และ Password ได้ที่ <http://www.gunbound.in.th> เมื่อได้ ID และ Password มาแล้วสามารถเข้าเล่นเกมได้ที่

Gunboundเป็นเกมที่ผู้เล่นจะต้องนำตัวละครของตนเองเข้าต่อสู้กับผู้เล่นอื่นๆ ทั้งในแบบ1-1, 2-2, 3-3 และ 4-4 ซึ่งการเล่นของgunboundนี้จะเป็นการเล่นในแนวของเกม shooting ซึ่งเมื่อเข้าไปในตัวเกมแล้วผู้เล่นจะต้องทำการกระชาระยะของการยิง, มุมที่จะทำการยิงของกระสุนออกไป, ความแรงของการยิง รวมทั้งต้องคำนึงถึงแรงลมที่พัดในแต่ละเทิร์นด้วย เมื่อผู้เล่นเล่นชนะหรือหากยิงทำพลังทำลายได้สูงก็จะได้รับ Gold(เงิน)และGP, Gold จะสามารถสะสมเพื่อไปทำการซื้อชุดเพิ่มความสามารถของตัวละครผู้เล่นได้ ส่วนGPนั้น คือค่าคะแนนระดับเมื่อค่าคะแนนระดับของผู้เล่นเพิ่มขึ้นไปถึงระดับที่กำหนดไว้ผู้เล่นก็จะเปลี่ยนขึ้นไปอยู่ในระดับที่สูงขึ้น และทำให้สามารถไปเล่นในserver ที่สูงขึ้นไปได้

แนะนำเครื่องยนต์(ตัวละคร)

ในเกมGunboundนี้ตัวละครของผู้เล่นจะมีลักษณะเป็นเครื่องยนต์ต่างๆ โดยที่ผู้เล่นสามารถที่จะเลือกเล่นเครื่องยนต์ใดๆก็ได้ โดยแต่ละเครื่องยนต์ก็จะมีข้อดีข้อเสียที่แตกต่างกันออกไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สามารถกดคีย์ Esc เพื่อปรับแต่งตัวเลือก หรือออกจากเกม
2. ข้อมูลด้านล่างคือ คดวงดาวแห่งปรากฏการณ์ต่างๆ (MoonDisk)
3. จำนวนทองที่ได้รับขณะเล่นเกม ค่าโบนัสที่ได้รับจากเกมการเล่น: โบนัสจาก; โจมตีโดยตรง, ยิงมุมสูง, เมื่อผู้เล่นตายจากการ โจมตีของคุณ, เมื่อ โจมตีผู้เล่นหนึ่งแล้วมีพลังรวมจากการ โจมตีนั้นถึง 1000 หรือ 2000 คุณจะ ได้รับคะแนนและ โบนัส สามารถใช้ทองนี้ซื้อ ไอเท็มพิเศษจากร้านเสื้อผ้าได้
4. หน้าต่างไอเท็ม (จากซ้ายไปขวา เป็นขอร์ตคัตตามช่องจาก F1-F6) (เมื่ออยู่ใน Death mode จะมีข้อความแสดงขึ้นที่หน้าต่างไอเท็ม และเป็นการจำกัดการใช้ไอเท็ม)
5. คุณสามารถกดการ โจมตีในรอบนั้น (ให้คลิกที่ 'TURN' หรือคคีย์ F8)
6. ทิศทางและความเร็วลมในเกม (ลูกศรสีแดงแสดงทิศทางลม และโดยปกติจะพัดขึ้นสูงและลงต่ำ ค่าความเร็วลมสูงสุดอยู่ที่ 32)
7. เวลาที่กำหนดในแต่ละรอบของผู้เล่น และแสดงเวลาแบบนับถอยหลัง
8. ฉากของเกม
9. รายชื่อผู้เล่นที่เข้าร่วมห้อง และอันดับของผู้เล่นแต่ละคนจะขยับสูงขึ้นตามรอบของผู้เล่น
10. หน้าต่างสนทนา (ถ้าเลือก "ทั้งหมด" และสนทนา ข้อความจะถูกส่งไปยังผู้เล่นทั้งหมดที่อยู่ในห้องแต่หากคลิกที่ปุ่ม "ทั้งหมด" อีกครั้ง จะเปลี่ยนเข้าสู่โหมด "ทีม" ทำให้สามารถสนทนากันในทีมได้ขอร์ตคัต F9)
11. แสดงข้อมูลของผู้เล่นในขณะนั้น
12. เลือกอาวุธ ได้ 2 แบบ (ขอร์ตคัต TAB)
13. ทักษะพิเศษ (อาวุธที่ 3 ของเครื่องยนต์ และมีทักษะพิเศษ แต่ใช้งานได้ทุกๆ 4 รอบเท่านั้น)
14. แสดงมุมในการยิง
15. เครื่องวัดที่แสดงความแรงในการยิงจากเครื่องยนต์ของคุณ (รูปแบบการยิงมี 3 แบบ เปลี่ยนได้ โดยการคลิก Esc ที่อยู่มุมซ้ายบนของหน้าจอ เพื่อเปิดหน้าต่างเมนู และเลือก) รูปแบบการยิงจะมีดังนี้คือ
 - Slice เมื่อกดSpacebar แล้วหลอดพลังการยิงจะเพิ่มขึ้นเรื่อยๆต้องการยิงให้ปล่อยSpacebar
 - Drag ใช้เมาส์คลิกแล้วลากเมาส์จากซ้ายไปขวาเพื่อเพิ่มหลอดพลังการยิงวิธีนี้สามารถ ลด Delay จากการยิงได้เป็นอย่างดี
 - Sniper คล้ายกับSlice แต่เมื่อปล่อยSpacebar จะมีจังหวะหยุดเพ็งให้อีกจังหวะหนึ่งเพื่อเพิ่มความแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 Ragnarok Online

Ragnarok Online เป็นเกม RPG ออนไลน์ Multiplayer ขนาดใหญ่ที่สร้างขึ้นโดยใช้โครงสร้างของแบคกราวด์และเนื้อเรื่องจากการ์ตูน 'Ragnarok' ซึ่งเขียนโดย Myung-Jin Lee และมันได้ถูกพัฒนาให้เป็นเกม RPG ที่เน้นคุณลักษณะในการ รวมกลุ่มเป็นสังคม

Ragnarok ได้ถูกออกแบบมาเพื่อเป็นตัวแทนลักษณะของผู้เล่นแต่ละคน เพื่อให้ ผู้เล่นแต่ละคน ได้มีส่วนร่วมในสังคมภายในเกม เมื่อเกมดำเนินไป เนื้อเรื่องของ Ragnarok จะสร้างความเพลิดเพลินให้กับผู้เล่น ได้อย่างไม่น่าเชื่อ

ตัวละครของเกมนี้เป็นแบบ 2 มิติท่ามกลางแบคกราวด์ 3 มิติ ซึ่งได้รับการอัปเดตขึ้นมาจากหนึ่งในเกมล่าสุดของ Gravity "Arcturus" และด้วยเสียงดนตรีและ BGM อันหลากหลาย ที่ประพันธ์โดย TeMP จะทำให้ Ragnarok เพิ่มความพิเศษขึ้นอีกมากทีเดียว

หน้าตาอินเตอร์เฟซในเกมจะทำให้ผู้เล่นสามารถใช้ที่เจอร์ส่วนมากในเกมได้ด้วยการใช้เมาส์ บังคับ และผู้เล่นยังสามารถปรับแต่งที่เจอร์ส่วนใหญ่ในเกมสำหรับเขา/เธอเพื่อให้เล่นเกมได้ดีขึ้นอีกด้วย

ระบบความต้องการขั้นต่ำของเครื่อง

เครื่องคอมพิวเตอร์ควรมีคุณสมบัติตามความต้องการขั้นต่ำของ Ragnarok Online ดังต่อไปนี้

CPU: Intel, AMD หรืออื่นๆ ความเร็วอย่างต่ำ 600 MHz

Ram: 192 MB

Harddisk: พื้นที่ว่างประมาณ 1.4 GB

VGA: ระบบแสดงผลที่รองรับ 3D และรองรับ DirectX 8.0

มี Ram 8 MB

Sound: ระบบเสียงที่รองรับ Direct Sound

Software OS: Windows 98, Windows ME, Windows 2000, Windows XP

อื่นๆ: DirectX 8.0, การเชื่อมต่อ Internet ความเร็ว 56 K

กฎกติกา และวิธีการเล่น

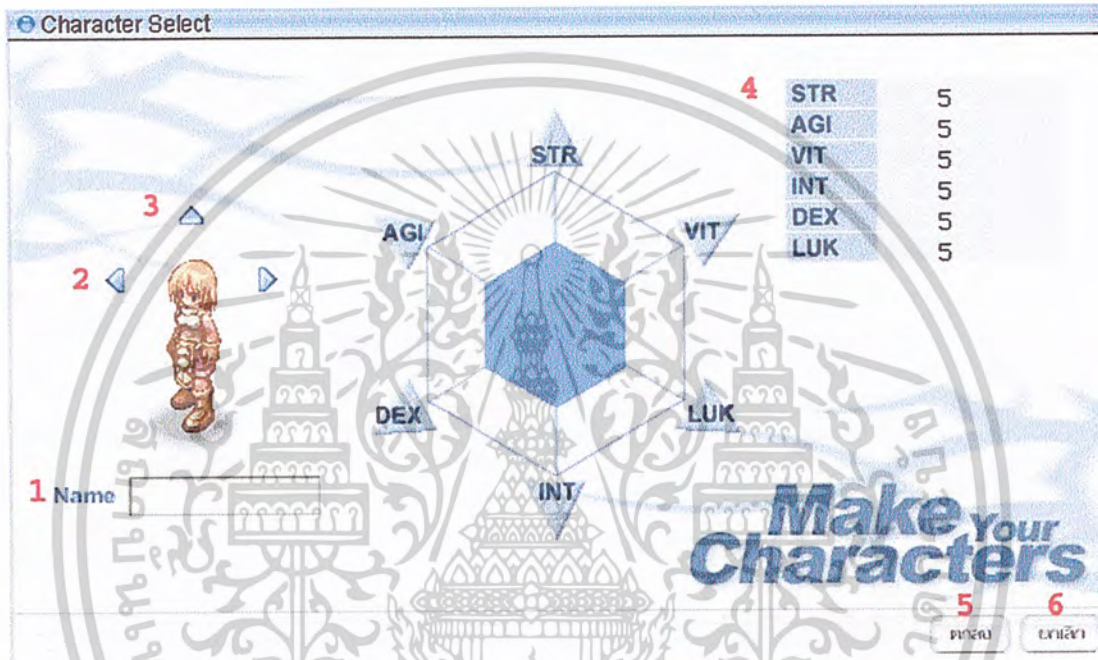
ผู้เล่นทำการสมัครเพื่อขอ ID และ Password ในการเล่นเกมได้ที่ <http://www.ragnarok.in.th> เมื่อได้ ID และ Password แล้วก็สามารถเข้าเล่นเกมได้ทันที

