สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เกมบนอุปกรณ์พกพาที่ใช้พิกัดทางภูมิศาสตร์

GPS – BASED MOBILE GAME

ณัฏฐ์ บุญญาภิสิทธิ์โสภา
ณัฐเดช ประจวบ

มส
ส163897
2550

เลขหมู่.....
เลขทะเบียน.....83121
วัน,เดือน,ปี.....-5 ส.ค. 2551

b.....11959393
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เกมบนอุปกรณ์พกพาที่ใช้พิกัดทางภูมิศาสตร์

GPS – Based Mobile Game

ผู้จัดทำ

1. นายณัฏฐ์ บุญญาภิสิทธิ์โสภา รหัสประจำตัว 47010213
2. นายณัฐเดช ประจวบ รหัสประจำตัว 47010219

ชุติเมษฎ์ ศรีนิลทา อาจารย์ที่ปรึกษา

(ดร. ชุติเมษฎ์ ศรีนิลทา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกมบนอุปกรณ์พกพาที่ใช้ข้อมูลพิกัดทางภูมิศาสตร์

นายณัฏฐ์ บุญญาภิสิทธิ์โสภา 47010213
นายณัฐเดช ประจวบ 47010219
ดร. ชุติเมษฎ์ ศรีนิลทา อาจารย์ที่ปรึกษา
ปีการศึกษา 2550

บทคัดย่อ

โครงงานชิ้นนี้เป็นการประยุกต์นำเอาระบบสารสนเทศภูมิศาสตร์ มาออกแบบและพัฒนาเป็นโปรแกรมเกมเพื่อความบันเทิง โดยพัฒนาลงบนอุปกรณ์พกพาประเภทพอกเกตพีซี หรือ พีดีเอ ที่มีการเชื่อมต่อกับตัวรับข้อมูลจีพีเอส เพื่อนำเอาข้อมูลตำแหน่งทางภูมิศาสตร์มาประยุกต์ใช้เป็นส่วนหนึ่งของการดำเนินกิจกรรมภายในเกม โดยตัวเกมจะมีการเชื่อมต่อกับระบบเครือข่ายและฐานข้อมูล ให้บริการข้อมูลแก่ผู้เล่นเกมให้สามารถร่วมทำกิจกรรมภายในเกมได้พร้อมกันหลายๆคน ทั้งนี้ในการศึกษาและพัฒนาโครงงาน ได้มีการใช้ทฤษฎีต่างๆอันได้แก่ เทคโนโลยี Java ME การแสดงแผนที่ SVG ด้วยเทคโนโลยี SVG Tinyline และการติดต่อกับแอพพลิเคชันเซิร์ฟเวอร์ด้วย Java Servlet สำหรับการเล่นเกมแบบออนไลน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GPS – Based Mobile Game

Mr. Nat Bunyapisitsopa 47010213

Mr. Nattadate Prachuab 47010219

Ms. Chutimet Srinilta Advisor

Academic Year 2007

ABSTRACT

The propose of this project was to design and develop entertainment game application that use geographic data to implement game's event and game play. This game run on mobile device, Pocket PC with GPS receiver connection current geographic position optained from GPS receiver represents game player position. Actual map in SVG format was used in the game. Playing game involved physical position of players and other objects such as buildings and ATM machines. The client side of the game was developed with Java ME. SVG Tinyline was used to display map on mobile device in SVG format. The server side employed Java Servlet and postgres database.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำปรึกษาแนะนำจาก ดร.ชุติเมษฎ์ ศรีนิลทา ซึ่งเป็นอาจารย์ที่ปรึกษา คอยให้ความช่วยเหลือต่าง ๆ จนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ซึ่งขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่อำนวยความสะดวกต่าง ๆ ทั้งในเรื่องของสถานที่วิจัย และอินเทอร์เน็ต ทำให้สามารถทำการวิจัยและการพัฒนา ได้อย่างต่อเนื่องและรวดเร็ว ขอขอบคุณพี่ๆ เพื่อนๆ ทุกคนในห้องวิจัยที่คอยให้คำแนะนำ ความช่วยเหลือต่างๆ และกำลังใจที่ทำให้สามารถทำงานได้อย่างราบรื่น

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่คอยให้การสนับสนุน ให้กำลังใจ ข้าพเจ้าอย่างเสมอมา และที่สำคัญได้ให้โอกาสให้ข้าพเจ้าเข้าการศึกษาเล่าเรียน ณ สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบังแห่งนี้ ข้าพเจ้าขอจึงกราบขอบพระคุณมา ณ ที่นี้ด้วย

ณัฏฐ์ บุญญาภิสิทธิ์โสภา
ณัฐเดช ประจวบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญรูป VIII

สารบัญตาราง XI

บทที่ 1 บทนำ 1

1.1 ความเป็นมาของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 2

1.4 ขั้นตอนการดำเนินงาน 2

บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในโครงงาน 3

2.1 การแสดงผลแผนที่ 3

2.2 ความรู้เบื้องต้นเกี่ยวกับ SVG 4

2.2.1 SVG (Scalable Vector Graphic) 4

2.2.2 รูปแบบการเขียน SVG 5

2.2.3 Basic Shape SVG 6

2.2.4 ประโยชน์ในการใช้ SVG 8

2.3 Global Positioning System 8

2.3.1 องค์ประกอบของระบบดาวเทียม GPS 8

2.3.2 หลักการหาตำแหน่งของระบบ GPS 10

2.3.3 สาเหตุความคลาดเคลื่อนที่เกิดขึ้นในระบบ GPS 12

2.3.4 ความคลาดเคลื่อนเนื่องจากการวางตัวของดาวเทียม (Dilution of Precision, DOP) 15

2.4 ข้อมูล NMEA และการวิเคราห์ข้อมูล 16

บทที่ 3 เทคโนโลยีที่เกี่ยวข้อง 19

3.1 เทคโนโลยี Java ME 19

3.1.1 สถาปัตยกรรม Java ME 19

3.1.2 คอนฟิกูเรชันใน Java ME 20

3.1.3 Mobile Information Device Profile 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.1.4 Personal Digital Assistant Profile 24
3.1.5 TinyLine SVG 25
3.1.6 คุณสมบัติพื้นฐานของ TinyLine 25
3.1.7 แพ็คเกจต่างๆ ของ TinyLine 25
3.1.8 รายละเอียดของคลาสในแพ็คเกจทั้งสองและอินเทอร์เฟส 25
3.1.9 ขั้นตอนการทำงานของ TinyLine 29
3.2 การเขียนโปรแกรมรับค่าพิกัดจีพีเอสด้วย Java ME 31
3.2.1 การสถาปนาการเชื่อมต่อ 31
3.2.2 การอ่านข้อมูล 32
3.3 เทคโนโลยี Java Servlet 33
3.3.1 ขั้นตอนการทำงานของ Servlet Server 33
3.3.2 เปรียบเทียบระหว่างเทคโนโลยี Servlets กับเทคโนโลยีอื่น 33
3.3.3 Servlet Life Cycle 34
3.3.4 Apache Tomcat 35
3.4 PostgreSQL 35
3.4.1 PostgreSQL คืออะไร 35
3.5 สถาปัตยกรรมการทำงานของ PostgreSQL 36
3.6 การใช้เทคโนโลยี Java ติดต่อกับฐานข้อมูล 39
บทที่ 4 การออกแบบระบบ 41
4.1 การออกแบบเกม 41
4.1.1 ออกแบบเนื้อเรื่องของเกม 41
4.1.2 วัตถุประสงค์ของเกม 41
4.1.3 มุมมองภายในเกม 41
4.1.4 กฎ กติกา และการเล่นเกม 41
4.1.5 รายละเอียดของวัตถุในเกม 42
4.1.6 การออกแบบกิจกรรมต่างๆในเกม 44
4.2 การออกแบบระบบ 46
4.2.1 การออกแบบสถาปัตยกรรมของระบบ 46
4.2.2 สถาปัตยกรรมของ Client 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

4.3 การวิเคราะห์ระบบ 49
4.3.1 Use Case Diagram 49
4.3.2 Design Class Diagram 53
4.3.3 รายละเอียดของตารางข้อมูลต่างๆ 55
4.3.4 การออกแบบลำดับการทำงานของระบบ 59
4.4 การออกแบบการทำงานส่วนเซิร์ฟเวอร์ 62
4.4.1 Class diagram ของส่วนเซิร์ฟเวอร์ 62
บทที่ 5 การทดลองและสรุปผลการทดลอง 74
5.1 อินเตอร์เฟสพื้นฐานภายในเกม 74
5.1.1 หน้าจอเมนูตอนเริ่มเกม 74
5.1.2 หน้าจอหลักภายในเกม 75
5.1.3 สัญลักษณ์ต่างๆภายในเกม 75
5.1.4 วิธีการซูมเข้าและซูมออกแผนที่ภายในเกม 75
5.2 การทดลองการเคลื่อนที่ด้วยจีพีเอส 76
5.3 การทดลองระบบการลงทะเบียนเกม 79
5.3.1 ขั้นตอนการลงทะเบียนสำหรับผู้เล่น 79
5.4 การทดลองระบบการ Sign In 80
5.4.1 ขั้นตอนการ Sign In สำหรับผู้เล่น 80
5.5 การทดลองระบบต่อสู้ 82
5.6 ทดลองระบบการซื้อขายทรัพยากรภายในเมือง 83
5.6.1 ขั้นตอนการซื้อขายทรัพยากรภายในเมือง 83
5.7 ทดลองระบบการซ่อมแซมเรือ 87
5.8 ทดลองระบบการอัพเกรดเรือและอาวุธ 88
5.8.1 ขั้นตอนการอัพเกรดเรือและอาวุธ 88
บทที่ 6 บทวิจารณ์และสรุป 91
6.1 สิ่งที่ได้จากโครงงาน 91
6.2 ปัญหาและอุปสรรคที่พบขณะดำเนินโครงงาน 91
6.3 ข้อเสนอแนะ 92
6.4 แนวทางการพัฒนาต่อ 92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บรรณานุกรม 913

ภาคผนวก ก 914

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

2.1 ลักษณะของข้อมูลประเภทราสเตอร์ (Raster) 3

2.2 ภาพดาวเทียม (Remote Sensing) เป็นข้อมูลประเภทราสเตอร์ (Raster) 4

2.3 ขั้นตอนการแสดงผลใน SVG 5

2.4 องค์ประกอบของระบบดาวเทียม GPS 9

2.5 แสดงส่วนควบคุม 9

2.6 แสดงส่วนอวกาศ 10

2.7 Ranging Intersection 11

2.8 การเปรียบเทียบรหัสเพื่อหาค่าต่างเวลา 11

2.9 ใช้ดาวเทียม 4 ดวงเพื่อหาค่า Latitude Longitude Height และ Time 12

2.10 แสดงชั้นบรรยากาศไอโอโนสเฟียร์ 12

2.11 แสดงผลกระทบที่เกิดจากชั้นบรรยากาศไปโอโนสเฟียร์ 13

2.12 แสดงชั้นบรรยากาศโทรโพสเฟียร์ 14

2.13 การเกิด Multipath 14

2.14 การวางตัวของดาวเทียมที่ให้ค่า DOP ไม่ดี 15

2.15 การวางตัวของดาวเทียมที่ให้ค่า DOP ดี 15

3.1 แสดงโครงสร้างของ Java เทคโนโลยี 19

3.2 แสดงความสัมพันธ์ระหว่าง Java SE และ คลาสไลบรารี่ของ CDC และ CLDC 22

3.3 แสดงขั้นตอนการทำงานของ TinyLine 29

3.4 แสดง Client Process ขอบริการ Postmaster 37

3.5 แสดง Postmaster สร้าง Postgres Process 38

3.6 แสดง Postgres Process ให้บริการแก่ Client Process 38

3.7 แสดงการขอบริการเซิร์ฟเวอร์ผ่านอินเตอร์เฟส JDBC,ODBC 38

4.1 แสดงสถาปัตยกรรมของระบบ 46

4.2 แสดงโครงสร้างของฝั่ง Client 47

4.3 แสดง Use Case Diagram 49

4.4 แสดง Class Diagram ของ Client 53

4.5 แสดง ER Diagram ของระบบ 55

4.6 แสดง Sequenial Diagram ของการระบุตำแหน่ง 59

4.7 แสดง Sequenial Diagram ของการเดินเรือ 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ **หน้า**

4.8 แสดง Sequenial Diagram ของการโจมตีเรือ 61

4.9 แสดง class diagram ของ Servlet 62

4.10 แสดงรายละเอียดของคลาส database 63

4.11 แสดงรายละเอียดของคลาส updateDataThread 64

4.12 แสดงรายละเอียดของคลาส updateDataByTime 65

4.13 แสดงรายละเอียดของคลาส playerProcess 66

4.14 แสดงรายละเอียดของคลาส msgProcess 71

5.1 แสดงหน้าจอแสดงเมนูตอนเริ่มเกม 74

5.2 แสดงหน้าจอหลักของตัวเกม 75

5.3 แสดงจุดเข้าสู่เกมบริเวณอาคารปฏิบัติการ Electronic 76

5.4 แสดงปลายทางการเดินทดสอบบริเวณตึก ECC 77

5.5 แสดงหน้าจอแสดงผลภายในเกม ณ จุดเริ่มการทดสอบ 77

5.6 แสดงหน้าฐานข้อมูลตำแหน่ง ณ จุดเริ่มการทดสอบ 78

5.7 แสดงการเปลี่ยนแปลงหน้าจอการทดสอบการเคลื่อนที่ 78

5.8 แสดงการเปลี่ยนแปลงฐานข้อมูลตำแหน่งของการทดสอบการเคลื่อนที่ 78

5.9 แสดงขั้นตอนการลงทะเบียน 79

5.10 แสดงข้อมูลของผู้เล่นที่สร้างขึ้นในตาราง player 79

5.11 แสดงข้อมูลเรือของผู้เล่นที่สร้างขึ้นในตาราง ship 80

5.12 แสดงข้อมูลที่ถูกสร้างขึ้นในตาราง playership 80

5.13 แสดงขั้นตอนการ Sign In สำหรับผู้เล่น 81

5.14 แสดงหน้าจอสำหรับกรอก username และ password 81

5.15 แสดงหน้าต่อสู้ระหว่างเรือ 82

5.16 แสดงการต่อสู้ระหว่างเรือ 82

5.17 แสดงข้อมูลเรือทั้งสองลำบนตาราง ship 83

5.18 แสดงการเข้าสู่เมือง 83

5.19 แสดง attribute ของตาราง ship ก่อนการเข้าเมือง 84

5.20 แสดง attribute ของตาราง player ก่อนการเข้าเมือง 84

5.21 แสดงหน้าจอภายในเมือง 84

5.22 แสดงหน้าตาของหน้าซื้อขายทรัพยากร 85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ **หน้า**

5.23 แสดงตาราง town ในขณะที่ผู้เล่นเข้าเมือง 85

5.24 แสดงหน้าจอการซื้อ(รูปซ้าย) และขาย(รูปขวา) 85

5.25 แสดง attribute ของตาราง player หลังการทดลองซื้อ 86

5.26 แสดง attribute ของตาราง ship หลังการทดลองซื้อ 86

5.27 แสดง attribute ของตาราง town หลังการทดลองซื้อ 86

5.28 แสดง attribute ของตาราง player หลังการทดลองขาย 86

5.29 แสดง attribute ของตาราง ship หลังการทดลองขาย 87

5.30 แสดง attribute ของตาราง town หลังการทดลองขาย 87

5.31 แสดงหน้าจอแสดงเมนูภายในเมือง 88

5.32 แสดงเมนูภายในเมือง 88

5.33 แสดงหน้าจอสำหรับอัพเกรดเรือและอาวุธ 89

5.34 แสดงAttribute ต่างๆในตาราง player ก่อนการอัพเกรด 89

5.35 แสดงAttribute ต่างๆในตาราง ship ก่อนการอัพเกรด 89

5.36 แสดง Attribute ต่างๆในตาราง weapon ก่อนการอัพเกรด 89

5.37 แสดง Attribute ต่างๆของตาราง player หลังการอัพเกรด 90

5.38 แสดงAttribute ต่างๆของตาราง ship หลังการอัพเกรด 90

ก.1 แสดง NetBean IDE 6.0 เวอร์ชันล่าสุดที่ในปัจจุบัน 94

ก.2 แสดงการสร้างโปรเจกใหม่ 95

ก.3 แสดงการสร้างโปรเจกชนิด MIDP Application 95

ก.4 แสดงการตั้งชื่อโปรเจก และสิ้นสุดการสร้างโปรเจกใหม่ 96

ก.5 แสดงการเลือกใช้ Emulator ในการทดสอบการทำงาน 96

ก.6 แสดงการสร้าง Virtual Midlet 97

ก.7 แสดงการออกแบบ Midlet ด้วยการใช้ Virtual Midlet 98

ก.8 แสดงการออกแบบองค์ประกอบต่างๆของ Form ด้วย Virtual Midlet 98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง **หน้า**

2.1 คำอธิบายและสัญลักษณ์ที่ใช้ของแท็ก circle 7
2.2 คำอธิบายและสัญลักษณ์ที่ใช้ของแท็ก path 7
3.1 แสดงคลาสต่าง ๆ ใน tinyline2d 26
3.2 แสดงคลาสในการกำหนด basic data types, data structure และ utilitie 26
3.3 คลาสที่เป็นอินเทอร์เฟสของแพ็คเกจ com.tinyline.svg 27
3.4 คลาสที่ใช้จัดการโปรเซสการเฟสของ SVG Tiny 27
3.5 คลาสที่ในส่วนสุดท้ายที่ใช้ติดต่อกับ SVGT element 28
3.6 แสดงค่า Parameter ของ URL String ที่ใช้เปิดการเชื่อต่อ Serial Port 32
4.1 แสดงค่าพารามิเตอร์ของเรือ 42
4.2 แสดงค่าพารามิเตอร์ของเมือง 43
4.3 แสดงค่าพารามิเตอร์ของอาวุธ 43
4.4 แสดงส่วนประกอบของตาราง ship 56
4.5 แสดงส่วนประกอบของตาราง playership 56
4.6 แสดงส่วนประกอบของตาราง ship 56
4.7 แสดงส่วนประกอบของตาราง shipweapon 57
4.8 แสดงส่วนประกอบของตาราง ship 57
4.9 แสดงส่วนประกอบของตาราง township 57
4.10 แสดงส่วนประกอบของตาราง townweapon 57
4.11 แสดงส่วนประกอบของตาราง weapon 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงาน

ในปัจจุบัน ตลาดของอุตสาหกรรมเกมได้ขยายใหญ่ขึ้น และด้วยเทคโนโลยีอุปกรณ์พกพาที่มีการพัฒนาอยู่ตลอดเวลา ไม่ว่าในแง่ของประสิทธิภาพของการประมวลผล หรือ การเชื่อมต่อสื่อสารกับระบบเทคโนโลยีการสนเทศต่างๆอย่างกว้างขวางโครงงานชิ้นนี้จึงเป็นแนวทางหนึ่งในการที่จะพัฒนาโปรแกรมเกมเพื่อความบันเทิง โดยเป็นโปรแกรมที่สามารถรองรับการก้าวไปของเทคโนโลยีอุปกรณ์พกพา มีความสามารถในการรับข้อมูลจากระบบสารสนเทศทางภูมิศาสตร์ และการระบุพิกัดตำแหน่งปัจจุบันของผู้เล่น ให้ผู้เล่นสามารถตอบสนองกับตัวโปรแกรมเกมด้วยการเคลื่อนที่ หรือการเดินทางไปยังสถานที่ต่างๆ เพื่อทำกิจกรรมที่มีการ โต้ตอบกันระหว่างผู้เล่นกับข้อมูลเกม หรือกับระหว่างผู้เล่นด้วยกันเอง

โครงงานชิ้นนี้ได้นำเสนอทฤษฎีการออกแบบ และพัฒนาเกมบนอุปกรณ์พกพา ที่มีการประยุกต์ใช้ระบบสารสนเทศทางภูมิศาสตร์ และการระบุตำแหน่งด้วยจีพีเอสเข้าด้วยกัน

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาการเขียนแอพพลิเคชันบนอุปกรณ์พกพา ศึกษาทางเลือกและข้อจำกัด ของการใช้ Java ME สำหรับพัฒนาแอพพลิเคชันที่ใช้ resource ของอุปกรณ์อย่างมาก

1.2.2 เพื่อศึกษาการทำงานของระบบสารสนเทศภูมิศาสตร์ และการประยุกต์ใช้ระบบสารสนเทศภูมิศาสตร์กับอุปกรณ์พกพา

1.2.3 เพื่อศึกษาการทำงานและสร้างแอพพลิเคชันที่มีการประยุกต์ใช้จีพีเอส กับเทคโนโลยี Java ME

1.2.4 เพื่อสร้างแอพพลิเคชันเพื่อความบันเทิงประเภทเกมบนอุปกรณ์พกพาในรูปแบบใหม่ที่ ให้ผู้เล่นสามารถตอบสนองกับตัวเกมด้วยการเคลื่อนที่ไปยังสถานที่ต่างๆ โดยอ้างอิงจากความเป็นจริง

1.2.6 เพื่อศึกษาการทำงานกับแผนที่ SVG และการใช้งาน SVG Tinyline บน Java ME

1.2.7 เพื่อศึกษาการสร้างแอพพลิเคชันบนอุปกรณ์พกพาให้สามารถทำงานกับระบบเครือข่าย และติดต่อกับเซิร์ฟเวอร์เพื่อร้องขอบริการและข้อมูลต่างๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

1.3.1 ออกแบบและพัฒนาเกมโดยใช้ Java ME ที่ทำงานบนพอกเกตพีซี

1.3.2 ศึกษาทฤษฎีที่เกี่ยวข้องกับการประยุกต์ใช้ระบบสารสนเทศภูมิศาสตร์กับอุปกรณ์พกพา

1.3.3 ศึกษาการเชื่อมต่อกันระหว่าง Java ME กับระบบฐานข้อมูลบน Servlet

1.3.4 พัฒนาเกมให้สามารถเล่นได้ในบริเวณของสถาบัน

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาทางเลือกในการพัฒนาโปรแกรมบนอุปกรณ์พกพา

1.4.2 ศึกษาการแสดงภาพ SVG บน Java ME

1.4.3 ศึกษาวิธีการติดต่อกับตัวรับข้อมูลจีพีเอส

1.4.4 ศึกษาการใช้ Servlet ติดต่อกับฐานข้อมูล

1.4.5 ออกแบบและพัฒนาเกมบนอุปกรณ์พกพา

1.4.6 ออกแบบและพัฒนาเซิร์ฟเวอร์ และฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในโครงงาน

2.1 การแสดงผลแผนที่

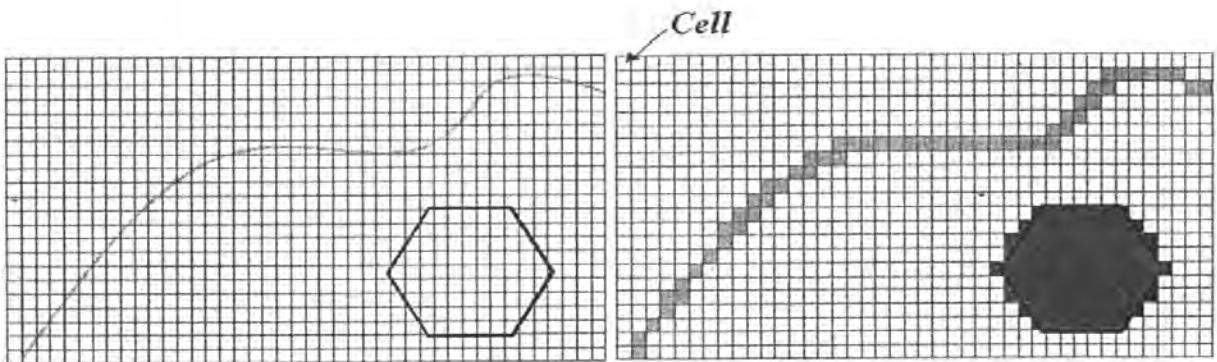
แบ่งออกได้เป็น 2 ประเภทคือ Vector Graphics และ Raster Graphics

1. Vector Graphics

ภาพกราฟิคแบบเวกเตอร์นั้น ถูกสร้างขึ้นด้วยเส้นตรงและเส้นโค้ง ที่อาศัยสมการคณิตศาสตร์ กำหนดรูปร่าง ขนาด และ ความสัมพันธ์กับส่วนอื่นๆ ของภาพ ดังนั้นจุดแต่ละจุด ในภาพแบบเวกเตอร์ จึงสัมพันธ์กันอย่างเป็นระบบ ภาพแบบนี้จึงสามารถย่อขยายได้ โดยไม่ทำให้คุณภาพของภาพนั้นด้อยลง (scalable) รายละเอียดของภาพนั้นจะคงเดิมอยู่ตลอด ไม่ขึ้นกับขนาดของภาพ ฟอร์แมทของภาพเหล่านี้ได้แก่ Scalable Vector Graphic (SVG) Postscript (EPS) Windows Meta File (WMF) AutoCAD drawing (DXF) และ Micrografx Designer Drawing (DRW) ซึ่งในเอกสารนี้จะเน้นภาพกราฟิคแบบเวกเตอร์แบบ Scalable Vector Graphic ซึ่งจะมีรายละเอียดในหัวข้อถัดไป