Ragnarok Online เป็นเกม Online แบบ Action กึ่ง RPG ในรูปแบบตัวละครแบบ 2D animation บนฉากแบบ 3D Full Polygon ซึ่งสามารถหมุนได้รอบด้านรวมทั้งยังสามารถปรับให้มองจากมุมสูงหรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มุมที่ต่ำมาก ๆ ได้ ลักษณะการเล่นผู้เล่นจะต้องนำตัวละครของตนเอง เข้าต่อสู้กับ monster โดยเมื่อสามารถ ฆ่า monster ได้แล้วผู้เล่นจะได้รับค่าประสบการณ์ทั้ง base และ job รวมถึง item ต่างๆ ที่ จะปรากฏแบบ สุ่มด้วย ค่าประสบการณ์ base เมื่อเกินจนครบ 100% แล้วก็จะทำให้ level up และสามารถเพิ่มขึ้น status ของตัวละครให้แข็งแกร่งขึ้นได้, ค่าประสบการณ์ job เมื่อครบ 100% จะทำให้ job up และได้ skill point เพิ่มขึ้น 1 point สำหรับเพิ่มค่า skill (ทำความสามารถต่างๆ) ให้กับตัวละคร ได้

วิธีการเล่นเบื้องต้น



หากเข้ามาในหน้าต่างสร้างตัวละคร จะพบตัวละคร Ragnarok Online ซึ่งมีเพศตามที่ท่านระบุเมื่อลง ทะเบียนชื่อของตัวละคร จะเหมือนกับชื่อตัวละครอื่นใน server เดียวกันไม่ได้ และตำแหน่งต่างๆ บนหน้า จอสร้างตัวละครจะมีความหมายดังนี้

1. ชื่อตัวละคร
2. ทรงผม
3. สีผม
4. Status (สามารถปรับค่าเริ่มต้นได้)
5. ปุ่มยืนยันการสร้างตัวละคร
6. ปุ่มยกเลิกการสร้างตัวละคร

Status ในเกมส์นี้จะมีอยู่ 6 ตัวได้แก่

Str - Strength กำลัง มีผลต่อพลังโจมตี , แบกรับน้ำหนักได้มากขึ้น

Agi - Agility ความคล่องตัว มีผลต่อโอกาสในการหลบหลีกจากการ โจมตีศัตรู, ความเร็วในการ

โจมตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dex - Dexterity ความเร็ว มีผลต่อความแม่นยำในการโจมตี, ทำให้ค่าต่ำสุดในการโจมตีของคุณเพิ่มขึ้น, ร่นระยะเวลาในการร่ายเวทมนต์และการใช้ Skill

Int - Intelligent ความฉลาด มีผลต่อ SP สูงสุด, พลังโจมตีของเวทมนตร์เพิ่มขึ้น, พลังป้องกันเวทมนตร์เพิ่มขึ้น

Luk - Luck โชค เพิ่มอัตราการโจมตีด้วย Critical , อัตราการหลบหลีกแบบ Perfect Dodge

Vit - Vitality พลังชีวิต มีผลต่อ HP สูงสุด, ความเร็วในการฟื้นฟู HP, พลังป้องกันมากขึ้น, ประสิทธิภาพของ Item เพิ่ม HP จะดีขึ้น

เมื่อสร้างตัวละครเรียบร้อยแล้ว สามารถเข้าเล่นเกมได้ที่

ข้อมูลแรกที่ท่านเห็นตั้งแต่เริ่มต้นเกมคือ แถบสีเขียวและแถบสีน้ำเงินที่อยู่ด้านล่างของตัวละคร แถบสีเขียว คือ Hit Point หมายถึงพลังสุขภาพ แถบสีน้ำเงินคือ Skill Point มีไว้สำหรับใช้ Skill

ข้อมูลพื้นฐาน		7	
~แก่นกะโหลก	1 HP 40 / 40	8 สถานะ:	ตัวริชชี่ 2
Novice	2 SP 1 / 1	9 ไอเท็ม:	สวมใส่ 3
3 Base Lv. 1		10 ม้า:	พบ 1 4
4 Job Lv. 1		11 ชุด:	พบ 5
5 Weight : 50 / 2270	6 Zeny : 0		

ข้อมูลถัดมาคือ กรอบข้อมูลพื้นฐานที่อยู่ทางซ้ายบน

(Alt+V) เป็นหน้าต่างที่บอกข้อมูลพื้นฐานต่างๆ ของ ตัวละคร

HP - Health Point พลังชีวิต

SP - Skill Point พลังสำหรับใช้ Skill

Base Level - ระดับของตัวละคร

Job Level - ระดับของอาชีพ

Weight น้ำหนัก item ที่ถือ / น้ำหนักสูงสุดที่ถือได้ หากน้ำหนักที่แบกอยู่เกิน 50% พลัง HP/SP จะไม่สามารถเพิ่ม อัปเดต โนม์ได้ หากน้ำหนักที่แบกเกิน 90% จะไม่สามารถทำการต่อสู้ได้

Zeny แสดงค่าเงินในเกม มีหน่วยเป็น Zeny

Minimize ปุ่มสำหรับย่อหน้าต่าง

Status หน้าต่างสถานะของตัวละคร

Items หน้าต่างไอเท็ม

Skill หน้าต่าง Skill

Chat หน้าต่างสร้างห้องสนทนา

Options หน้าต่างตัวเลือก

Equip หน้าต่างอุปกรณ์

Map ปุ่มเปิด/ปิด แผนที่

Friend หน้าต่างจัดการกับเพื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 Survival project

Survivor Project คือ เกม action online ที่เล่นผ่านระบบเครือข่าย ผู้เล่นสามารถต่อสู้กับผู้เล่นอื่นพร้อมๆกัน(online) โดยแบ่งออกเป็น 8 รูปแบบ โดยจะมีการกิจ แตกต่างกันไป และมีแผนที่ถึง 8 แผนที่ซึ่งผู้เล่นจะได้สัมผัสกับบรรยากาศอันหลากหลาย ภายในเกมมีตัวละครให้เลือกทั้งหมด 8 ตัวละคร แต่ละตัวจะมีอาวุธประจำกาย และข้อดีข้อเสียแตกต่างกัน ผู้เล่นสามารถเลือกเป็นตัวละครใดก็ได้ และสามารถเลือกแข่งกับคู่ต่อสู้แบบเป็นทีม ตั้งแต่ 2 ทีมขึ้นไป จนถึง 4 ทีม

Survival project เป็นเกมรูปแบบ real time ในเกมผู้เล่นจะได้พบกับระบบการเล่นมากมาย เช่น ระบบสมาคม (Guild) และ benefactor system ที่ช่วยผู้เล่นในการสร้างสังคมภายในเกม ซึ่งระบบของ graphic จะเป็นแบบ 2D animation ซึ่งเมื่อผู้เล่นสะสมเงินรางวัลและการ์ดต่างๆ ได้แล้ว ก็จะสามารถนำไป upgrade อาวุธ, ชุดป้องกัน, เวทมนต์ ให้มีประสิทธิภาพยิ่งขึ้น และเพิ่มระดับ Rang ของตนเองเพื่อ ไปเล่นยัง channel ที่ยากยิ่งขึ้นไปอีกได้

กฎ กติกาและวิธีการเล่น

เมื่อผู้เล่นทำการสมัครเพื่อขอ ID และ Password ในการเข้าเล่นเกมแล้วก็จะสามารถเข้าเล่นเกมได้ทันที

ตัวละคร

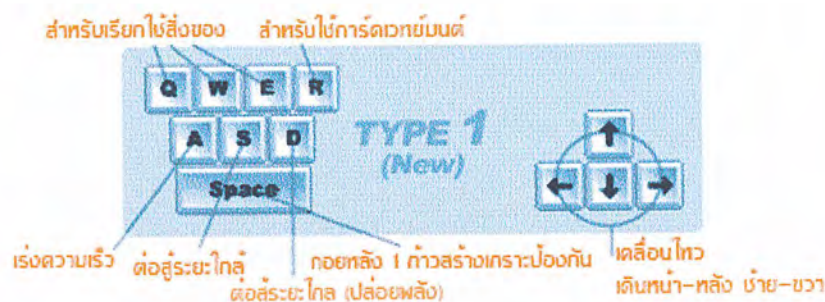
ตัวละครของ survival project จะมีให้เลือกทั้งหมด 8 ตัวด้วยกัน ซึ่งแต่ละตัวมีความสามารถต่างกันออกไป

ระบบlevelของตัวละคร

ผู้เล่นจะมีระดับหรือ Rang ของตนเอง ซึ่งตัวเกมจะแบ่งออกเป็นห้องๆ แต่ละห้องจะกำหนด Rang ของผู้เล่นที่สามารถเข้าไปเล่นได้เอาไว้ Rang ของผู้เล่นสามารถเพิ่มขึ้นได้ ด้วยการเก็บค่าประสบการณ์จากการเข้าเล่นใน modeต่างๆ ของเกม อาทิเช่น quest, survival, event เป็นต้น เมื่อตัวละครเพิ่มระดับค่าประสบการณ์ถึงระดับที่กำหนด Rang ก็จะเพิ่มขึ้น 1 Rang ซึ่งในเกมนี้ lv ของตัวละครจะเสมือนระดับที่กำหนดห้องให้เข้าไปเล่นเท่านั้น จะ ไม่มีผลต่อระดับความสามารถของตัวละครแต่อย่างใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการควบคุม



Survival project เป็นเกมที่มีการควบคุมตัวละครดังนี้

- ขึ้นลงซ้ายขวา บังคับทิศทางตัวละคร ไปยังทิศทางบนล่างซ้ายขวา
- A เคลื่อนที่ตัวละครอย่างรวดเร็ว
- S โจมตีระยะใกล้
- D โจมตีระยะไกล
- Q,W,E ใช้ item ลำดับที่ 1,2,3
- R ใช้เวทย์มนต์
- Spacebar ทำการป้องกันหรือกอดค้างไว้เพื่อเดินถอยหลัง

ระบบItem

Item ในเกม survival project จะแบ่งออกเป็น 2 แบบคือ

1.Card

ระบบของการ์ด สามารถแบ่งออกเป็น 4 ประเภท ได้แก่ การ์ดทอง (Gold type), การ์ดดำ (Black type), การ์ดขาว (White type) และ การ์ดเงิน (Silver type)

การ์ดทอง (Gold Type)

เป็นการ์ดที่จะเพิ่มความสามารถให้กับอาวุธ หากมีการรวมการ์ดเข้ากับการ์ดขาว (White type) การ์ดทองมีคุณสมบัติคล้ายๆกันกับการ์ดดำ (Black type) แตกต่างกัน ตรงที่มีโอกาสสำเร็จมากกว่า เมื่อมีการรวมเข้ากับการ์ดชนิดอื่นๆ เมื่อผ่านพ้นช่วง เวลาหนึ่งไป การ์ดทองจะเปลี่ยนเป็นการ์ดดำ

การ์ดดำ (Black Type)

คุณสมบัติโดยทั่วไปของการ์ดดำจะเหมือนกับการ์ดทอง แตกต่างกันที่ การ์ดดำ จะมีโอกาสสำเร็จน้อยกว่าการ์ดทอง เมื่อมีการรวมกับการ์ดอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ์ดเงิน (Silver Type)

เป็นการ์ดที่มอบสิทธิพิเศษให้กับคุณ นอกเหนือจากการต่อสู้ การ์ดนี้จะอนุญาตให้คุณสามารถเลือก แผนที่พิเศษ ระบบพิเศษ ตัวละครพิเศษ และการแสดงอารมณ์ต่างๆ และคุณยังสามารถจำกัดระบบและจำนวนคนที่เข้ามาในห้องได้อีกด้วย แต่จะเป็นแค่ เพียงช่วงเวลาหนึ่งเท่านั้น เมื่อผ่านพ้นไป การ์ดนี้จะอันตรายหายไปทันที

การ์ดขาว (White Type)

การ์ดใบนี้ ถือเป็นการ์ดที่เป็นวัตถุพิเศษ ที่คุณ ไม่สามารถใช้การ์ดนี้เพียงใบเดียวได้ เป็นการ์ดที่จะต้องมีการรวมความสามารถเข้ากับการ์ดทอง (Gold type) และการ์ดดำ (Black type) เมื่อรวมความสามารถแล้ว การ์ดใบนี้จะอันตรายหายไปทันที

และการ์ดดำ (Black Type) กับการ์ดขาว (White Type) ซึ่งเป็นการ์ดที่เปรียบเทียบได้ว่าเป็นหัวใจหลักของitemในเกมนี้ ยังแบ่งได้ออกเป็นหมวดหมู่ตามลักษณะได้ดังนี้

1.1 อวรุช - จะแบ่งออกเป็น 4 ธาตุคือ ดิน, น้ำ, ลม, ไฟ ซึ่งแต่ละธาตุก็จะมีอวรุชอยู่ 4 คือ ขวาน, คบ ซึ่ง เป็นอวรุชประเภทเพิ่มความรุนแรงให้การโจมตีระยะใกล้ และ ทากกับธนู ซึ่งเป็นอวรุชประเภทเพิ่มความรุนแรงให้การโจมตีระยะไกล ในแต่ละธาตุนั้นก็จะมีความสามารถที่เป็นจุดเด่นจุดด้อยต่างกันออกไป เช่น ธาตุไฟจะเน้นที่ความรุนแรงในการ โจมตีแต่จะ โจมตีช้า, ธาตุลมจะเน้นความเร็วในการโจมตีแต่จะ โจมตีเบา เป็นต้น

1.2 ชูคป้องกัน - จะแบ่งออกเป็น 4 ธาตุ เหมือนกับอวรุช (ดิน, น้ำ, ลม, ไฟ) ซึ่งแต่ละธาตุก็จะมีชูคป้องกันอยู่ 4 ชนิดก็คือ

โล่ - ทำงานเมื่อผู้เล่นกดป้องกันเท่านั้นจะช่วยลดความเสียหายจากการ โคน โจมตีได้มาก (หากไม่กดป้องกันก็จะไม่เกิดความสามารถนี้)

เกราะ - ทำงานอัตโนมัติเมื่อผู้เล่น โคน โจมตีจะช่วยลดความเสียหายจากการ โคน โจมตีได้แต่น้อยกว่าโล่

สร้อย - ทำงานเมื่อผู้เล่นทำการพักอยู่กับที่จะช่วยเพิ่มความเร็วในการฟื้นคืนค่า HP, MP, BP

รองเท้า - ทำงานเมื่อเราทำการวิ่ง(กดปุ่มA) โดยจะช่วยเพิ่มความเร่งและความเร็วสูงสุดให้การวิ่งของตัวละคร

1.3 เวทย์มนต์ - จะแบ่งออกเป็น 4 ธาตุเช่นกัน (ดิน, น้ำ, ลม, ไฟ) ซึ่งเมื่อกดใช้เวทย์แล้วตัวละครจะใช้ระยะเวลาในการร้ายเวทย์มนต์ซักระยะหนึ่ง และเวทย์มนต์มีสิทธิ์ที่จะร้ายล้มเหลวได้ด้วย (ร้ายเวทย์ไม่ออก) โดยที่เวทย์มนต์ธาตุน้ำและดินจะเป็นการ โจมตีระยะใกล้ ส่วนเวทย์มนต์ ไฟและลมจะเป็นการโจมตีระยะไกล

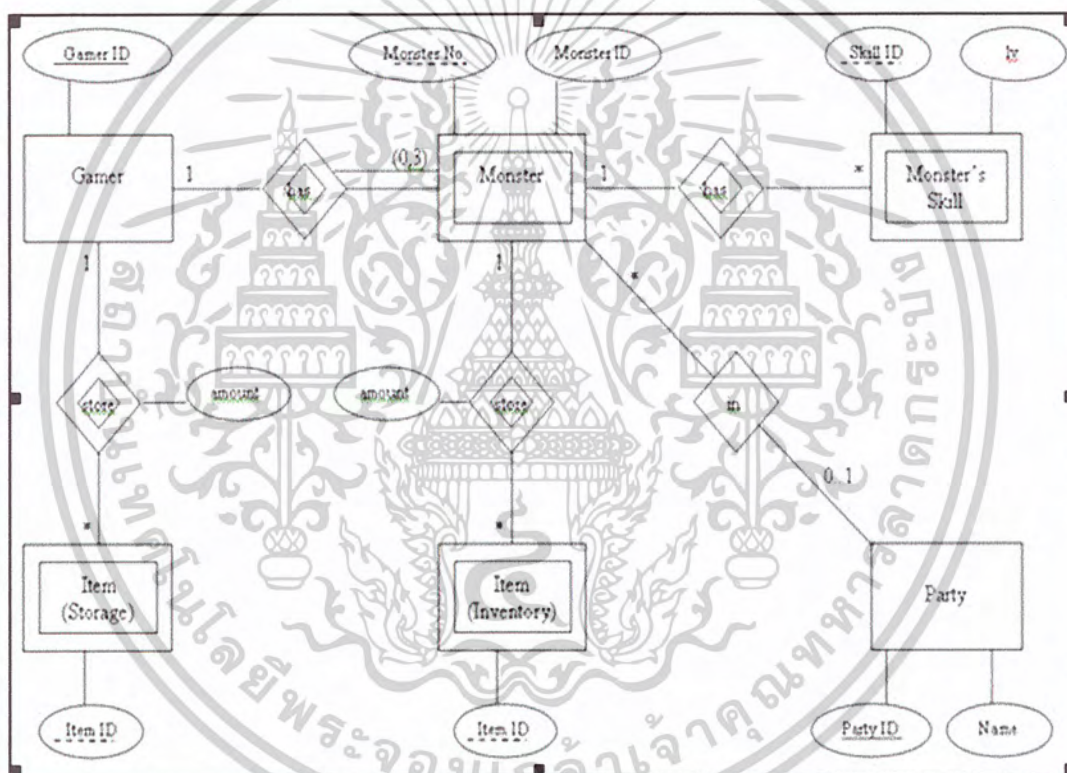
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.Scroll

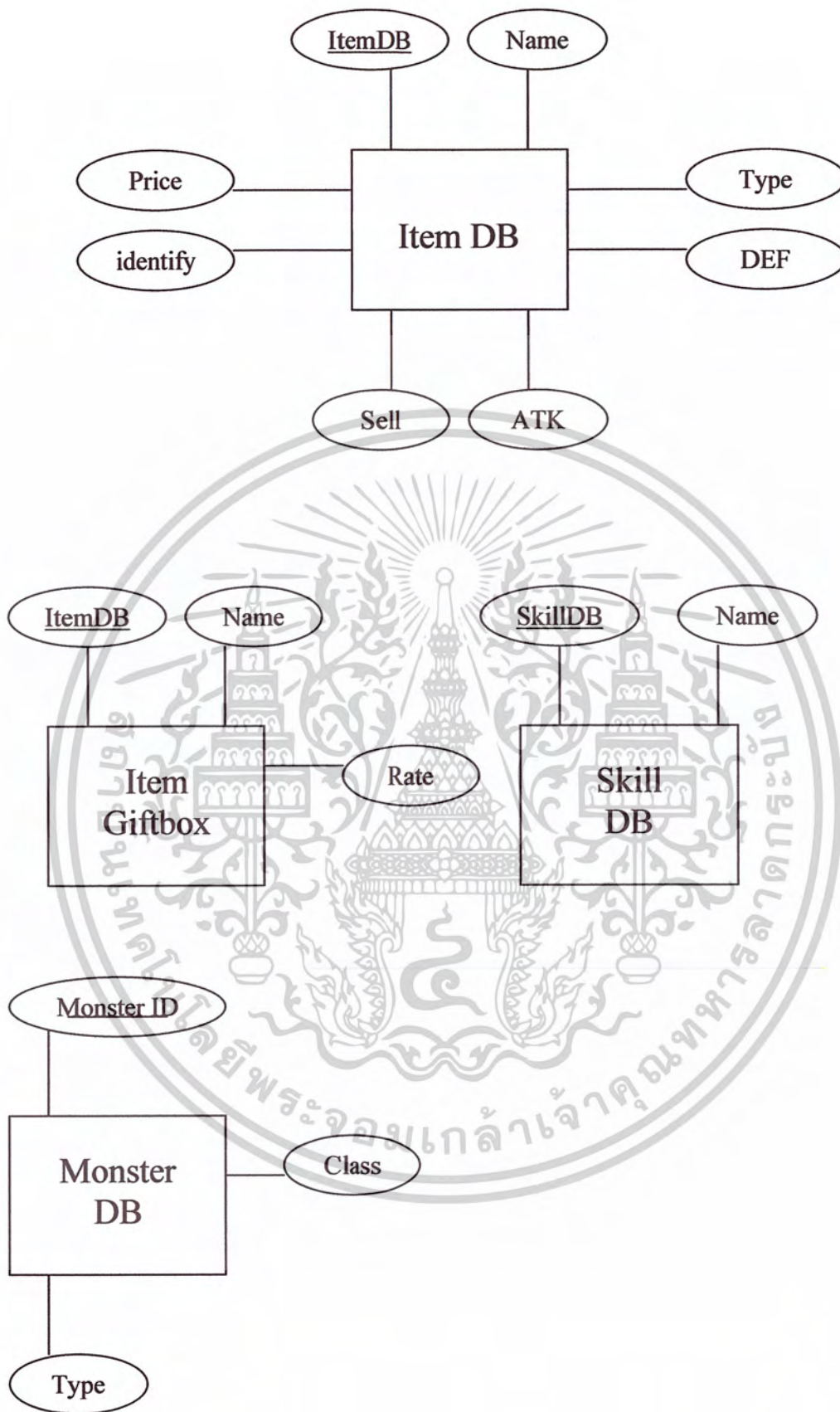
เป็น item แบบใช้ครั้งเดียวแล้วหมดไปต้องซื้อมาใช้ใหม่ โดย scroll แต่ละอันจะมีความสามารถแตกต่างกันออกไป อย่างเช่น ทำให้สามารถยิงกระสุน โจมตีระยะไกลได้ 3 นัดพร้อมๆกัน, หรือจะเป็น scroll ป้องกันการ โจมตีทุกชนิดเป็นระยะเวลา 5 วินาที, scroll ค้นหาผู้เล่นทั้งหมดที่อยู่ในmap ทำให้รู้ว่าผู้เล่นใดอยู่ที่ไหนบ้าง, scroll เพิ่มmpหรือbp ให้กับผู้เล่นมี2แบบคือเพิ่มให้กับตนเองคนเดียวกับแบบที่เพิ่มให้กับปาร์ตี้เดียวกับผู้เล่นทั้งปาร์ตี้ เป็นต้น