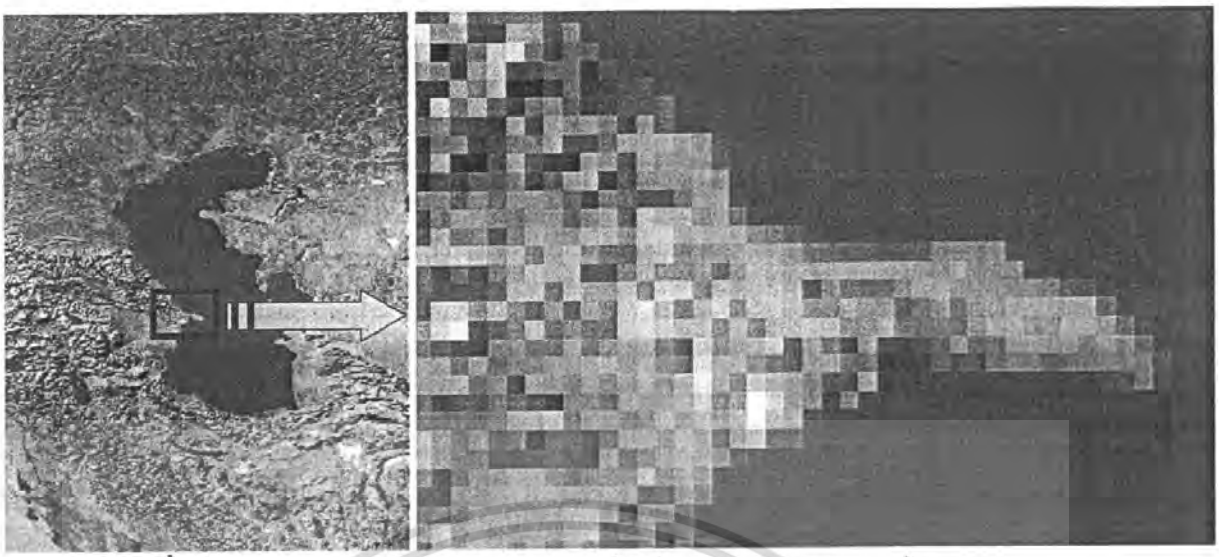
2. Raster Graphics

ภาพกราฟิคแบบราสเตอร์หรือที่รู้จักในอีกคำหนึ่งว่า Bipmapped Graphics สร้างขึ้นโดยการวางจุดสีลงไปบนภาพตรงๆ โดยแต่ละจุด ไม่มีความเกี่ยวข้องสัมพันธ์ใดใด ยกเว้นในเรื่องของลำดับเท่านั้น (เลยเรียกว่าบิตแมพ หรือแผนที่บิต) ภาพที่สร้างขึ้นมา โดยวิธีการแบบบิตแมพ จะมีรายละเอียดที่เปลี่ยนแปลง ขึ้นกับความสามารถของ การแสดงผลกราฟิค หากมีการย่อหรือขยาย ซึ่งจะมีการเพิ่มหรือลดจำนวนจุด ก็จะทำให้คุณภาพของภาพนั้นสูญเสียไป ทั้งนี้เพราะจุดต่างๆ ที่ประกอบเป็นภาพ ไม่ได้มีส่วนรับรู้ถึงความหมายของภาพ ซึ่งต่างจากภาพแบบเวกเตอร์ ที่มีองค์ประกอบคือ เส้นตรงและเส้นโค้ง ฟอร์แมทของภาพแบบบิตแมพได้แก่ PCX JPEG (.JPG) TIFF GIF Windows Bitmap (BMP) เป็นต้น



รูปที่ 2.1 ลักษณะของข้อมูลประเภทราสเตอร์ (Raster)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ภาพดาวเทียม (Remote Sensing) เป็นข้อมูลประเภทราสเตอร์ (Raster)

2.2 ความรู้เบื้องต้นเกี่ยวกับ SVG

2.2.1 SVG (Scalable Vector Graphic)

ภาษา SVG เป็นภาษาที่ถูกสร้างขึ้นมาโดยมีพื้นฐานแนวคิดมาจากภาษา XML (Extensible Markup Language) ซึ่ง SVG นั้นเป็นมาตรฐานของภาษาที่ใช้แสดงผลข้อมูลเวกเตอร์กราฟิก แต่ก็สามารถแสดงผลราสเตอร์ได้ด้วย มาตรฐานดังกล่าวนี้ออกแบบและเผยแพร่โดยองค์กรที่มีชื่อว่า World Wide Web Consortium หรือ W3C ซึ่งองค์กรนี้เป็นองค์กรระดับโลก มีหน้าที่ในการกำหนดมาตรฐานต่าง ๆ ที่เกี่ยวกับการทำงานบนเว็บที่เรารู้จักกันดีเช่น HTML, XML, CSS, Web Service ฯลฯ

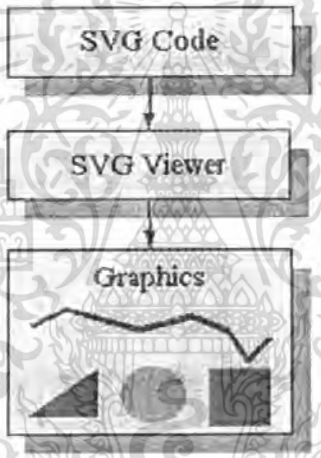
SVG นั้นถูกสร้างขึ้นโดยกลุ่มขององค์กรหลายองค์กรซึ่งส่วนใหญ่เป็นบริษัทเอกชนที่มีธุรกิจเกี่ยวเนื่องกับเว็บมารวมตัวกัน ภายใต้ W3C เช่น Nokia, Ericsson AB,Microsoft Corporation, Corel Corporation, Autodesk Inc., Adobe System Inc., America Online Inc., IBM, Sun Microsystems, Sharp Corporation, Macromedia Inc. เป็นต้น จะเห็นว่ามาตรฐานภาษา SVG ที่ออกโดย W3C มีความน่าเชื่อถือสูง เนื่องจากได้รับการพิจารณาโดยทีมงานที่เป็นผู้เกี่ยวข้องหลักและร่วมกันคิดร่วมกันออกแบบอย่างดี SVG จึงมีแนวโน้มสูงที่จะเป็นมาตรฐานหลักในการนำเสนอข้อมูลกราฟิคบนเว็บต่อไป

ข้อมูลส่วนใหญ่ที่มนุษย์เราเกี่ยวข้องด้วย มักจะมีความเชื่อมโยงกับข้อมูลเชิงตำแหน่งเสมอ เช่น ข้อมูลโรงเรียน โรงพยาบาลซึ่งสัมพันธ์กับตำแหน่งที่ตั้ง ข้อมูล สถานที่ท่องเที่ยว หรือประชากร ซึ่งมีการกระจายตัวสัมพันธ์กับลักษณะภูมิศาสตร์ ข้อมูลยอดขายสินค้ากับลักษณะการกระจายตัวของจุดกระจายสินค้าและปริมาณความหนาแน่นของเที่ยวรถบรรทุกสินค้าในแต่ละเส้นทาง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SVG สามารถนำมาใช้ช่วยแสดงผลกราฟิกข้อมูลเชิงตำแหน่งที่ต้องการมีปฏิสัมพันธ์กับผู้ใช้สูง รองรับการประมวลผลบนฝั่งเครื่องผู้ใช้ได้มากกว่าแผนที่ที่เป็นรูปภาพธรรมดา พร้อมทั้งแสดงผลร่วมกับสารสนเทศอื่นได้อย่างมีประสิทธิภาพการแสดงผลหรือวิเคราะห์ร่วมกันระหว่างสารสนเทศทั้งสองแบบจะช่วยให้ระบบสารสนเทศใหม่มีความสามารถตอบสนองความต้องการผู้ใช้งานได้กว้างและลึกขึ้น

เมื่อเรามีภาษาแล้วก็แน่นอนว่าเราต้องมีเครื่องมือหรือโปรแกรมที่ทำหน้าที่อ่านภาษา SVGแล้วแปลออกมาเป็นเวกเตอร์กราฟฟิก ตามเจตนารมณ์ที่ SVG ถูกออกแบบมา โปรแกรมที่สามารถเข้าใจ และแสดงผลข้อมูล SVG ให้อยู่ในรูปกราฟฟิกได้เรียกว่า โปรแกรม SVG Viewer ในกรณีที่ผู้ใช้ต้องการให้เว็บบราวเซอร์ดูข้อมูลกราฟิกที่อยู่ในรูป SVG ได้นั้น ผู้ใช้จะต้องติดตั้งโปรแกรมเสริมเพิ่มเติมประเภทดังกล่าว



รูปที่ 2.3 ขั้นตอนการแสดงผลใน SVG

2.2.2 รูปแบบการเขียน SVG

ทดลองดูตัวอย่าง code SVG ซึ่งก็จะสังเกตได้ว่ามีลักษณะเป็น XML (คล้ายกับ HTML) ซึ่งต้องมีแท็กเปิดและปิด และมีแท็กที่อยู่ในแท็กด้วย ซึ่งจะสังเกตได้ว่า SVG ไฟล์ จะต้องมี <svg> อยู่เป็นแท็กนอกสุด ดังตัวอย่างต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1: <?xml version="1.0" encoding="iso-8859-1"?>
2: <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000303 Stylable//EN"
3: "http://www.w3.org/TR/2000/03/WD-SVG-20000303/DTD/svg-
20000303-stylable.dtd">
4: <svg xml:space="preserve" width="5.5in" height=".5in">
5: <!-- SVG content goes here -->
6:      ...
7: </svg>
```

รูปที่ 2.4 แสดงตัวอย่างของ SVG แท็ก

- บรรทัดที่ 1-3 เป็นการบอกให้ผู้ที่นำ SVG ไฟล์นี้ไปอ่านทราบว่า code ต่อไปนี้อ้างอิงจากมาตรฐาน XML version 1.0 โดยข้อมูลที่อยู่ในแต่ละแท็กจะใช้มาตรฐาน iso-8859-1 ในการเขียน

- บรรทัดที่สองจะอ้างอิงถึง DTD หรือ Document Type Definition (DTD เป็นเอกสารที่ใช้บอกถึงชนิดของข้อมูลที่จะใช้ใน XML และ ยังระบุอีกด้วยว่าเอกสารนั้นๆ มีโครงสร้างเป็นอย่างไร) บรรทัดที่ 4 เป็นการเริ่มเขียนส่วนที่เป็น SVG element โดยใช้แท็กเปิด <svg> และปิดด้วย </svg> ในบรรทัดที่ 7

- บรรทัดที่ 5 Comment ของ SVG อยู่ใน < ! - - xxxxxxxxxx - ->

- บรรทัดที่ 6 สามารถที่จะบรรจุ text, shapes และ paths ได้ ตามตัวอย่างนี้

```
<rect style="fill:blue;" width="250" height="100"/>
```

2.2.3 Basic Shape SVG

SVG มีการกำหนดรูปทรงมาตรฐานขึ้น 6 แบบได้แก่

- รูปทรงสี่เหลี่ยม (rectangles)
- รูปทรงกลม (circles)
- รูปทรงรี (ellipses)
- เส้นตรง (lines)
- เส้นตรงแบบต่อเนื่อง (polylines)
- รูปทรงหลายเหลี่ยม (polygons)

รูปทรงเหล่านี้สามารถที่จะกำหนดรายละเอียดได้เช่นเดียวกับเส้นทางนั้นทำให้เราสามารถประยุกต์เอาคุณสมบัติต่างๆ ของเส้นทางเข้ามาใช้ได้โดยตรง

ตัวอย่างการเขียน SVG รูปทรงมาตรฐาน

1. รูปวงกลม หรือ element "circle"

แท็กที่ชื่อว่า circle ในภาษา SVG จะมีรูปแบบดังนี้