3.2 โครงสร้างและการทำงานของระบบ

3.2.1 การออกแบบระบบฐานข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จาก ER ใดอะแกรมทั้งหมด เมื่อนำไปทำการนอมอลไลซ์ จะได้ตารางฐานข้อมูลดังนี้

ตารางฐานข้อมูลที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ตาราง Gamer ใช้เก็บข้อมูลส่วนตัวต่างๆของผู้เล่น

Name	Type	NULL	Default	Extras
PRIMARY	GamerID			unique
GamerID	varchar(20)	No	0	
Password	varchar(20)	No		
Name	varchar(40)	No		
IDCard	varchar(13)	No		
Gender	char(1)	No		
Address	text	No		
BirthDate	date	No	0000-00-00	
Tel	varchar(9)	No		
Email	text	No		

2. ตาราง Monster ใช้เก็บรายละเอียดต่างๆของสัตว์ประหลาดที่ผู้เล่นเลี้ยง

Name	Type	NULL	Default	Extras
PRIMARY	MonsterNo,MonsterID,GamerID			unique
GamerID	GamerID			
MonsterNo	int(4) unsigned	No		auto_increment
GamerID	varchar(20)	No		
MonsterID	int(4) unsigned	No	0	
Name	varchar(11)	No		
Level	int(4) unsigned	No	0	
Exp	int(4) unsigned	No	0	
GP	int(10)	No	0	
PartyID	int(4) unsigned	No	0	
Str	int(4) unsigned	No	0	
Agi	int(4) unsigned	No	0	
Vit	int(4) unsigned	No	0	
int	int(4) unsigned	No	0	
Dex	int(4) unsigned	No	0	
BStr	int(4) unsigned	No	0	
BAgi	int(4) unsigned	No	0	
BVit	tinyint(4) unsigned	No	0	
BInt	int(4) unsigned	No	0	
BDex	int(4) unsigned	No	0	
Left_Hand	int(4) unsigned	No	0	
Right_hand	int(4) unsigned	No	0	
Body	int(4) unsigned	No	0	
Eye	int(4) unsigned	No	0	
Muffler	int(4) unsigned	No	0	
Head	int(4) unsigned	No	0	
Feet	int(4) unsigned	No	0	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

◆ Emotion	int(4) unsigned	No	0
◆ Dirt	int(4) unsigned	No	0
◆ MaxHp	int(4) unsigned	No	0
◆ MaxSp	int(4) unsigned	No	0
◆ Color	int(4) unsigned	No	0
◆ StatusPoint	int(4) unsigned	No	0
◆ SkillPoint	int(4) unsigned	No	0
◆ Health	int(4) unsigned	No	0
✓ Constraint monster_ibfk_1 (GamerID) -> gamer (GamerID)			

3. ตาราง MonsterDB ใช้เก็บข้อมูลและรายละเอียดของสัตว์ประหลาดแต่ละประเภทภายในเกม

Name	Type	NULL	Default	Extras
PRIMARY	MonsterID			unique
◆ MonsterID	int(10) unsigned	No		auto_increment
◆ Type	int(4) unsigned	No	0	
◆ Class	int(4) unsigned	No	0	

4. ตาราง MonsterSkill ใช้เก็บรายละเอียดสกิลของสัตว์ประหลาดแต่ละตัวที่ผู้เล่นแต่ละคนเลี้ยงไว้

Name	Type	NULL	Default	Extras
PRIMARY	GamerID, MonsterNo			unique
◆ MonsterNo	int(4) unsigned	No	0	
◆ GamerID	varchar(11)	No	0	
◆ SkillID	int(4) unsigned	No	0	
◆ Skill_Lv	int(4) unsigned	No	0	

5. ตาราง ItemDB ใช้เก็บข้อมูลและรายละเอียดของไอเท็มทั้งหมดภายในเกม

Name	Type	NULL	Default	Extras
PRIMARY	itemID			unique
◆ itemID	int(6) unsigned	No		auto_increment
◆ Name	varchar(20)	No		
◆ type	int(4) unsigned	No	0	
◆ Atk	int(4) unsigned	No	0	
◆ Def	int(4) unsigned	No	0	
◆ identify	text	No		
◆ Price	int(4) unsigned	No	0	
◆ Sell	int(4) unsigned	No	0	

6. ตาราง Item_Giftbox ใช้เก็บข้อมูลของไอเท็มหายาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name	Type	NULL	Default	Extras
PRIMARY	itemID			unique
◆ itemID	int(6) unsigned	No	0	
◆ rate	int(4) unsigned	No	0	

7. ตาราง Inventory ใช้เก็บข้อมูลรายละเอียดของไอเท็มที่ติดตัวอยู่กับสัตว์ประเภทแต่ละตัวของผู้เล่น

Name	Type	NULL	Default	Extras
PRIMARY	inventNo,MonsterNo,GamerID			unique
GamerID	GamerID			
◆ inventNo	int(4) unsigned	No		auto_increment
◆ MonsterNo	int(4) unsigned	No	0	
◆ GamerID	varchar(20)	No		
◆ itemID	int(4) unsigned	No	0	
◆ amount	int(4) unsigned	No	0	
◆ refine	int(4) unsigned	No	0	

8. ตาราง Storage ใช้เก็บข้อมูลรายละเอียดต่างๆของไอเท็มที่อยู่ในที่เก็บของส่วนตัวของผู้เล่นแต่ละ ID

Name	Type	NULL	Default	Extras
PRIMARY	StoreNo,GamerID			unique
GamerID	GamerID			
◆ StoreNo	int(4) unsigned	No		auto_increment
◆ GamerID	varchar(20)	No		
◆ itemID	int(4) unsigned	No	0	
◆ amount	int(4) unsigned	No	0	
◆ refine	int(4) unsigned	No	0	
✓ Constraint storage_ibfk_1	(GamerID) -> gamer (GamerID)			

9. ตาราง Party ใช้เก็บข้อมูลของปาร์ตี้ที่เล่นสร้างขึ้น

Name	Type	NULL	Default	Extras
PRIMARY	PartyID			unique
◆ PartyID	int(6) unsigned	No		auto_increment
◆ Name	varchar(11)	No		
◆ LeaderID	varchar(11)	No		

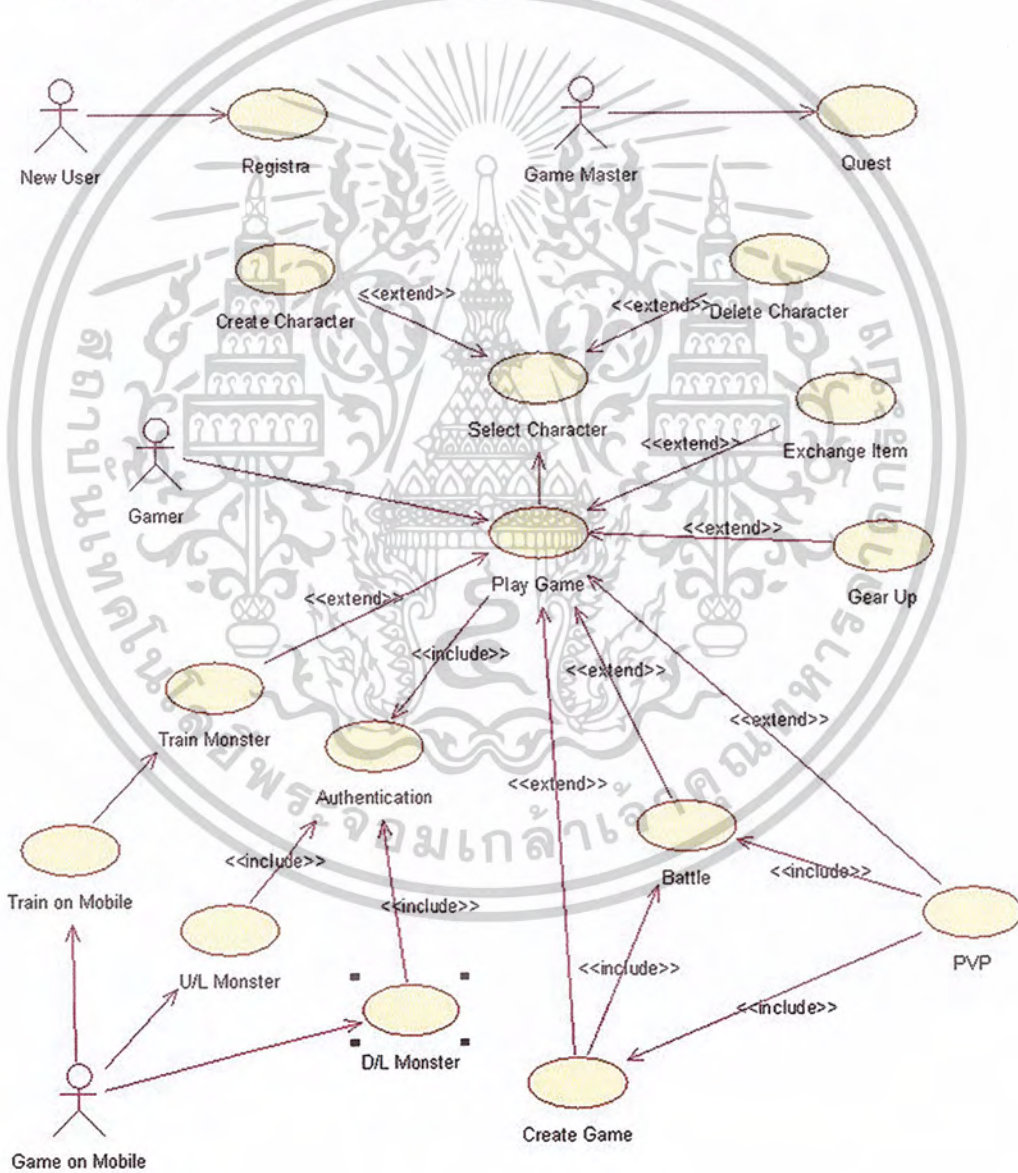
10. ตาราง SkillDB ใช้เก็บข้อมูลและรายละเอียดของสกิลทั้งหมดภายในเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name	Type	NULL	Default	Extras
PRIMARY	SkillID			unique
SkillID	int(6) unsigned	No		auto_increment
Name	varchar(20)	No		

3.2.2 การออกแบบระบบ

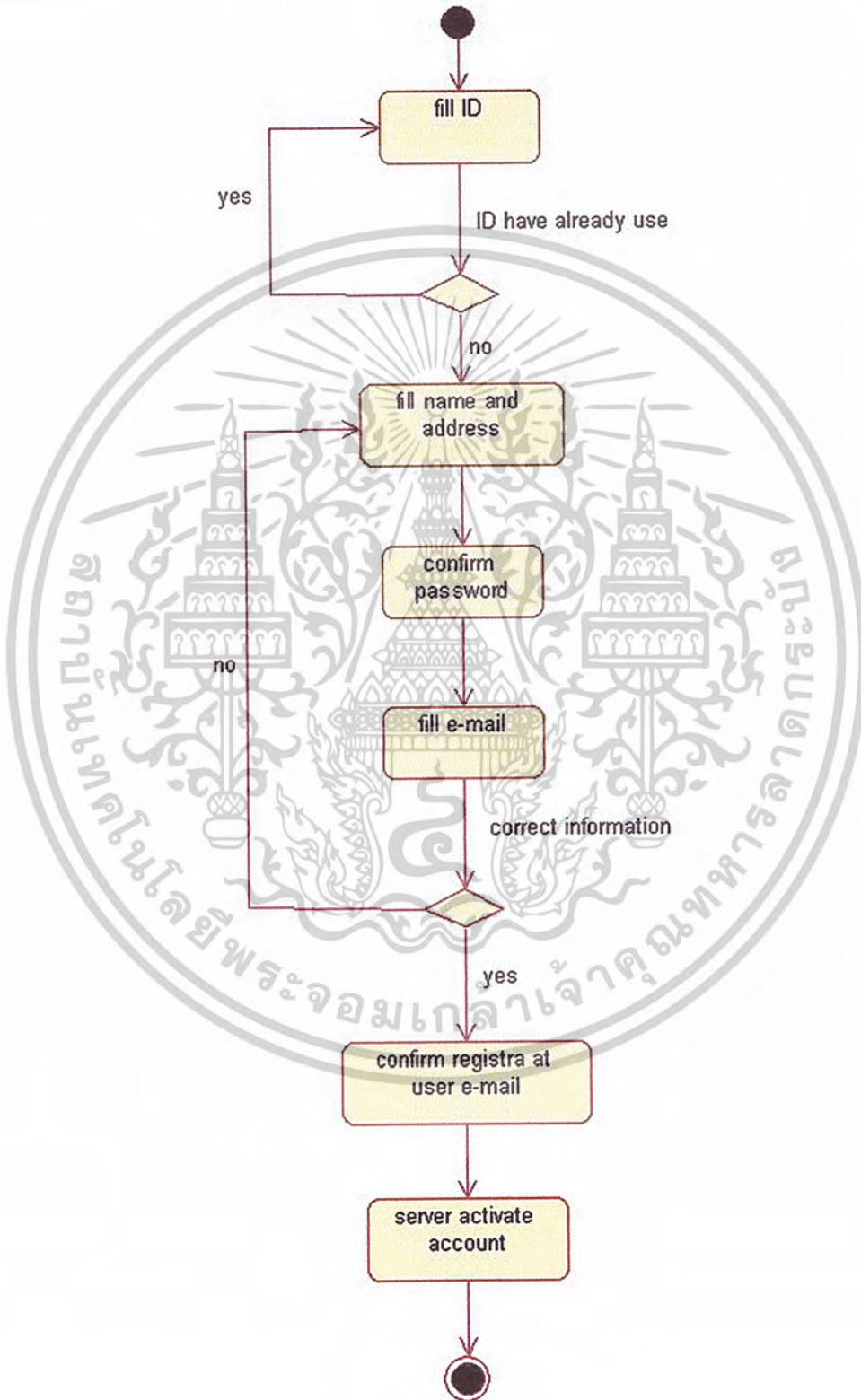
3.2.2.1 Use Case Diagram แสดงความสัมพันธ์ของผู้ใช้กับระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 Activity Diagram

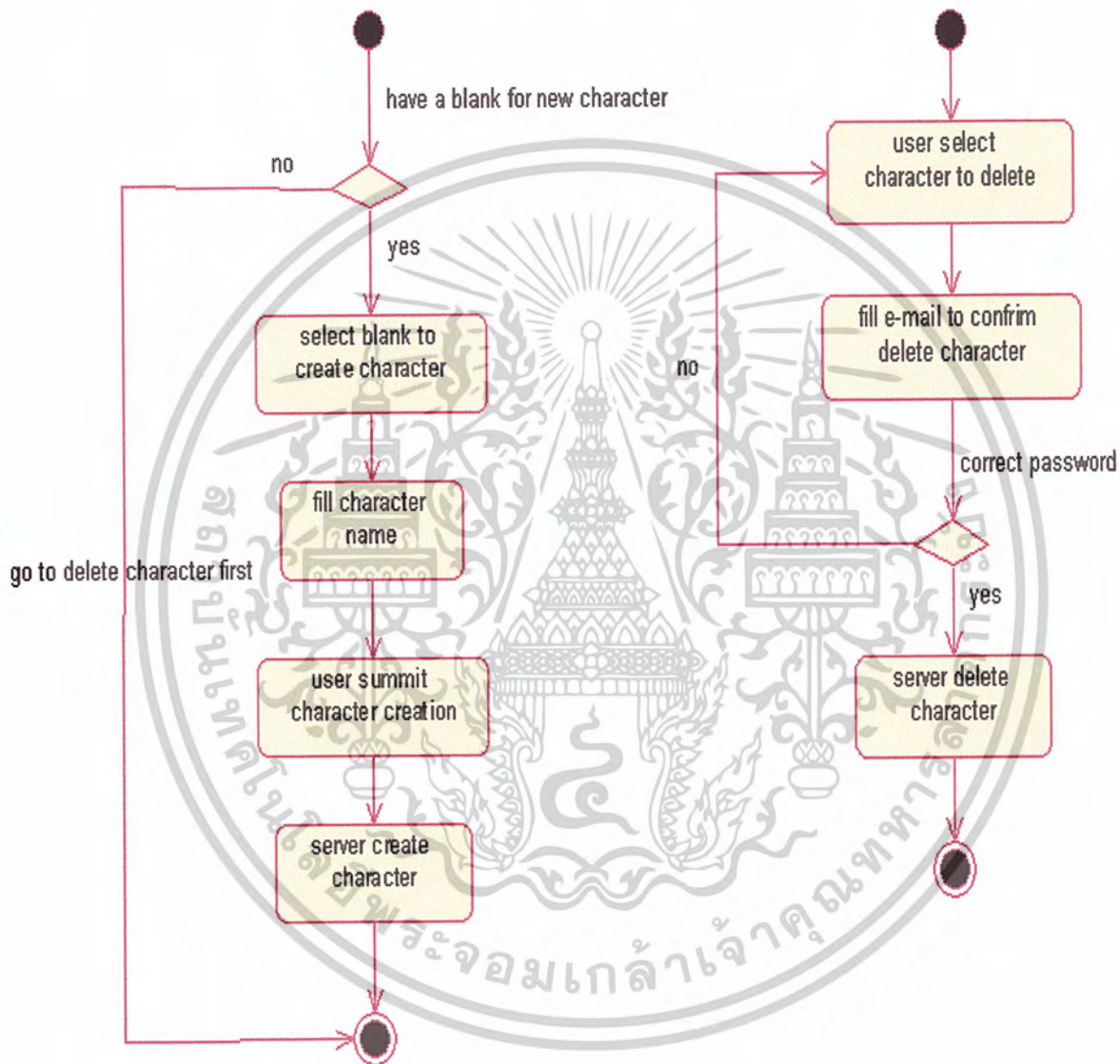
แสดงขั้นตอนการลงทะเบียนเพื่อขอ ID และ Password



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Activity Diagram ขั้นตอนการสร้างตัวละคร

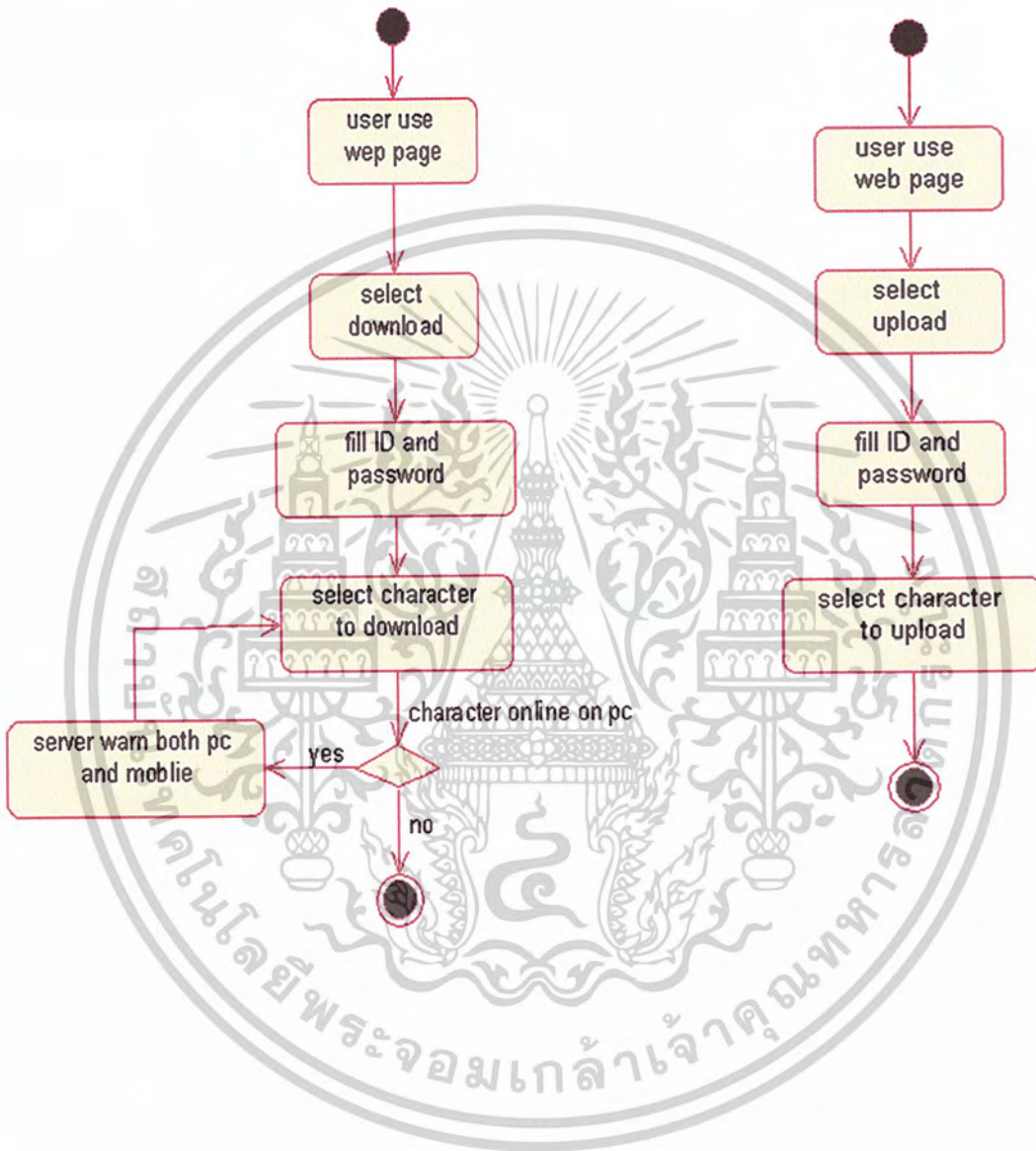
Activity Diagram ขั้นตอนการลบตัวละคร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