```
<circle cx=”250" cy=”250" r=”80" fill=”blue”></circle>
```

ความหมายของสัญลักษณ์ดังกล่าวหมายถึงรูปวงกลมซึ่งมีคุณสมบัติดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.1 คำอธิบายและสัญลักษณ์ที่ใช้ของแท็ก circle

คำอธิบาย	สัญลักษณ์ที่ใช้
มีจุดศูนย์กลางที่พิกัด (x,y) ที่ (250,250)	cx="250" cy="250"
รัศมี 80 หน่วย	r="80"
ระบายสีน้ำเงินไว้ภายในวงกลม	fill="blue"

2. รูปปิด หรือ element "path"

แท็กที่ชื่อว่าเส้นทางจะมีรูปแบบดังนี้

<path style="opacity:0.3;fill:#CC0000;stroke:none;" d="M 0 260,L140 260, L14
60"> </path>

ความหมายของสัญลักษณ์ดังกล่าวหมายถึงรูปปิดซึ่งมีคุณสมบัติดังนี้

ตาราง 2.2 คำอธิบายและสัญลักษณ์ที่ใช้ของแท็ก path

คำอธิบาย	สัญลักษณ์ที่ใช้
ความเข้มสี (ความโปร่งแสง) 30%	opacity:0.3
ระบายสีภายในรูปปิด (สีแดง)	fill:#CC0000
สีของเส้นขอบรูป (ไม่มี)	stroke:none
เริ่มต้นลากจากจุด 0,260 ไปยัง 140,260 สิ้นสุดที่ 140,60	d="M 0 260, L140 260, L140 60"

Element อื่นๆของ SVG ที่มีการใช้งานบ่อย

<g>	ใช้ในการแท็กต่างๆให้อยู่ในกลุ่มเดียวกัน
<desc>	ใช้ในการพิ่มเติมข้อมูลหรือคำอธิบาย
<defs>	ใช้ในการกำหนดค่าต่างๆ
<script>	ใช้ในการเขียน script
<cursor>	ใช้ในการเปลี่ยนรูป mouse cursor
<textpath>	ใช้ในการเขียนตัวอักษรให้เป็นไปตามเส้นทางที่กำหนด
<image>	ใช้ในการแสดงรูปภาพในรูปแบบอื่นๆเช่น jpeg, gif และ png เป็นต้น

แอตทริบิวต์อื่นๆของ SVG ที่มีการใช้งานบ่อย

id	ใช้ในการแยกแยะวัตถุต่างๆ
Xmlns	ใช้ในการอ้างถึง namespace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Visibility	ใช้กำหนดว่าให้วัตถุนั้นๆปรากฏอยู่หรือไม่
Point-event	ใช้กำหนดปฏิสัมพันธ์ที่มีกับวัตถุ
ViewBox	ใช้กำหนดขนาดการแสดงผลภาพ
zoomAndPan	ใช้กำหนดความสามารถในการย่อและขยายภาพ
Onclick	ใช้กำหนดพฤติกรรมของวัตถุเมื่อคลิกเมาส์
onmouseover	ใช้กำหนดพฤติกรรมของวัตถุเมื่อเมาส์อยู่บนวัตถุ
onmouseout	ใช้กำหนดพฤติกรรมของวัตถุเมื่อเลื่อนเมาส์ออกจากวัตถุ
Transform	ใช้ในการเคลื่อนย้าย, หมุนและปรับอัตราส่วนของภาพ
Xlink:href	ใช้ในการอ้างถึง URL

2.2.4 ประโยชน์ในการใช้ SVG

1. SVG สามารถอ่านและแก้ไขได้โดยโปรแกรมหลายโปรแกรม เช่น notepad
2. สามารถเปลี่ยนขนาดของภาพได้โดยคุณภาพของภาพไม่ลดลง
3. SVG สามารถทำงานร่วมกับภาษาจาวาได้
4. SVG เป็นมาตรฐานที่เปิดเผยให้คนทั่วไปทราบ
5. ข้อความใน SVG มีความสามารถในการเลือกและค้นหาจึงเหมาะแก่การนำมาทำเป็นแผนที่

2.3 Global Positioning System

GPS (Global Positioning System) คือระบบกำหนดตำแหน่งบนพื้นโลก หรือชื่อเต็มคือ NAVSTAR GPS (Navigation System with Timing And Ranging Global Positioning System) โดยริเริ่มจากกระทรวงกลาโหมของประเทศสหรัฐอเมริกา (US. Department of Defense, DoD) ที่มีความต้องการระบบที่ให้ค่าตำแหน่งที่ถูกต้องและครอบคลุมทั่วโลกตลอด 24 ชั่วโมงทุกสภาพอากาศ ทั้งนี้ในเริ่มแรกจุดมุ่งหมายของระบบสร้างขึ้นเพื่อมารองรับงานด้านการทหาร

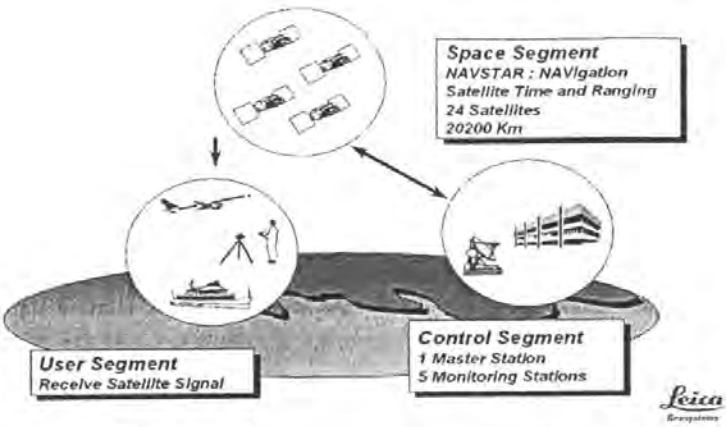
2.3.1 องค์ประกอบของระบบดาวเทียม GPS

ส่วนควบคุม(Control Segment)

ส่วนอวกาศ(Space Segment)

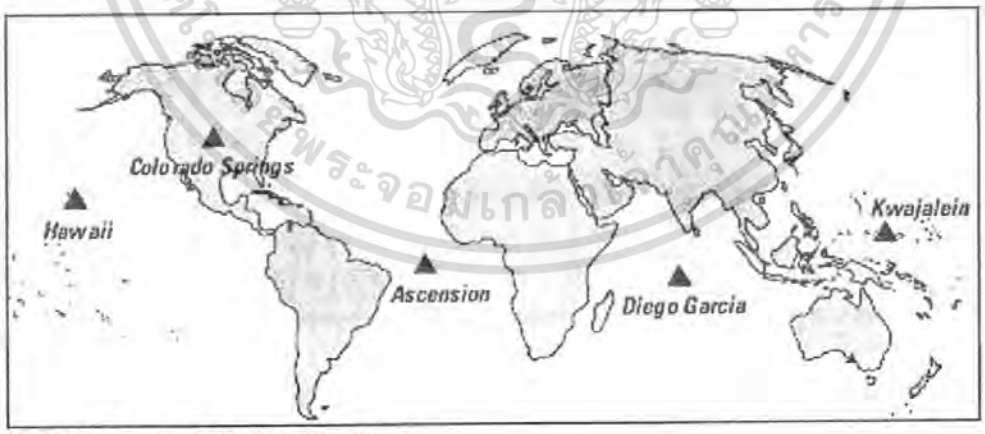
ส่วนผู้ใช้(User Segment)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 องค์ประกอบของระบบดาวเทียม GPS

2.3.1.1 ส่วนควบคุม (Control Segment) ในส่วนของสถานีควบคุมจะประกอบด้วย 5 สถานีย่อย (Monitor Station) ตั้งอยู่ที่เกาะ Diego Garcia, Ascension Island, Kwajalein, และ Hawaii ส่วนสถานีควบคุมหลัก (Master Control Station) 1 สถานี ซึ่งเป็นศูนย์ควบคุมการทำงานของระบบดาวเทียมจีพีเอส ตั้งอยู่ที่เมือง Colorado Springs รัฐ Colorado สหรัฐอเมริกา สถานีควบคุมต่าง ๆ เหล่านี้มีหน้าที่คอยติดต่อสื่อสาร (Tracking) กับดาวเทียม ทำการคำนวณผล (Computation) เพื่อบอกตำแหน่งของดาวเทียมแต่ละดวง และส่งข้อมูลปรับแก้ไขข้อมูลวงโคจร (Ephemeris)และข้อมูลเวลา(Clock Correction)ไปยังดาวเทียมอยู่ตลอดเวลา ทำให้ข้อมูลที่ได้เป็นข้อมูลที่ทันสมัยอยู่เสมอ



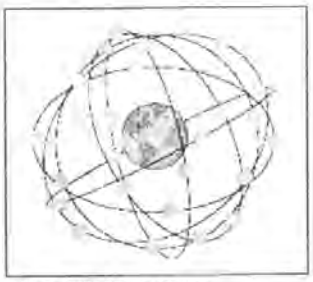
Control Segment Station Locations

รูปที่ 2.6 แสดงส่วนควบคุม

2.3.1.2 ส่วนอวกาศ (Space Segment) ในระบบดาวเทียม GPS ที่ออกแบบไว้มีจำนวนดาวเทียมทั้งหมด 24 ดวงโคจรอยู่รอบโลกใน 6 ระนาบ โดยแต่ระนาบนั้นจะเอียงทำมุม 55° กับเส้นศูนย์สูตร (Equator) ในแต่ละระนาบมีดาวเทียมอยู่ 4 ดวง อยู่สูงจากพื้นโลกประมาณ 20,200km มีวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โคจรทุก 12 ชั่วโมง ด้วยเหตุนี้ทำให้ผู้ใช้งานสามารถรับสัญญาณดาวเทียมได้อย่างน้อย 5 ดวงตลอด 24 ชั่วโมง



GPS Satellite Constellation

รูปที่ 2.7 แสดงส่วนอวกาศ

2.3.1.3 ส่วนผู้ใช้ (User Segment) ประกอบด้วย 2 ส่วนใหญ่ๆคือส่วนที่เกี่ยวข้องกับทางทหาร(Military)และทางพลเรือน(Civilian) ซึ่งทางพลเรือนจะได้รับสัญญาณโดยไม่คิดค่าใช้จ่ายแต่ผู้ใช้ต้องหาจานรับ(Antenna)และเครื่องรับ(Receiver)ด้วยตนเอง

นโยบายการให้บริการข้อมูล จีพีเอส ของรัฐบาลสหรัฐฯ มีดังนี้

Precise Positioning Services : PPS

- ใช้ในการทางทหารเป็นหลัก
- ข้อมูลจะมีการเข้ารหัส เฉพาะผู้ที่มีเครื่องถอดรหัสจึงจะสามารถใช้งานได้
- ความถูกต้องของพิกัด คือ 22 เมตร ในแนวราบ, 27.7 เมตร ในแนวดิ่ง และ 200 nanosecond (UTC)

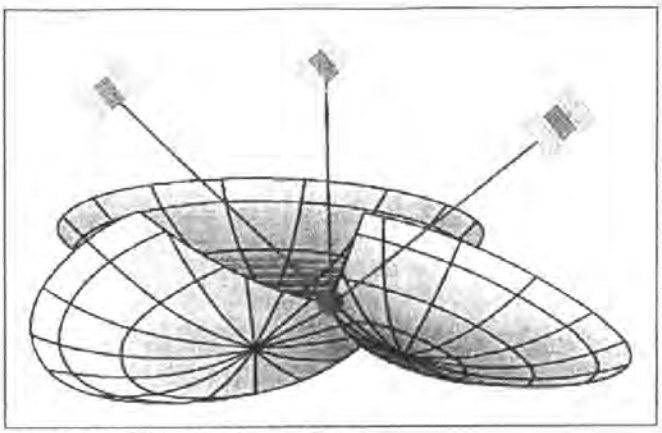
Standard Positioning Services : SPS

- ใช้ในกิจการพลเรือนเป็นหลัก
- ความถูกต้องลดลงเนื่องจาก Selective Availability (SA)
- ความถูกต้องของพิกัดคือ 100 เมตร ในแนวราบ, 156 เมตร ในแนวดิ่ง และ 340 nanosecond (UTC)

2.3.2 หลักการหาตำแหน่งของระบบ GPS

หลักการหาตำแหน่งจะใช้หลักการเล็งสกัดย้อน(Intersection) โดยการหาระยะทางจากเครื่องรับสัญญาณจีพีเอสไปยังดาวเทียมจีพีเอสที่มีตำแหน่งของตัวเอง อย่างน้อย 3 ดวง

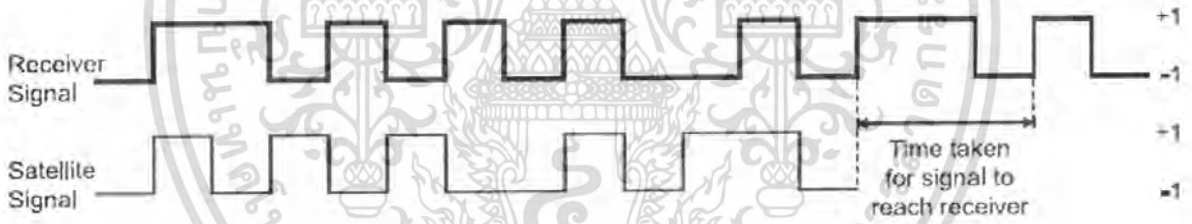
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Intersection of three imaginary spheres

รูปที่ 2.8 Ranging Intersection

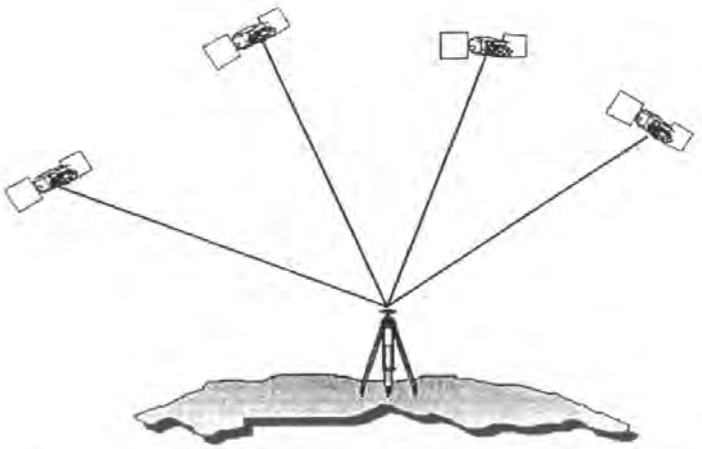
ระยะทางที่หาได้โดยการหาค่าต่างเวลาที่คลื่นเดินทางจากดาวเทียมจีพีเอสมายังเครื่องรับสัญญาณดาวเทียมจีพีเอสคูณกับค่าความเร็วของคลื่นแม่เหล็กไฟฟ้า (3x108 m/s) สำหรับการหาค่าต่างเวลานั้นได้จากการที่เครื่องรับสัญญาณดาวเทียมทำการสร้างรหัส C/A Code เทียบกับ C/A Code ที่ส่งมาจากดาวเทียมจีพีเอสค่าต่างเวลาที่เกิดขึ้นคือเวลาคลื่นใช้เดินทาง



รูปที่ 2.9 การเปรียบเทียบรหัสเพื่อหาค่าต่างเวลา

แต่เนื่องด้วยวิธีดังกล่าวจะต้องใช้ Oscillator ของเครื่องรับสัญญาณดาวเทียมที่มีความเที่ยงสูงมากแต่ก็ยังมีความถูกต้องน้อยกว่านาฬิกาอะตอมที่ใช้ในดาวเทียมจีพีเอสซึ่งจะต้องใช้ดาวเทียมอย่างน้อย 4 ดวงเพื่อลดผลของความคลาดเคลื่อนที่เกิดจากเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ใช้ดาวเทียม 4 ดวงเพื่อหาค่า Latitude Longitude Height และ Time