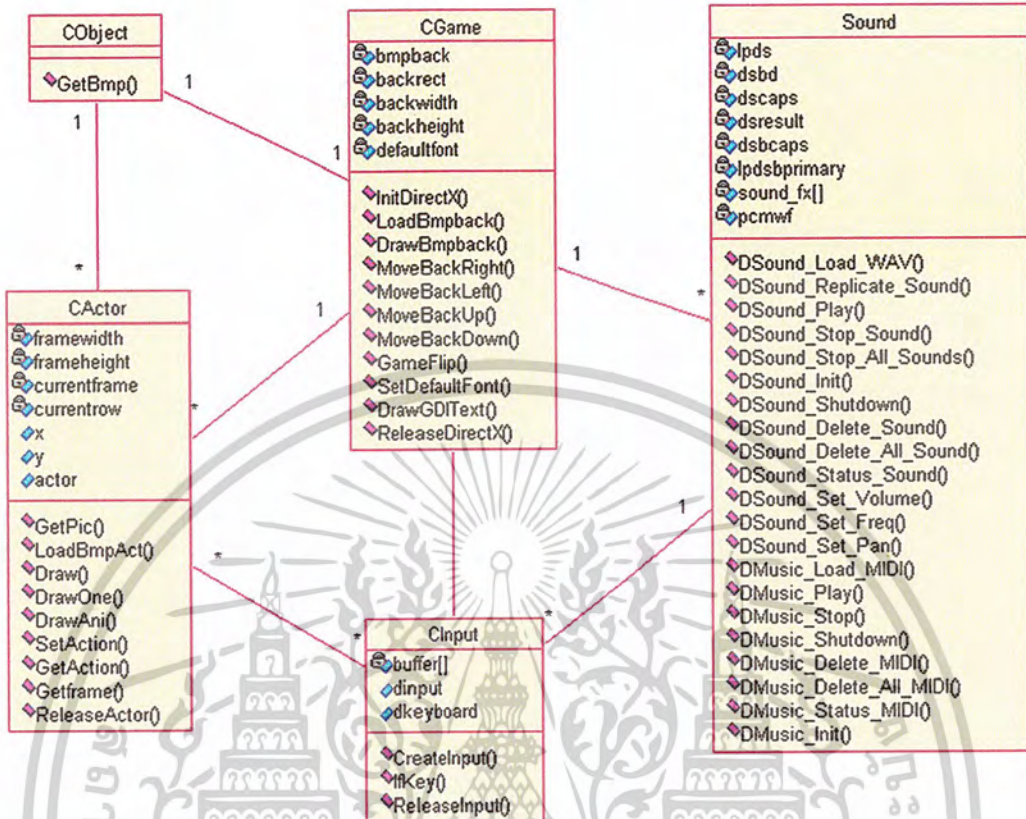
Activity Diagram ขั้นตอนdownloadตัวละคร

Activity Diagram ขั้นตอนuploadตัวละคร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.3 class diagram



คลาส CObject

จะทำหน้าที่ในการโหลดภาพเท่านั้น

คลาส CGame

จะทำหน้าที่ในการเข้าสู่โหมดของ DirectX และวาดภาพฉากหลัง, วาดตัวอักษรรวมถึงข้อความต่างๆได้

คลาส CActor

จะทำหน้าที่ในการโหลดตัวละคร หรือวัตถุใดๆที่อยู่ในเกม โดยเราสามารถตั้งให้ตัวละครหรือวัตถุแสดงได้ทั้งในแบบของอนิเมชันและแบบภาพนิ่ง

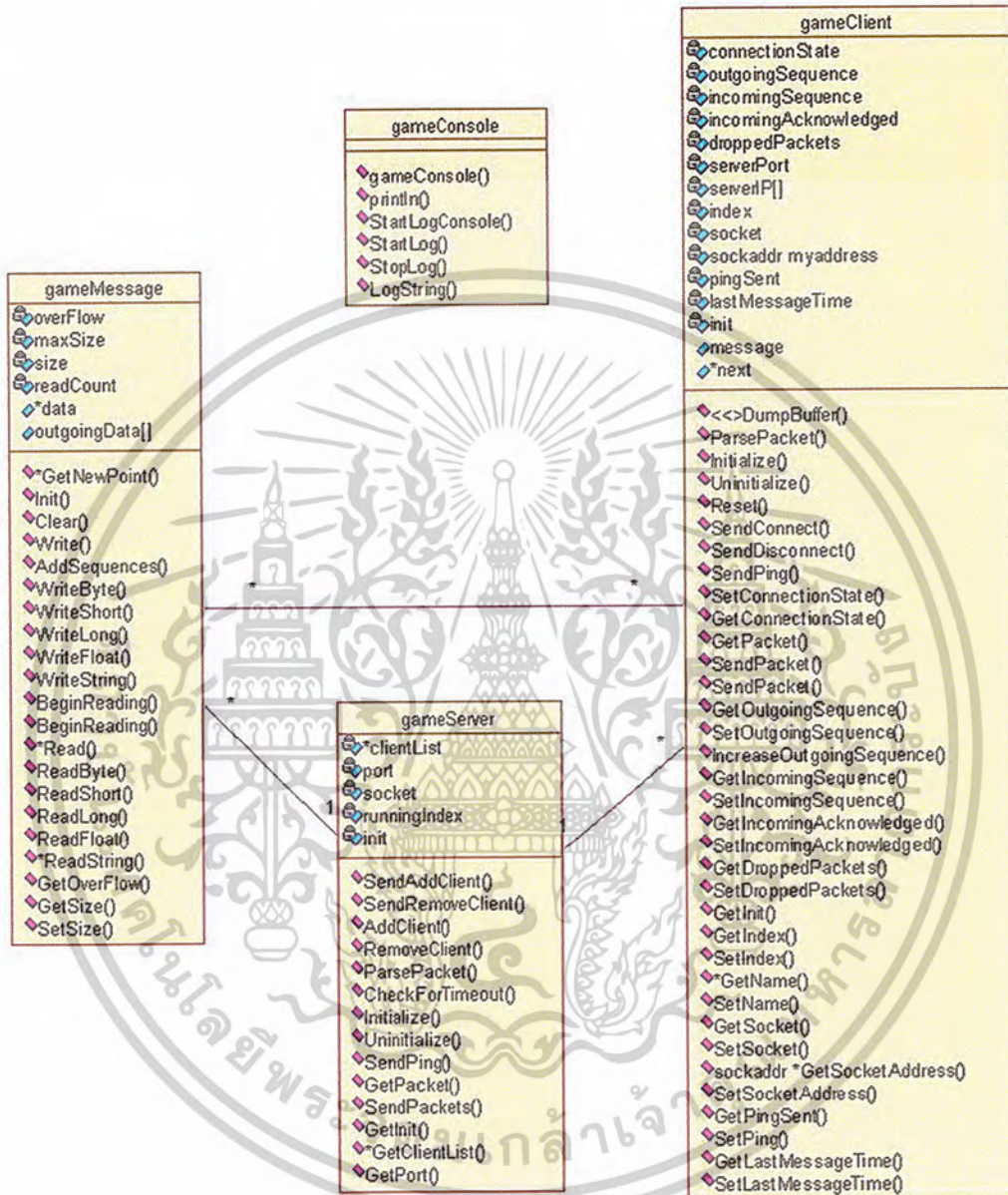
คลาส CInput

จะทำหน้าที่ในการจัดการกับ input ที่เข้ามา ซึ่งจะทำหน้าที่เก็บค่าของคีย์ที่ถูกกดว่าเป็นคีย์ใดจากนั้นก็ทำตามที่ได้ตั้งไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส Sound

จะเป็นคลาสที่ไว้จัดการกับเสียง โดยเราสามารถนำเสียงทั้งแบบ midi และ แบบ wave มาใช้ใน เกมได้ ซึ่งแต่ละแบบก็จะมีฟังก์ชันที่ไว้ใช้จัดการต่างกันออกไป



class gameMessage

ใช้สำหรับสร้าง message ที่จะส่งออกและทำการส่งต่อ message ที่ได้รับเข้า

class gameclient

คลาสนี้จะถูกใช้ทั้งบนฝั่ง เซิร์ฟเวอร์ และทางฝั่ง ไคลเอน ในส่วนของ เซิร์ฟเวอร์ นั้น คลาสนี้จะทำงานเปรียบเสมือน link list สำหรับเก็บ ไคลเอน ที่เข้ามาติดต่อกับ เซิร์ฟเวอร์ เมื่อ เซิร์ฟเวอร์ ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะส่งข้อมูลไปยัง ไคล์แอน เซิร์ฟเวอร์ ก็จะใช้ object ที่อยู่ใน list นี้ สำหรับในส่วนของ ไคล์แอน นั้น class นี้จะใช้สำหรับเก็บข้อมูลและ สภาพการติดต่อกับ เซิร์ฟเวอร์ เอาไว้

class gameserver

คลาสนี้ใช้สำหรับรองรับ การร้องขอการเชื่อมต่อจาก ไคล์แอน และทำการสื่อสารกับ ไคล์แอน ตลอดการเชื่อมต่อนั้น คลาสนี้จะมี link list สำหรับ ไคล์แอน ทุกคนที่เชื่อมต่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ลักษณะตัวเกม