2.3.3 สาเหตุความคลาดเคลื่อนที่เกิดขึ้นในระบบ GPS

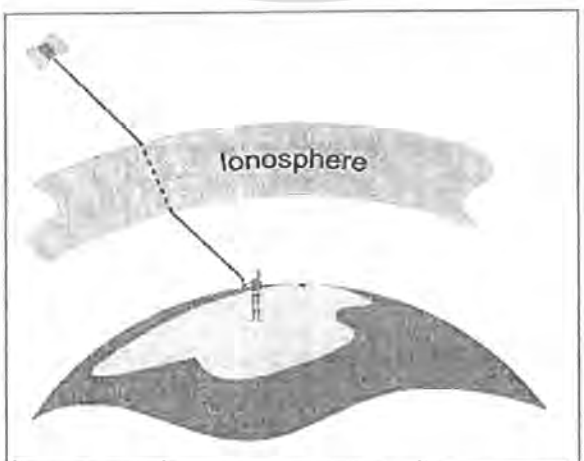
2.3.3.1 ความคลาดเคลื่อนเนื่องจากดาวเทียม (Satellite Errors) ประกอบด้วย

- ความคลาดเคลื่อนเนื่องจากนาฬิกาดาวเทียม ถึงแม้ว่านาฬิกาอะตอมจะมีความแม่นยำสูง แต่ก็ยังคงมีความผิดพลาดอยู่ (ประมาณ 3 nanosecond) ก็ส่งผลให้คลื่นที่ส่งออกมามีความผิดพลาดด้วย ดังนั้น Control Segment จึงจะต้องมีการปรับแก้ให้เที่ยงตรงอยู่เสมอ

-ความคลาดเคลื่อนเนื่องจากวงโคจร ดาวเทียมที่โคจรอยู่นอกโลกจะแรงภายนอกมากระทำเช่น แรงดึงดูดของดวงจันทร์ แรงดึงดูดของดวงอาทิตย์ ลมสุริยะ ซึ่งส่งผลต่อตำแหน่งวงโคจรของดาวเทียม ดังนั้น Control Segment จึงจะต้องมีการปรับแก้ให้เที่ยงตรงอยู่เสมอ

2.3.3.2 ความคลาดเคลื่อนเนื่องจากชั้นบรรยากาศ (Atmospheric Errors) ประกอบด้วย

- ความคลาดเคลื่อนเนื่องจากชั้นบรรยากาศไอโอโนสเฟียร์ (Ionosphere) ไอโอโนสเฟียร์เป็นชั้นบรรยากาศที่อยู่สูงจากพื้นโลกประมาณ 50 km ถึง 1,000 km ซึ่งเป็นชั้นบรรกาศชั้นแรกที่คลื่นจะต้องเดินทางผ่านก่อนมาถึงพื้นโลก

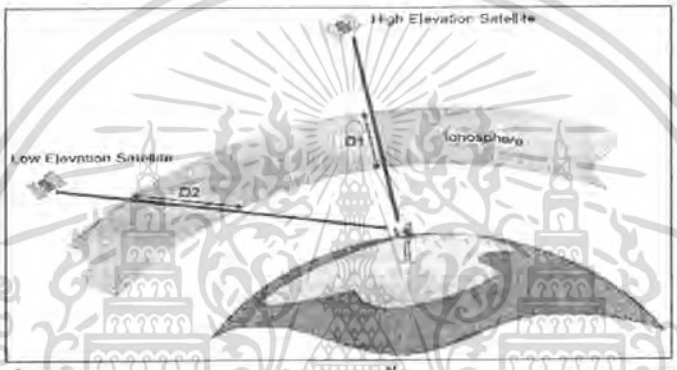


รูปที่ 2.11 แสดงชั้นบรรยากาศไอโอโนสเฟียร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในบรรยากาศไอโอโนสเฟียร์ประกอบด้วยประจุของอิเลคตรอนกระจายอยู่ ความหนาแน่นของประจุอิเลคตรอนนี้แปรเปลี่ยนไปตามผลของรังสีอัลตราไวโอเลต ผลกระทบที่มีต่อคลื่นสัญญาณจีพีเอสในลักษณะแปรผกผันกับความถี่ของคลื่น ทำให้ข้อมูลเฟสของคลื่นส่งเดินทางได้เร็วขึ้น (การวัดด้วยคลื่นส่ง) และข้อมูลซูโดเรนจ์ (การวัดด้วยข้อมูล C/A Code) เดินทางได้ช้าลง ส่งผลให้การวัดระยะทางระหว่างดาวเทียมกับเครื่องรับสัญญาณผิดพลาดไป

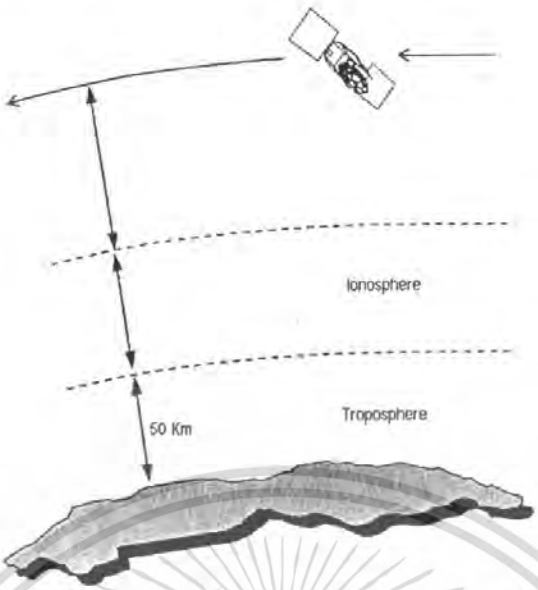
ผลกระทบของชั้นบรรกาศไอโอโนสเฟียร์จะมีผลต่อคลื่นสัญญาณจีพีเอสมากขึ้นเมื่อดาวเทียมอยู่ในระดับต่ำเมื่อเทียบกับจุดรับสัญญาณ เนื่องจากระยะทางที่คลื่นทางผ่านชั้นบรรยากาศจะมากกว่าคลื่นที่มาจากดาวเทียมในระดับสูงกว่า



รูปที่ 2.12 แสดงผลกระทบที่เกิดจากชั้นบรรยากาศไปโอโนสเฟียร์

ความคลาดเคลื่อนเนื่องจากชั้นบรรยากาศโทรโพสเฟียร์ (Troposphere) อยู่สูงจากพื้นโลกประมาณ 50 km ชั้นบรรยากาศโทรโพสเฟียร์สามารถแบ่งออกได้เป็น 2 ส่วนประกอบคือ ส่วนประกอบแห้งและส่วนประกอบชื้น โดยส่วนประกอบแห้งจะมีขนาดประมาณ 90 % ของการหักเหในชั้นบรรยากาศโทรโพสเฟียร์ ส่วนประกอบชื้นจะขึ้นอยู่กับความกดดันของไอน้ำในอากาศ และมีขนาดประมาณ 10 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

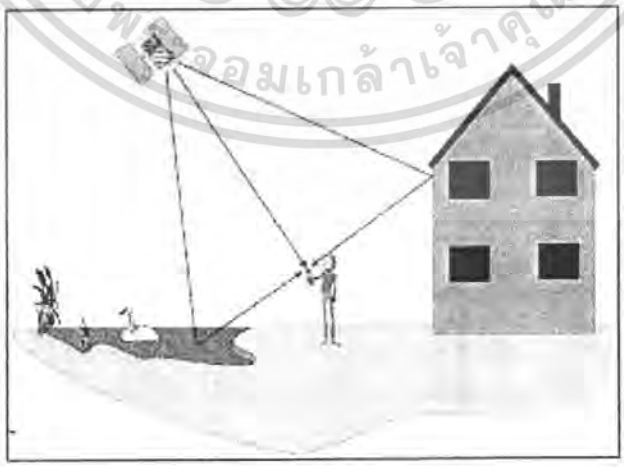


รูปที่ 2.13 แสดงชั้นบรรยากาศโทรโพสเฟียร์

การหักเหในชั้นบรรยากาศนี้คลื่นจะเดินทางมายังเครื่องรับสัญญาณช้ากว่าความเป็นจริง ทั้งข้อมูลเฟสของคลื่นส่ง(การวัดด้วยคลื่นส่ง) และข้อมูลซูโดเรนจ์ (การวัดด้วยข้อมูล C/A Code) ค่าความคลาดเคลื่อนนี้จะขึ้นอยู่กับอุณหภูมิ ความดันบรรยากาศและความชื้นในอากาศขณะที่ทำการวัด

2.7.3.3 ความคลาดเคลื่อนเนื่องจากนาฬิกาของเครื่องรับสัญญาณ (Receiver Clock Errors)

2.7.3.4 ความคลาดเคลื่อนเนื่องจากคลื่นหลายวิถี (Multipath Errors) เกิดจากคลื่นสัญญาณจีพีเอสเดินทางไปกระทบวัตถุสะท้อนคลื่นก่อนเข้าสู่เครื่องรับสัญญาณ วัตถุสะท้อนคลื่นเช่น ผนังคอนกรีต ผิวน้ำ ส่งผลให้ข้อมูลระยะทางจากดาวเทียมไปยังเครื่องรับสัญญาณผิดไปจากเดิม



รูปที่ 2.14 การเกิด Multipath

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 ความคลาดเคลื่อนเนื่องจากการวางตัวของดาวเทียม (Dilution of Precision, DOP)

การหาตำแหน่งค่าพิกัดนอกจากสัญญาณดาวเทียมได้อย่างน้อย 4 ดวงแล้วการวางตำแหน่งของดาวเทียมบนท้องฟ้าก็เป็นส่วนสำคัญ ค่า Dilution of Precision หรือ DOP เป็นค่าที่ได้จากกระบวนการทางคณิตศาสตร์ที่บ่งบอกความไม่แน่นอนด้านตำแหน่ง ค่า DOP ที่สูงค่าให้ความไม่แน่นอนด้านตำแหน่งมากกว่า ค่า DOP ที่ต่ำ โดยแบ่งได้ดังนี้

HDOP= Horizontal Dilution of Precision ความไม่แน่นอนในองค์ประกอบด้านตำแหน่งในแนวราบ

VDOP= Vertical Dilution of Precision ความไม่แน่นอนในองค์ประกอบด้านความสูง

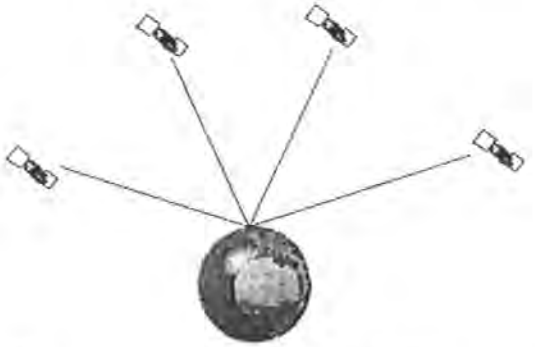
PDOP= Position Dilution of Precision ความไม่แน่นอนในองค์ประกอบด้านตำแหน่งในสามมิติ โดย PDOP = HDOP + VDOP

TDOP= Time Dilution of Precision ความไม่แน่นอนในองค์ประกอบด้านเวลา

GDOP = Geometry Dilution of Precision โดย GDOP = PDOP + TDOP

โดยถ้าแบ่งท้องฟ้าออกเป็นสี่จตุภาค ดาวเทียมจะต้องอยู่เหนือเครื่องสัญญาณเกระจายอยู่ในทั้งสี่จตุภาคจึงจะให้ค่า DOP ที่ดี

รูปที่ 2.15 การวางตัวของดาวเทียมที่ให้ค่า DOP ไม่ดี



รูปที่ 2.16 การวางตัวของดาวเทียมที่ให้ค่า DOP ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ข้อมูล NMEA และการวิเคราะห์ข้อมูล

NMEA ย่อมาจาก National Maritime Electronics Association ซึ่งเป็นสมาคมที่มุ่งเน้นศึกษาและพัฒนาอุปกรณ์อิเล็กทรอนิกส์ที่ใช้งานทางทะเล หรืออุปกรณ์เดินเรือ

โดยปกติการที่ระบบจะใช้ข้อมูลจีพีเอสได้นั้น อุปกรณ์รับข้อมูลจีพีเอสเอาต้องมีตัวเชื่อมต่อกับอินเตอร์เฟส RS232 คำสั่งหรือข้อมูลที่ผ่านทาง RS232 โดยข้อมูลเหล่านี้จะอยู่ภายใต้มาตรฐานของ NMEA ซึ่งปัจจุบันมาตรฐานของอุปกรณ์รับข้อมูลจีพีเอส จะอยู่ในรูปแบบ NMEA 0183 เป็นลักษณะข้อความภาษา ASCII

NMEA-0183 ใช้รหัสอักษร ASCII และการสื่อสารข้อมูลแบบ Serial Connection ในการส่งข้อมูล จาก Talker ตัวหนึ่งไปยัง Listeners หนึ่งหรือหลายตัว อัตราการส่งข้อมูลอยู่ที่ 4,800 Baud Rate โดยมาตรฐาน

ชนิดของข้อมูล NMEA จะปรากฏในคำแรกของประโยคข้อมูลที่ขึ้นต้นด้วยตัวอักษร "$" โดยแต่ละชนิดจะสื่อถึงความหมายและข้อมูลที่แตกต่างกัน

ชนิดของข้อมูล NMEA ที่สำคัญมีดังต่อไปนี้

2.4.1 $GPGAA

คือข้อมูลปรับค่า ที่ใช้ในการปรับค่าข้อมูลความแม่นยำและจัดหาข้อมูลพิกัดแบบสามมิติ

ตัวอย่าง

" $GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47 "

โดย

GGA คือ ชนิดของ NMEA (Global Positioning System Fix Data)

123519 คือได้รับการปรับค่า ณ เวลา 12:35:19 UTC

4807.038,N คือค่า Latitude 48 deg 07.038' N

01131.000,E คือค่า Longitude 11 deg 31.000' E

1 คือค่า Fix quality โดยแต่ละค่ามีความหมายต่อไปนี้

0 = invalid

1 = GPS fix (SPS)

2 = DGPS fix

3 = PPS fix

4 = Real Time Kinematic

5 = Float RTK

6 = estimated (dead reckoning) (2.3 feature)

7 = Manual input mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8 = Simulation mode

08 คือจำนวนดาวเทียมที่กำลังติดตามค่า

0.9 ค่า Horizontal dilution ของพิกัด

545.4,M ความสูงเหนือจากระดับน้ำทะเล

46.9,M ระยะสูงจากระดับน้ำทะเลเฉลี่ยเทียบจาก WGS84 ellipsoid

(empty field) ระยะเวลา (วินาที) จาก DGPS update

(empty field) DGPS station ID number

*47 ข้อมูล checksum

2.4.2 $GPGSA

คือข้อมูลบอกสถานะของดาวเทียม บอกค่าเจือจางของความแม่นยำ หรือ DOP ซึ่งเป็นสัญญาณของผลจากการปรับค่าความแม่นยำของดาวเทียมภูมิศาสตร์

ตัวอย่าง

$GPGSA,A,3,04,05,,09,12,,,24,,,,,2.5,1.3,2.1*39

GSA Satellite status

A บอกว่าเป็น Auto selection ของ2D or 3D fix โดยจะมีค่า M เมื่อ manual

3 ค่า 3D fix โดยจะมีค่าเป็น

1 = no fix

2 = 2D fix

3 = 3D fix

04,05... เลข PRNs ของดาวเทียมที่ใช้ปรับแต่ง

2.5 PDOP (dilution of precision)

1.3 Horizontal dilution of precision (HDOP)

2.1 Vertical dilution of precision (VDOP)

*39 ข้อมูล checksum

4.2.3 $GPGSV

คือข้อมูลของดาวเทียมที่รับสัญญาณได้

$GPGSV,2,1,08,01,40,083,46,02,17,308,41,12,07,344,39,14,22,228,45*75

โดย

GSV Satellites in view

2 จำนวนของ Sentence ทั้งหมดของข้อมูลเต็ม

1 เป็น sentence ที่ 1

08 จำนวนของดาวเทียมที่รับค่าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

83121

01 เลข PRN ของดาวเทียม

40 Elevation, degrees

083 Azimuth, degrees

46 ค่า SNR

*75 ข้อมูล checksum

4.2.4 $GPRMC

คือ ข้อมูลองค์ประกอบสำคัญของจีพีเอสประกอบไปด้วยข้อมูลที่สำคัญคือ ตำแหน่ง ความเร็ว และเวลา

ตัวอย่าง

$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A

โดย

RMC Recommended Minimum sentence C

123519 คือได้รับการปรับค่า ณ เวลา 12:35:19 UTC

A สถานะ โดย A=active or V=Void.

4807.038,N Latitude 48 deg 07.038' N

01131.000,E Longitude 11 deg 31.000' E

022.4 ความเร็วบนพื้นผิด (knots)

084.4 องศาที่ทำ track ค่า

230394 วันที่ 23 rd of March 1994

003.1,W Magnetic Variation

*6A ข้อมูล checksum

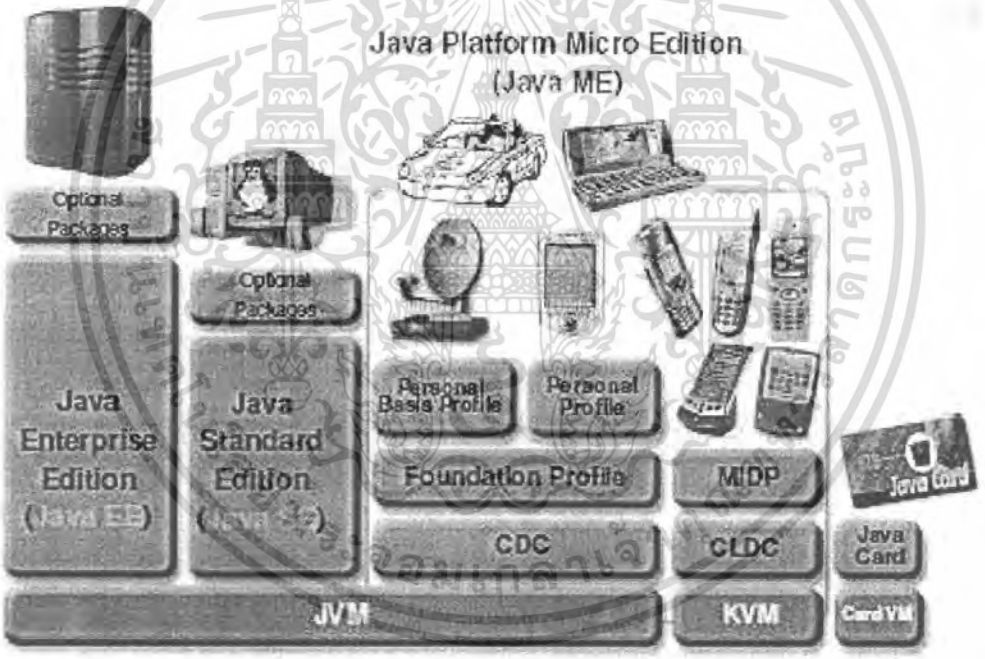
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
เทคโนโลยีที่เกี่ยวข้อง

3.1 เทคโนโลยี Java ME

3.1.1 สถาปัตยกรรม Java ME

Java ME เป็นแพลตฟอร์มจาวาที่ออกแบบมาโดยมีจุดมุ่งหมายสำหรับอุปกรณ์ที่มีขนาดเล็กซึ่งมีทรัพยากรจำกัดทางด้านพลังงาน ความสามารถในการประมวลผล หน่วยความจำ โดยแอพพลิเคชันที่พัฒนาจาก Java ME จะทำงานบนอุปกรณ์ขนาดเล็กเช่น โทรศัพท์มือถือ พีดีเอ โทรศัพท์พร้อมจอภาพที่ต่อกับอินเตอร์เน็ตโทรทัศน์ดิจิตอลขนาดเล็ก อุปกรณ์บันเทิงและระบบนำทางในรถยนต์ สวิตช์ในระบบเครือข่าย คอมโพเน็นท์ของเครื่องอำนวยความสะดวกภายในบ้าน เป็นต้น



รูปที่ 3.1 แสดงโครงสร้างของ Java เทคโนโลยี

Java ME ได้นำโครงสร้างแบบโมดูลที่มีความยืดหยุ่นสูงเข้ามาใช้เพื่อให้สามารถสนับสนุนการทำงานอุปกรณ์หลากหลายประเภท Java ME กำหนดชั้นของซอฟต์แวร์ (Software) ไว้ 3 เลเยอร์ (Layer) ด้วยกัน โดยเลเยอร์ทั้งหมดจะอยู่เหนือชั้นระบบปฏิบัติการของอุปกรณ์ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.1 เลเยอร์จาวาเวอร์ชวลแมชชีน (Java Virtual Machine)

เป็นเลเยอร์ของจาวาเวอร์ชวลแมชชีนเป็นส่วนของระบบจัดการที่ควบคุมและทำงานให้ Java และระบบปฏิบัติการของอุปกรณ์ทำงานร่วมกันได้ โดยจะปรับแต่งให้ Java ทำงานร่วมกับระบบปฏิบัติการของอุปกรณ์ได้ โดยมากจะเป็นการแปลงจาก code Java ไปเป็นคำสั่งที่ระบบปฏิบัติการของอุปกรณ์เข้าใจ โดยเวอร์ชวลแมชชีนของ Java ME ได้แก่ Connected Virtual Machine (CVM) และ Kilobyte Virtual Machine (KVM) ทั้ง CVM และ KVM ต่างสนับสนุนยูติลิตี้ JavaCodeCompact หรือคลาส Prelinker preloader และ Romnizer ยูติลิตี้นี้จะโยงจาวาคลาสเข้ากับเวอร์ชวลแมชชีน ช่วยลดระยะเวลาเริ่มใช้งาน VM ได้มากขึ้น ตัวอย่างเช่นเรียก KVM และ CLDC ขึ้นมาก่อนโหลดไปยังหน่วยความจำแบบ ROM ของอุปกรณ์ คลาส CLDC จะถูกโยงเข้าใน KVM โดยตรง เพื่อปรับปรุงประสิทธิภาพการทำงาน โดยเราจะเรียกขั้นตอนนี้ว่า การโหลดเข้าหน่วยความจำแบบ ROM (ROMized)

3.1.1.2 เลเยอร์คอนฟิกูเรชัน (Configuration)

เป็นเลเยอร์ของคอนฟิกูเรชันของ Java ME ซึ่งเป็นตัวกำหนดฟีเจอร์และคลาสไลบรารีของอุปกรณ์ใช้งานทั่วไปหรือกลุ่มอุปกรณ์ที่มีความต้องการหน่วยความจำและหน่วยประมวลผลใกล้เคียงกัน คอนฟิกูเรชันในจาวาสำหรับอุปกรณ์ขนาดเล็กมี 2 ประเภทด้วยกันคือ CDC (Connected Device Configuration) และ CLDC (Connected Limited Device Configuration) โดยแต่ละตัวจะใช้ VM (Virtual Machine) ที่ต่างกันด้วย

นอกจากนี้แล้วสำหรับอุปกรณ์ที่มีข้อจำกัดในการทำงานจะเป็นบรรทัดฐานในการแบ่งกลุ่ม โดยคุณสมบัติพื้นฐานทั้งหมดนี้จะใช้ในการจัดแบ่งกลุ่มของอุปกรณ์ซึ่งมีความคล้ายกันในเรื่องของหน่วยความจำ หน้าจอ เครือข่ายในการเชื่อมต่อ และพลังงาน

3.1.1.3 เลเยอร์โพรไฟล์ (Profile)

เป็นเลเยอร์ที่สร้างเหนือเลเยอร์คอนฟิกูเรชัน จึงเป็นตัวกำหนดฟังก์ชันในการทำงานพร้อมทั้งเป็นตัวกลางระหว่างแอพพลิเคชันกับสภาวะแวดล้อมของ Java ME ดังนั้นโพรไฟล์จึงเกี่ยวข้องกับคุณลักษณะทางด้านฮาร์ดแวร์ของอุปกรณ์แต่ละตัว ตัวอย่างของโพรไฟล์ในเลเยอร์นี้ได้แก่ PDA Profile, MID Profile, Foundation Profile และ Personal Profile เป็นต้น

3.1.2 คอนฟิกูเรชันใน Java ME

คอนฟิกูเรชันและโพรไฟล์เป็นองค์ประกอบหลักของ Java ME โดยมีจุดสำคัญคือเพื่อปรับแต่งเวอร์ชวลแมชชีนและคลาสไลบรารี่ ให้เหมาะสมกับอุปกรณ์แต่ละประเภท คอนฟิกูเรชันคือชุดที่มีคุณสมบัติขั้นต่ำของจาวาเวอร์ชวลแมชชีนและจาวาคลาสสำหรับอุปกรณ์แต่ละประเภทเป็นตัวแทนของอุปกรณ์ที่มีลักษณะคล้ายคลึงกัน อาจกล่าวได้ว่าคอนฟิกูเรชันเป็นตัวกำหนดคุณสมบัติหรือไลบรารีขั้นต่ำของแพลตฟอร์มจาวาที่นักพัฒนาคาดว่าจะต้องมีในทุกอุปกรณ์ ขณะที่คลาสไลบรารี่ที่กำหนดในคอนฟิกูเรชันจะมีในทุกอุปกรณ์ที่อยู่ในกลุ่มเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบันคอนฟิกูเรชันใน Java ME แบ่งเป็น 2 ประเภทคือ CDC (Connected Device Configuration) และ CLDC (Connected Limited Device Configuration)

1. CDC เป็นอุปกรณ์ใช้ร่วมกับอุปกรณ์อื่นที่ติดตั้งตายตัวและใช้เชื่อมต่อข้อมูล โดยปรกติมักมีส่วนติดต่อกับผู้ใช้หลากหลายแบบ มีหน่วยความจำประมาณ 2-16 เม็กกะไบต์ ใช้หน่วยประมวลผลแบบ 32 บิตหรือมากกว่า เชื่อมต่อเครือข่ายที่มีแบนด์วิดท์ (Bandwidth) สูงอย่างต่อเนื่องโดยอาศัยพอร์ต TCP/IP ตัวอย่างอุปกรณ์ประเภทนี้ได้แก่ โทรทัศน์ดิจิตอลขนาดเล็ก อินเตอร์เน็ตทีวี โทรศัพท์พร้อมจอภาพที่เชื่อมต่อกับอินเตอร์เน็ต อุปกรณ์สื่อสารที่มีความซับซ้อนสูง อุปกรณ์บันเทิงและระบบนำทางในรถยนต์

2. CLDC เป็นอุปกรณ์ประเภทที่พกพาได้และใช้เชื่อมต่อข้อมูล โดยปรกติมักมีส่วนติดต่อกับผู้ใช้แบบง่ายๆเมื่อเทียบกับระบบบนเครื่องคอมพิวเตอร์ตั้งโต๊ะ มีหน่วยความจำประมาณ 128 กิโลไบต์ถึง 1 เม็กกะไบต์ ใช้หน่วยประมวลผลแบบ 16 หรือ 32 บิต เชื่อมต่อกับเครือข่ายที่มีแบนด์วิดท์ต่ำเป็นระยะเวลาสั้นๆ โดยไม่อาศัยพอร์ต TCP/IP ตัวอย่างอุปกรณ์ประเภทนี้ ได้แก่ โทรศัพท์มือถือแบบไม่ซับซ้อนมากนัก เพจเจอร์รับส่งข้อความ เครื่องปาล์มโอเอสแบบพกพา