4.1.1 ต่อสู้ออนไลน์



ลักษณะของหน้าจอการต่อสู้

รายละเอียดในเมนูบาร์ด้านล่างของหน้าจอมีดังนี้

- แถบสีฟ้า จะแสดงถึงข้อมูลของตัวละครผู้เล่นมีดังนี้
 - lv แสดงถึงระดับความสามารถของตัวละคร
 - HP แสดงพลังของตัวละครทั้งค่าปัจจุบันและค่าสูงสุด
 - SP แสดงพลังในการใช้ท่าของตัวละครทั้งค่าปัจจุบันและค่าสูงสุด
 - speed แสดงถึงความเร็วในการเข้าถึงเทิร์น
- แถบสีเหลือง จะแสดงถึงไอเทมที่สามารถใช้ในขณะต่อสู้ได้ ที่ตัวละครมีอยู่ทั้งหมด ซึ่งสามารถมีได้สูงสุด 3 อันต่อ 1 ครั้งการต่อสู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. แถบสีเขียว ที่อยู่ด้านล่างแถบ ไอเทมนั้นจะแสดงถึงเวลาที่เหลืออยู่ในแต่ละเทรนของผู้เล่น หากเวลานี้นับถอยหลังจนถึง 0 จะถือว่าผู้เล่นจบเทิร์นทันที
4. แถบสีแดง จะแสดงก็ต่อเมื่อผู้เล่นได้ทำการ โจมตี หรือทำการ ใช้ท่าเข้าโจมตีใส่ผู้เล่นอีกคนหนึ่ง โดยจะแสดง damage ที่จะเกิดขึ้นให้ผู้เล่นได้ทราบ
5. แถบสีม่วง ซึ่งอยู่ทางขวาสุดของจอ นั้น จะแสดงให้ทราบถึงค่า speed สะสม ของผู้เล่นแต่ละคนที่อยู่ในเกม หากค่า speed ของผู้เล่นคนใดเพิ่มจนครบ 100 เมื่อไหร่ก็จะถึงเทิร์นของผู้เล่นคนนั้น

4.1.2 เลี้ยงมอนเตอร์บนมือถือ

ลักษณะหน้าจอการเลี้ยง Monster บนมือถือ



รูปที่ 4-1 หน้าแรก

จากรูปที่ 4-1 เป็นหน้าเริ่มแรกของการเลือกเกม เมื่อผู้เล่นได้ทำการดาวน์โหลดเกมลงบนมือถือแล้ว ก็จะปรากฏตัว เกมขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-2

จากรูปที่ 4-2 เป็นหน้าจอแสดงรายการ โหลด ก่อนจะเข้าเล่นเกม



รูปที่ 4-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่4-3 เป็นหน้าจอ menu เพื่อให้ผู้เล่นได้ใช้ฟังก์ชันต่างตามที่กำหนดไว้



รูป4-4

จากรูปที่4-5 เป็นหน้าจอการเลือกMonster



รูปที่4-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้เล่นได้ทำการเลือกMonster แล้ว ก็จะเข้าสู่หน้าจอ การฝึกMonster ดังรูปที่4-5



รูปที่ 4-6

ถ้าผู้เล่นอยู่ในหน้าจอฝึกMonster แล้ว ได้มีการออกมาที่หน้าจอmenu จะมีการเก็บค่าสถานะต่างๆไว้เหมือนเดิม ซึ่งถ้าผู้เล่นจะเข้าไปฝึกMonsterอีก ก็สามารถเข้าไปที่ continue ได้เลย ดังรูปที่ 4-6



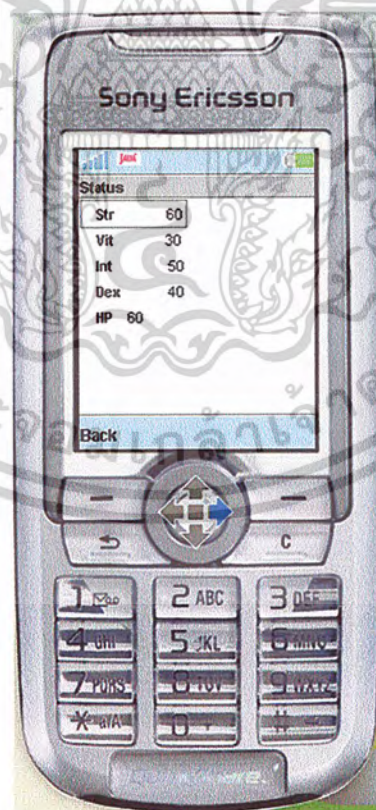
รูปที่4-7

เมื่อฝึก Monster นานๆ ก็จะต้องมีการเลี้ยงอาหาร Monster ซึ่งการให้อาหารMonsterนั้น ถ้ามีการให้ที่บ่อยเกินไปอาจจะทำให้ความสนิทสนมของMonsterที่มีต่อผู้เลี้ยงลดลงได้ แต่ถ้าเราขาดการดูแล ไม่ค่อยได้ให้อาหารMonster ก็จะทำให้ตายได้ ดังนั้นการให้อาหาร Monster ก็ต้องให้อย่างเหมาะสมด้วย



รูปที่ 4-8

จากรูปที่ 4-8 เป็นหน้าจอเลือกเกมฝึกมอนสเตอร์



รูปที่ 4-9

จากรูปที่ 4-9 เป็นหน้าจอบอกstatus ของมอนสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4-10

จากรูปที่4-10 เลือกคำสั่งUpload เพื่อ Upload มอนสเตอร์สู่เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 บทวิจารณ์และสรุป

สามารถนำเทคโนโลยีหลายๆ ชนิดมาประยุกต์ใช้ในการสร้างเกมได้อย่างประสบผลสำเร็จ ซึ่งในการเล่นแบบออนไลน์แล้ว และสามารถนำตัวละครของเราไปเลี้ยงลงในมือถือได้เป็นแนวความคิดใหม่ซึ่งยังไม่มีใครทำ

5.2 ปัญหาที่เกิดขึ้นในการพัฒนาและแนวทางในการแก้ปัญหา

1. เนื่องจากหากต้องการที่จะสร้าง sprite ของตัวละครเอง ให้เป็นไปตามที่ต้องการ อาจทำให้ sprite นั้นไม่สวยเท่าที่ควรเนื่องจาก ไม่มีฝ่ายออกแบบตัวละครโดยเฉพาะ ทำให้รูป sprite ต้องหามาเท่านั้น ทำให้ไม่ตรงกับที่ต้องการนัก
2. เช่นเดียวกันกับไฟล์เสียงที่ใช้ต้องไปค้นหามา ทำให้ได้ไม่ตรงกับที่ต้องการนัก
3. ความไม่เข้ากันของเวอร์ชันของซอฟต์แวร์ที่ใช้พัฒนา ทำให้เกิดปัญหาในด้าน โปรแกรมเกิดขึ้น

5.3 ข้อจำกัดของระบบ

1. ทรัพยากรบนมือถือที่มีอยู่อย่างจำกัด
2. tool ที่ใช้ sprite ต้องเป็นรูปสกุล .bmp ซึ่งขนาดของไฟล์ใหญ่มาก ทำให้เมื่อเวลาโหลดเข้าเกมต้องเสียเวลาโหลดเข้าเกมเป็นเวลานาน

5.4 แนวทางการพัฒนา

1. สามารถเพิ่มในส่วนของกราฟิกของตัวละครให้สามารถเปลี่ยนรูปร่างได้เมื่อมีระดับความสามารถที่มากขึ้นได้
2. สามารถเพิ่มไอเทมต่างๆ ทั้งไอเทมสวมใส่และไอเทมที่ใช้ในการต่อสู้ให้เพิ่มมากขึ้นได้
3. อาจเพิ่มฉากหลังในฉากการต่อสู้ให้เป็นแบบ 3 มิติซึ่งจะทำให้เกมดูมีมิติน่าเล่นมากยิ่งขึ้น
4. อาจทำให้เมื่อโหลดคอมพิวเตอร์ลงบนมือถือแล้วสามารถเล่นต่อสู้กับผู้เล่นคนอื่นผ่านระบบบลูทูธหรืออินเทอร์เน็ตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Andrew Mulholland and Teijo Hakala , “Programming Multiplayer Games ” , “2004 Worldware Publishing, Inc.”
- [2] สาธิต ชัยวิวัฒน์ตระกูล, “เก่ง JSP ให้ครบสูตร”, “ดวงกมลสมัย”, 2002, 1-464
- [3] กาญจนา ตันวิสุทธิ, “เขียนเกมและโปรแกรมบนมือถือ J2ME”, “ไอดีซีซี”, 2004, 1-288
- [4] นิรุช อำนวยศิลป์, “เขียนเกมอย่างมืออาชีพด้วย Visual C++ และ DirectX”, “Infopress”, 2003, 1-328
- [5] สุวัฒน์ ปุณณชัยยะ , คณิต ศุภกิจวงษ์, สุพจน์ ปุณณชัยยะ, “เปิดโลก TCP/IP และโปรโตคอลของอินเทอร์เน็ต”, “โปรวิชั่น”, 2003, 1-59
- [6] <http://java.sun.com/>
- [7] <http://www.gamedev.net/>
- [8] <http://www.thaigamedevx.com/>
- [9] <http://www.gamasutra.com/>
- [10] <http://www.midletzone.com/>
- [11] <http://www.narisa.com/>
- [12] <http://www.thaidev.com/>
- [13] <http://www.mysql.com/>
- [14] <http://www.developer.com/>
- [15] <http://www.pantip.com/tech/developer/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

การสร้างแพ็คเกจและจัดเตรียมแอปพลิเคชันสำหรับ MIDlet

การสร้างแพ็คเกจ และจัดเตรียมแอปพลิเคชัน MIDP ใน J2ME นั้น ประการแรกไฟล์คลาสทั้งหมดของแอปพลิเคชัน MIDlet จะต้องเก็บอยู่ในไฟล์ JAR เพียงไฟล์เดียว ทั้งนี้ไฟล์หนึ่งๆอาจมีแอปพลิเคชันได้มากกว่าหนึ่งแอปพลิเคชัน กลุ่มของแอปพลิเคชันนี้เรียกว่า “ชุดโปรแกรม MIDlet” (MIDlet suit)

ประการต่อมาไฟล์ JAR สามารถดาวน์โหลด และติดตั้งลงบนอุปกรณ์ไร้สายผ่านสายเคเบิล อุปกรณ์ที่ต่อกับเครื่องคอมพิวเตอร์ส่วนบุคคล หรือผ่านเครือข่ายไร้สายก็ได้ เมื่อติดตั้งไฟล์ JAR แล้ว ผู้ใช้งานสามารถเลือก และสั่งการกระทำแอปพลิเคชันจากเมนูเริ่มต้นได้

ไฟล์ที่เกี่ยวข้องในการสร้าง และจัดเตรียมแพ็คเกจของ MIDlet ได้แก่ไฟล์ manifest และไฟล์ application descriptor โดยนำไฟล์ manifest มาใช้อธิบายเนื้อหาของไฟล์ JAR ของชุดโปรแกรม MIDlet ส่วนไฟล์ application descriptor นำมาใช้เพื่ออธิบายเนื้อหาของรายการข้อมูล ตำแหน่งไฟล์ JAR ในชุดโปรแกรม MIDlet ฯลฯ