อุปกรณ์ไร้สายจะทำงานสนับสนุน Java ME ได้ดีเมื่อมีคุณสมบัติตรงตามข้อกำหนด หากต้องการให้ KVM ทำงานได้อย่างมีประสิทธิภาพโดยใช้ไลบรารี่ CLDC จะต้องมีคุณสมบัติของระบบขั้นต่ำ ดังนี้

มีหน่วยความจำ 160 – 512 กิโลไบต์ สำหรับสร้างแพลตฟอร์มจาวา

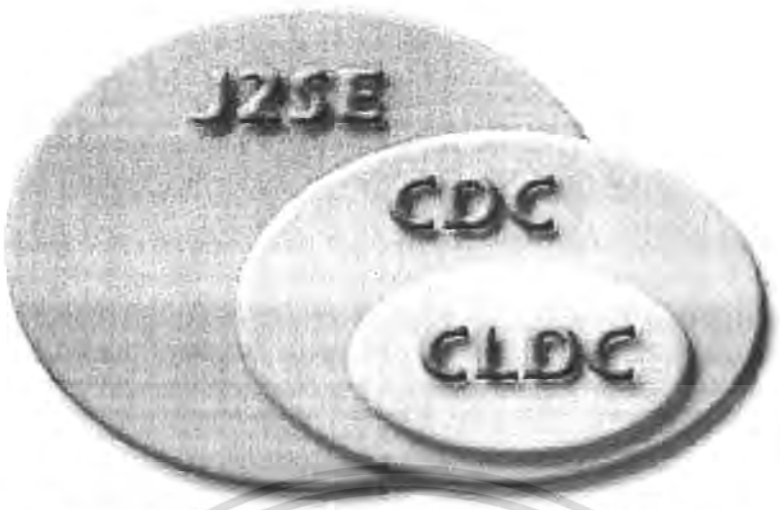
มีหน่วยประมวลผลแบบ 16-32 บิตความเร็ว 25 เม็กกะเฮิรซ์

พลังงานโดยมากมักทำงานโดยใช้แบตเตอรี

เชื่อมต่อกับเครือข่ายได้ในช่วงสั้นๆ อาศัยระบบไร้สายเป็นส่วนใหญ่ แบนด์วิดท์ค่อนข้างจำกัด ความเร็ว 9600 ไบต์ต่อวินาทีหรือน้อยกว่า

มีหน่วยความจำชั่วคราวขนาด 32 กิโลไบต์ สำหรับเก็บจาวารันไทม์และหน่วยความจำของอ็อบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงความสัมพันธ์ระหว่าง Java SE และ คลาสไลบรารีของ CDC และ CLDC

ในชั้นของคอนฟิกูเรชันมีคลาส 2 ประเภทด้วยกันคือคลาสที่นำมาจาก Java SE และคลาสที่ออกแบบเฉพาะอุปกรณ์ขนาดเล็ก คลาสที่นำมาจาก Java SE จะมีคุณสมบัติอย่างเดียวกันกับคลาสใน Java SE หรือเป็นซับคลาสของ Java SE เช่น แพ็คเก็จ java.io และ java.util จากรูป 2.2 แสดงความสัมพันธ์ระหว่าง Java SE และคลาสไลบรารีของ CDC และ CLDC จากภาพจะเห็นว่าคลาสใน CLDC ส่วนใหญ่ทำงานร่วมกับคลาสใน CDC ได้ดีเช่นเดียวกับกรณีของคลาสใน CLDC และคลาสใน Java SE

คลาสที่ไม่ได้นำมาจาก Java SE และออกแบบเพื่อใช้เฉพาะอุปกรณ์มักทำงานร่วมกับ JavaSE ได้ไม่ค่อยดีใน CLDC คลาสเหล่านี้จะอยู่ในกลุ่มกรอบการติดต่อสื่อสารทั่วไป (Generic Connection Framework) โดยระบุไว้ในแพ็คเก็จ javax.microedition.io

คอนฟิกูเรชันยังระบุคุณสมบัติของจาวาเวอร์ชวลแมชชีนในเลเยอร์ด้านล่างอีกด้วย ในโครงสร้างปัจจุบัน CDC และ CLDC มีเวอร์ชวลแมชชีนที่ปรับแต่งมาเฉพาะตัวอยู่แล้ว เวอร์ชวลแมชชีนของ CDC คือ C Virtual Machine (CVM) มีคุณสมบัติครบถ้วนเหมือน Java 2 Virtual Machine แต่ขนาดเล็กกว่าออกแบบมาสำหรับอุปกรณ์ที่สลับซับซ้อน CVM มีความต้องการหน่วยความจำ 256 กิโลไบต์ ขณะที่หน่วยความจำแบบ ROM ของ CDC มีขนาด 1 เม็กกะไบต์ เวอร์ชวลแมชชีนของ CLDC คือ K Virtual Machine (KVM) แม้จะมีขนาดเล็กแต่มีความสามารถในการทำงานสูง ออกแบบมาเพื่อใช้งานกับอุปกรณ์ที่มีทรัพยากรจำกัด K ใน KVM หมายถึง กิโล โดยเรียกตามหน่วยความจำที่นับเป็นกิโลไบต์ ส่วนคอมพิวเตอร์แบบตั้งโต๊ะนั้นหน่วยนับเป็นเม็กกะไบต์ KVM เหมาะจะนำมาใช้กับหน่วยประมวลผล RISC/CISC แบบ 16/32 บิต ซึ่งมีหน่วยความจำทั้งหมดไม่กี่ร้อยกิโลไบต์เท่านั้นประมาณ 128 กิโลไบต์ปัจจุบัน KVM มีความต้องการหน่วยความจำอยู่ในช่วง 40 – 80 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 Mobile Information Device Profile

MIDP เป็นโปรไฟล์ที่เป็นส่วนหนึ่งของ Configuration CLDC โดยโปรไฟล์นี้จะเป็นมาตรฐานสำหรับอุปกรณ์พกพาไร้สายที่มีขนาดเล็กและมีทรัพยากรอยู่อย่างจำกัด ซึ่งจัดว่าเป็นโพรไฟล์ที่มีการพัฒนาอยู่ตลอดและถูกใช้ในการพัฒนาแอพพลิเคชันบนอุปกรณ์พกพากันอย่างแพร่หลาย

โพรไฟล์นี้จะทำให้สามารถพัฒนาแอพพลิเคชันบนอุปกรณ์พกพาหรืออุปกรณ์ที่มีการเชื่อมต่อไร้สายอย่างโทรศัพท์มือถือที่มีหน้าจอที่มีขนาดเล็ก มีแบตเตอรีและความสามารถของ CPU จำกัด โดยส่วนใหญ่แล้วอุปกรณ์เหล่านี้จะอำนวยความสะดวกในเรื่องของการสื่อสารด้วยเสียงและการให้บริการข้อมูลผ่านทางแอพพลิเคชันต่างๆ

โพรไฟล์นี้ที่มีการรวมเอาคอนฟิกูเรชัน CLDC เข้าด้วยกัน ดังนั้นจึงสามารถทำให้แอพพลิเคชันบนอุปกรณ์เหล่านี้เป็นที่ตอบสนองความต้องการบรรดาลูกค้าต่างๆที่ใช้อุปกรณ์พกพา ซึ่งโพรไฟล์นี้จะสามารถนำไปพัฒนาแอพพลิเคชันได้หลากหลายจุดประสงค์และอุปกรณ์หลากหลายชนิดเช่น การเชื่อมต่อกับข้อมูลต่างๆ เครือขายไร้สาย รวมถึงแอพพลิเคชันสำหรับการให้บริการต่างๆ และข้อมูลสื่อบันเทิงต่างๆ (ยกตัวอย่างเช่น ตารางคะแนนของกีฬา, ข้อมูลทางการเงิน e-commerce และเกม)

MIDP เป็นโพรไฟล์สำหรับอุปกรณ์ที่รองรับ Java ME และมีลักษณะเฉพาะดังต่อไปนี้

- Display:

  Screen-size: 96x54

  Display depth: 1-bit

  Pixel shape (aspect ratio): approximately 1:1

- Input:

  มีวิธีการในการ Input หนึ่งวิธีหรือมากกว่า เช่น keypad, Touch screen

- Memory:

  128 kilobytes สำหรับ MIDP components

  8 kilobytes สำหรับ application-created persistent data

  32 kilobytes สำหรับ Java runtime (e.g., the Java heap)

- Networking:

  เป็นชนิดไร้สาย , สองทิศทาง, ไม่ต่อเนื่องและถูกจำกัด Bandwidth

  Feature ต่างๆที่ MIDP ได้จัดเตรียมไว้ได้แก่

- User Interface จัดการเกี่ยวกับการแสดงผล

- Persistent storage จัดการเกี่ยวกับการเก็บข้อมูลและฐานข้อมูล

- Networks จัดการเกี่ยวกับการเชื่อมต่อเน็ตเวิร์ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Application life-cycle จัดการเกี่ยวกับลำดับขั้นตอนการทำงาน

- Event handling จัดการเกี่ยวกับอีเวนต์ต่างๆ

ซึ่งจะมี Package ที่สำคัญคือ

javax.microedition.midlet.* เป็น API ในการสร้างโปรแกรมหลัก

javax.microedition.lcdui.* เป็น API ในการจัดการ User Interface

javax.microedition.rms.* RMS (Record Management System) เป็นส่วนของการเก็บข้อมูลเช่นเดียวกับฐานข้อมูล

3.1.4 Personal Digital Assistant Profile

PDAP เป็นโพรไฟล์ที่เจาะจงสำหรับกลุ่มอุปกรณ์ประเภท Personal Digital Assistants (PDA) โดย PDAP จะเสมือนเป็นแพ็คเกจเสริมของพีดีเอบนแพลตฟอร์ม Java ME ที่ประกอบเป็นสองแพ็คเกจซึ่งทั้งสองแพ็คเกจจะเป็นถูกนำไปใช้และเป็นส่วนหนึ่งของ Configuration CLDC

แพ็คเกจเหล่านี้จะรวมเอาจุดเด่นต่างๆที่สามารถพบได้ในหลายๆพีดีเอและโทรศัพท์มือถือ ซึ่งแพ็คเกจเหล่านี้ประกอบไปด้วยสองส่วนคือ

1. Personal Information Management (PIM) - แพ็คเกจนี้จะช่วยให้อุปกรณ์ Java ME ในการเข้าถึงการจัดการข้อมูลส่วนบุคคลของผู้ใช้ โดยข้อมูลที่ถูกเข้าถึงนี้จะถูกบรรจุอยู่ในส่วนต่างๆ คือ ข้อมูล Address book, To do lists, Calendar event management เป็นต้น รวมถึงแพ็คเกจนี้จะช่วยในการแชร์ข้อมูลระหว่าง desktop PCs หรือระหว่างพีดีเอเครื่องอื่นๆ

2. File Connection (FC) – แพ็คเกจนี้จะสามารถทำให้อุปกรณ์ Java ME เข้าถึง file systems ที่อยู่ในอุปกรณ์พกพา ในการใช้งานเบื้องต้นของ API นี้คือการอนุญาตให้เข้าไปยังส่วนเก็บข้อมูลไฟล์ที่สามารถลบทิ้งได้ เช่น memory card

พีดีเอที่สามารถรองรับโพรไฟล์นี้จะต้องรองรับ Java ME CLDC และมีลักษณะเฉพาะอย่างต่ำดังต่อไปนี้

- Display

  Resolution 128x128 pixels ขั้นต่ำ

  1 Bit ความลึกของหน้าจอเป็นอย่างต่ำ

- Input

  มีอุปกรณ์สำหรับเลื่อนและชี้ตำแหน่ง เช่น ปากกา

  มีความสามารถ Input ตัวอักษรได้

- Memory

  Memory ทั้งหมด (ROM + RAM) 1000 KB เป็นอย่างต่ำ และมีที่สำหรับ Java runtime และlibrary

- Power

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีการจำกัดพลังงาน เช่น แบตเตอรี

- Networking

เป็นชนิดไร้สาย , สองทิศทาง, ไมต่อเนื่อง และถูกจำกัด Bandwidth

3.1.5 TinyLine SVG