Manifest

ในชุด โปรแกรม MIDlet จะต้องมีไฟล์ manifest รวมอยู่ด้วยเสมอ เพื่ออธิบายเนื้อหาในไฟล์ JAR และข้อมูลอื่นๆเช่น ชื่อ รุ่นและผู้ผลิต โปรแกรม MIDlet

ไฟล์ mani ประกอบด้วยรายการแอตทริบิวต์ MIDlet ที่แสดงในลักษณะคู่ของชื่อ และค่าซึ่งคั่นกลางด้วยเครื่องหมาย : การใช้แอตทริบิวต์เหล่านี้จะช่วยให้นักพัฒนาสามารถอธิบายให้ผู้ใช้งานทราบว่า MIDlet ใดบ้างในชุดโปรแกรม MIDlet

ชื่อแอตทริบิวต์	รายละเอียด
MIDlet-Name	ชื่อของชุด โปรแกรม MIDlet
MIDlet-Version	รุ่นของชุด โปรแกรม MIDlet
MIDlet-Vendor	ผู้ผลิตชุด โปรแกรม MIDlet
MIDlet-Icon	ชื่อไฟล์ PNG ในไฟล์ JAR ใช้แทนชุดโปรแกรม MIDlet ไฟล์นี้จะใช้เมื่อซอฟต์แวร์จัดการแอปพลิเคชันแสดง ไอคอนเพื่อระบุชุดโปรแกรมบนหน้าจอ
MIDlet-Description	คำอธิบายรายละเอียดของชุด โปรแกรม MIDlet
MIDlet-Info-URL	URL สำหรับค้นคำอธิบายของชุด โปรแกรม MIDlet เพิ่มเติม
MIDlet-<n>	ชื่อ ไอคอน และคลาสของ MIDlet ลำดับที่ n ในไฟล์ JAR คั่นด้วยเครื่องหมาย , โดยค่าต่ำสุดของ n เริ่มจาก 1 และไล่ไปตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Name เพื่อใช้จำแนก MIDlet ให้ผู้ใช้งานทราบ Icon เป็นชื่อของคลาสที่ขยายคลาสของ MIDlet สำหรับ MIDlet ลำดับที่ n โดยคลาสจะต้องมี constructor แบบ no-args ที่ประกาศเป็นแบบ public
MIDlet-Jar-URL	URL ที่สามารถดาวน์โหลดไฟล์ JAR ได้
MIDlet-Data-Size	ขนาดขั้นต่ำของพื้นที่เก็บข้อมูลแบบคงตัว ที่ MIDlet ใช้ มีหน่วยเป็นไบต์ อุปกรณ์อาจจัดสรรเนื้อที่เพิ่มเติมให้ ทั้งนี้ขึ้นอยู่กับอุปกรณ์ ค่าโดยปริยายคือ 0
MicroEdition-Profile	โพรไฟล์ของ J2ME ที่ต้องการ ใช้ฟอร์แมต และค่าเดียวกับพรีอเพอร์ตี microedition.profile(เช่น MIDP-1.0)
MicroEdition-Configuration	คอนฟิกูเรชันของ J2ME ที่ต้องการ ใช้ฟอร์แมต และค่าเดียวกับพรีอเพอร์ตี microedition.configuration(เช่น CLDC-1.0)

ตาราง แอตทริบิวต์ต่างๆของไฟล์ manifest

ตัวอย่างไฟล์ manifest(MANIFEST.MF) ของโปรแกรม Hellow

MIDlet-Name : Hellow World

MIDlet-Vendor : Sam Publishing

MIDlet-Version : 1

MIDlet-1 : HellowWorld,/Icon.png,HellowWorld

MicroEdition-Configuration : CLDC-1.0

MIDlet-Data-Size : 0

ชื่อชุดโปรแกรม MIDlet คือ Hellow World ผู้ผลิตคือ Sam Publishing และรุ่นของแพ็คเกจคือ 1.0 มีเฉพาะแอตทริบิวต์ MIDlet-1 เนื่องจากมี MIDlet คเพียงตัวเดียวในชุดโปรแกรมนี้ หากมีมากกว่าหนึ่งตัว จะต้องใส่รายการ MIDlet-n เพิ่มเข้าไปโดย n ได้แก่ เลขลำดับต่อมาของโปรแกรม MIDlet อื่นๆ ในพิวต์ MIDlet-1 ค่าตัวแรกของ HellowWorld หมายถึง ชื่อของ MIDlet โดย/Icon.png คือ พารามิเตอร์ของไฟล์ไอคอน ค่าลำดับที่สองคือชื่อคลาสของ MIDlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างแพ็คเกจแอปพลิเคชัน MIDlet

ไฟล์ JAR ทั่วไปในชุดโปรแกรม MIDlet จะประกอบไปด้วยไฟล์คลาสทั้งหมดในแอปพลิเคชัน MIDlet ไฟล์ทรัพยากร และไฟล์ manifest สำหรับไฟล์คลาสจะต้องคอมไพล์ไครก่อน ไฟล์ทรัพยากรมักจะเป็นไฟล์ข้อมูล และรูปภาพต่างๆที่ MIDlet มาใช้ในขณะทำงาน

MIDlet ต่างๆในชุดโปรแกรม MIDlet อาจใช้ไฟล์คลาสร่วมกันได้ ทำให้คุณสามารถสร้างโปรแกรม MIDlet หลายๆโปรแกรม และเก็บไว้ในแพ็คเกจเดียวกัน โดยออกแบบให้ MIDlet ต่างๆในแอปพลิเคชันใช้ไฟล์คลาส และไฟล์ทรัพยากรร่วมกัน เพื่อลดขนาดของแพ็คเกจลงได้

ตัวอย่างการรวม MIDlet ต่างๆเข้าไว้ในไฟล์ JAR

```
C:>c:\jdk1.3\bin\jar cvmf MANIFEST.MF HelloWorld.jar Hello.class icon.png
```

Jar เป็นโปรแกรมยูทิลิตี้ที่มากับ JDK1.3 ตัวเลือก cvmf คำสั่งมีความหมายดังนี้ c หมายถึง การสร้างไฟล์ใหม่ v หมายถึง ให้แสดงข้อความออกแบบ ไม่ย่อ m หมายถึง ให้อ่านข้อมูล manifest และ f ระบุชื่อไฟล์ JAR ที่ต้องการ

Application Descriptor

ไฟล์ application descriptor จะมีฟอร์แมตเหมือนกับไฟล์ manifest ทว่ามีวัตถุประสงค์การใช้งานที่แตกต่างกัน คือ ไฟล์ manifest ใช้เพื่อสร้างแพ็คเกจของชุดโปรแกรม MIDlet ขณะที่ไฟล์ application descriptor ใช้เพื่อจัดเตรียมชุดโปรแกรม MIDlet โดยเฉพาะในกรณีที่ต้องการแจกจ่ายแอปพลิเคชันแบบไร้สาย หรือที่เรียกว่า Over The-Air(OTA)deployment

ก่อนที่จะดาวน์โหลดไฟล์ JAR ลงในอุปกรณ์ โปรแกรมจัดการแอปพลิเคชัน (AMS) จะอ่านไฟล์ application descriptor เพื่อตรวจสอบให้แน่ใจว่าชุดโปรแกรม MIDlet สามารถใช้ได้กับอุปกรณ์ที่จะติดตั้งแอปพลิเคชันนั้นๆ ซึ่งจะขึ้นกับเงื่อนไขในกรณีที่เครือข่ายไร้สายที่ใช้มีแบนด์วิดธ์ต่ำ หรือไม่เสถียรภาพ

แอตทริบิวต์ที่กำหนดไว้ล่วงหน้า

แอตทริบิวต์ที่จำเป็นต้องมีในไฟล์ application descriptor นั้นมี 7 แอตทริบิวต์คือ

MIDlet-Name

MIDlet-Version

MIDlet-Vendor

MIDlet-Jar-URL

MIDlet-Jar-Size

MicroEdition-Configuration

และอีก 4 แอตทริบิวต์ที่อาจมี หรือ ไม่มีก็ได้คือ

MIDlet-Description

MIDlet-Icon

MIDlet-Info-URL

MIDlet-Data-Size

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนจะดาวน์โหลดไฟล์ JAR ลงในอุปกรณ์ โปรแกรมจัดการแอปพลิเคชัน(AMS) จะตรวจสอบแอตทริบิวต์บังคับทั้ง 7 ในไฟล์ application descriptor เพื่อดูว่าแอปพลิเคชันเหมาะสมกับอุปกรณ์หรือไม่ เช่น ถ้าอุปกรณ์ไม่สนับสนุนไฟล์ หรือรุ่นของโปรไฟล์ที่กำหนดไว้ ก็จะไม่ดาวน์โหลดไฟล์ JAR มา และหากขนาดของไฟล์ JAR ที่ระบุในแอตทริบิวต์ MIDlet-Jar-Size มีขนาดใหญ่เกินไปสำหรับอุปกรณ์นั้นๆ ก็จะไม่ดาวน์โหลดไฟล์ JAR เช่นกัน

ถึงแม้ว่าไฟล์ application descriptor และไฟล์ manifest จะมีวัตถุประสงค์ต่างกัน ทว่าค่าแอตทริบิวต์ MIDlet-Name, MIDlet-Version, MIDlet-Vendor ในไฟล์ทั้งสองจะต้องเหมือนกัน มิฉะนั้นจะไม่สามารถดาวน์โหลดชุดโปรแกรม MIDlet ได้

การกำหนดแอตทริบิวต์ใหม่

นอกจากแอตทริบิวต์ที่กำหนดไว้แล้ว ยังสามารถกำหนดแอตทริบิวต์ขึ้นเพื่อใช้งานเองได้ในแอปพลิเคชัน โดยฟิลด์ที่ผู้ใช้กำหนดขึ้นเองจะเริ่มต้นด้วย MIDlet- ไม่ได้

ค่าแอตทริบิวต์ทั้งหมดในไฟล์ manifest และไฟล์ application descriptor จะถูกอ่าน โดยโปรแกรม MIDlet ขณะทำงาน เช่นค่าแอตทริบิวต์เฉพาะแอปพลิเคชัน(Language-Support Target-Device Display-Width และ Display-Height)จะถูกใช้เป็นพารามิเตอร์สำหรับคอนฟิกูเรชันค่า Display-Width และ Display-Height โปรแกรม MIDlet จะใช้ค่าพารามิเตอร์ที่อ่านได้ในขณะทำงานในการปรับตัวเองให้เข้ากับอุปกรณ์ปลายทางประเภทต่างๆ สิ่งที่ต้องคือสร้างไฟล์ application descriptor เฉพาะอุปกรณ์ สำหรับอุปกรณ์ปลายทางทุกประเภทเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้