TinyLine เป็นแอพพลิเคชันที่ใช้จัดการเกี่ยวกับการแสดงรูปภาพ SVG บนอุปกรณ์พกพาที่พัฒนาขึ้นด้วยภาษาจาวา TinyLine SVG มีฟังก์ชันการทำงานต่างๆมากมาย ไม่ว่าจะเป็นการแสดงรูปภาพที่เป็นไฟล์ .svg การย่อ ขยาย เลื่อนและหมุนรูป และแก้ไขการแสดงผล เป็นต้น

3.1.6 คุณสมบัติพื้นฐานของ TinyLine

1. รองรับแพลตฟอร์มแบบ SVG Tiny
2. รองรับรูปแบบ SVG ประเภท fonts, image และ text elements, paths, animations และ event เป็นต้น
3. รองรับขนาดของไฟล์ SVG ประมาณ 100 Kbytes
4. สามารถทำงานได้บนจาวา SDK แบบ CLDC 1.0
5. รองรับไฟล์ SVG แบบ textual และ gzip
6. ใช้งานกับ Java ME ได้ง่าย

3.1.7 แพ็คเกจต่างๆ ของ TinyLine

TinyLine นั้นประกอบด้วยแพ็คเกจ 2 แพ็คเกจด้วยกันคือ com.tinyline.svg และ com.tinyline.tiny2d ซึ่งแพ็คเกจทั้งสองแพ็คเกจนั้นใช้บางส่วนของคลาสใน CLDC 1.0 เท่านั้น โดยคลาสในจาวาที่ TinyLine ใช้มีดังนี้คือ

java.lang.Exception

java.io.InputStream

java.lang.Object

java.lang.System

java.lang.Throwable

3.1.8 รายละเอียดของคลาสในแพ็คเกจทั้งสองและอินเทอร์เฟส

1. แพ็คเกจ Com.tinyline.tiny2d

ความหมายของคลาสต่างๆ ที่เป็นพื้นในกราฟิค 2 มิติอย่างเช่น color, transformation, point และเส้นทาง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.1 แสดงคลาสต่างๆใน tinyline2d

TinyBitmap	เป็นคลาสที่ใช้แทนอาเรย์ของพิกเซลของสี่เหลี่ยมมุมฉากและรูป
TinyColor	เป็นคลาสที่กำหนดสีในพื้นที่สีแบบ ARGB
TinyFont	เป็นคลาสที่ประกอบด้วยกลุ่มของคลาส TinyGlyph กับข้อมูลที่สำคัญในการใช้กริด
TinyGlyph	เป็นคลาสที่ใช้แทนคอนเทนของหน่วยของเรนเดอร์ที่ใช้ในฟอร์น
TinyMatrix	เป็นคลาสที่ใช้แทนเมตริกซ์จำลองการแปลงรูปสองมิติ ซึ่งแสดงการแมพเส้นตรงจากพิกัดหนึ่งไปยังพิกัดหนึ่ง ซึ่งคง “straightness” และ “parallelness” ของเส้นไว้
TinyPath	เป็นคลาสที่ใช้แทนเส้นของรูปเรขาคณิตจากเส้นตรง, สี่เหลี่ยมจตุรัส, เส้นโค้งและลูกบาศก์
TinyPixbuf	เป็นคลาสที่ใช้แทนอาเรย์ของพิกเซลของสี่เหลี่ยมมุมฉาก
TinyPoint	เป็นคลาสที่ใช้แทนตำแหน่งพิกัด (x,y) ในพื้นที่ที่กำหนด
TinyRect	เป็นคลาสที่ใช้กำหนดพื้นที่ในพื้นที่ตำแหน่งที่ถูกล้อมรอบด้วยจุดบนซ้าย (xmin,ymin) และจุดล่างขวา(xmax,ymax) ของออฟเจ็ค TinyRect ในพื้นที่ระนาบ
TinyTransform	เป็นคลาสที่อิมพลีเมนท์ส่วนประกอบย่อย(เช่น ‘scale(..)’ หรือ ‘matrix(...)’)ในแอทริบิวต์ transform

ตาราง 3.2 แสดงคลาสในการกำหนด basic data types, data structure และ utilitie

TinyHash	เป็นคลาสที่อิมพลีเมนท์ hashtable ซึ่งแมพคีย์กับตัวแปร
TinyNumber	เป็นคลาสที่อิมพลีเมนท์รูปแบบพื้นฐานประเภท ‘number’
TinyParsers	เป็นคลาสที่ใช้แทน parser ที่รองรับรูปแบบพื้นฐานของ TinyLine 2D และออฟเจ็ค
TinyString	เป็นคลาสที่ใช้แทนสตริงค์
TinyUtil	เป็นคลาสที่บรรจุเมทธอดสำหรับแสดง numeric operations พื้นฐาน
TinyVector	เป็นคลาสที่อิมพลีเมนท์อาเรย์ของออฟเจ็ค

2. แพ็คเกจ Com.tinyline.svg

แพ็คเกจ com.tinyline.svg รองรับ SVG Tiny ตั้งแต่ 1.1 ขึ้นไป ซึ่งหมายความว่า TinyLine รองรับโมดูล gradient และโมดูล opacity ซึ่งเป็นความสามารถสำคัญของ SVG Tiny 1.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.3 คลาสที่เป็นอินเทอร์เฟสของแพ็คเกจ com.tinyline.svg

AnimationCallback	อินเทอร์เฟสนี้จะทำการย้ายอีเวนย์ SMIL จาก animation element ไปยังปาร์ตี้ที่สนใจ
ImageLoader	เป็นอินเทอร์เฟสที่ใชแก้ไขความแตกต่างของเมทธอดในการโหลดภาพแรสเตอร์ อันเนื่องมาจากความแตกต่างของโปรไฟล์ในจาวา
SVGImageProducer	เป็นอินเทอร์เฟสสำหรับออฟเจ็ค ซึ่งสามารถสร้างข้อมูลรูปแบบ SVGโดย TinyLine SVG API รองรับรูปสองโมเดลคือ โมเดลแบบ producer/consumer (push) โมเดลแบบ via pixel buffer ถ้าแอพพลิเคชันต้องการโมเดลแบบ producer/consumer ต้องทำการออิมพลีเมนท์อินเทอร์เฟส SVGImageProducer
XMLHandler	เป็นอินเทอร์เฟสสำหรับเรียก XML
XMLParser	เป็นอินเทอร์เฟสสำหรับการเฟสเอกสาร XML

ตาราง 3.4 คลาสที่ใช้จัดการโปรเซสการเฟสของ SVG Tiny

SVG	เป็นคลาสที่ใช้กำหนดค่าคงที่ของ SVG Tiny อย่างเช่น element, attributeและค่าที่ใช้สำหรับกำหนดรายละเอียดของ SVG Tiny
SVGAttr	เป็นคลาสที่อิมพลีเมนท์เฟสของ SVGT attribute
SVGDocument	เป็นคลาสที่อิมพลีเมนท์เฮดเดอร์ของออฟเจ็คในออฟเจ็คของ SVGNode
SVGParser	เป็นคลาสที่อิมพลีเมนท์ XMLHandler สำหรับเรียกเอพีไอสำหรับเฟสในอินพุทสตรีมของ SVGT
SVGRaster	เป็นคลาสที่จัดการออฟเจ็คที่เป็นทรีของ SVGNode บนพิคเซลบัฟเฟอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

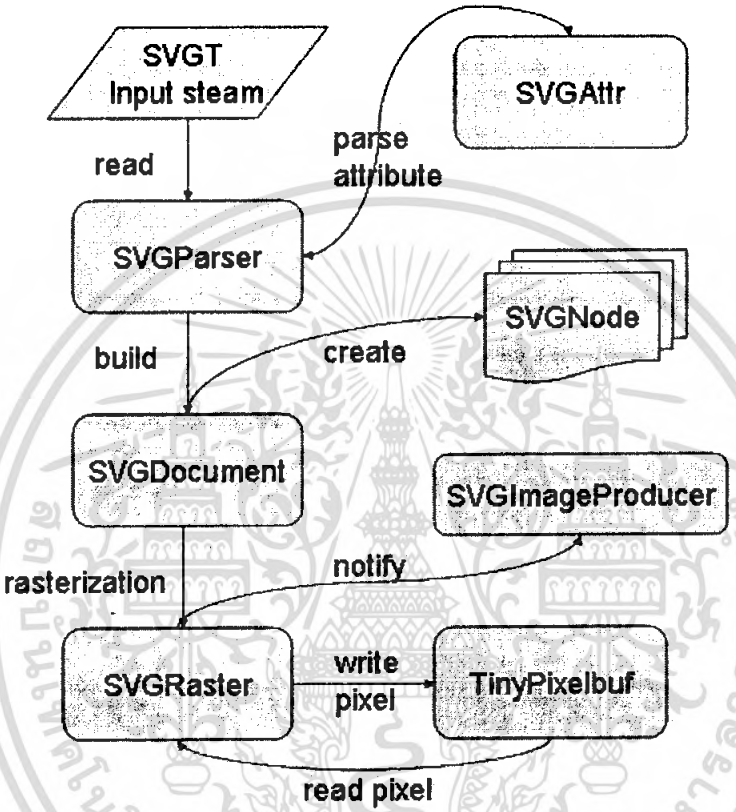
ตาราง 3.5 คลาสที่ในส่วนสุดท้ายที่ใช้ติดต่อกับ SVGT element

SMILTime	เป็นคลาส datatype ซึ่งทำหน้าที่แทนเวลาใน timegraph
SVGAnimationElem	เป็นคลาสที่อิมพลีเมนท์ animation element ซึ่งกำหนดใน SMIL Animation และโปรไฟล์ 3GPP SMIL Language
SVGEllipseElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ ‘circle’และ‘ellipse’
SVGFontElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'font'
SVGFontFaceElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'font-face'
SVGGlyphElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'glyph'
SVGGradientElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'linearGradient' หรือ 'radialGradient'
SVGGroupElem	เป็นคลาสที่อิมพลีเมนท์คอนเทนเนอร์อิรีเมนทร์
SVGImageElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'image'
SVGLineElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'line'
SVGMPathElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'mpath'
SVGNode	เป็นคลาสที่อิมพลีเมนท์คลาสพื้นฐานสำหรับอิรีเมนทร์ทั้งหมดในภาษา SVG
SVGPathElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'path'
SVGPolygonElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'polyline' และ 'polygon'
SVGRect	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'rectangle'
SVGRectElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'rect'
SVGSVGElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'svg'
SVGStopElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'stop'
SVGTextElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'text'
SVGUnknownElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'unknown'
SVGUseElem	เป็นคลาสที่อิมพลีเมนท์อิรีเมนทร์ 'use'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.9 ขั้นตอนการทำงานของ TinyLine

จากรูป 2.13 นั้นเป็นการแสดงขั้นตอนการทำงานของแอพพลิเคชัน Viewer ที่ใช้วาดรูปที่เป็น SVG Tiny ซึ่งจะแสดงว่าอินพุทสตรีมของ SVG Tiny เปลี่ยนเป็นทรีของ SVGNode และแสดงออกมาบนหน้าจอได้อย่างไร



รูปที่ 3.3 แสดงขั้นตอนการทำงานของ TinyLine

ในเริ่มแรกนั้น SVGPaser จะทำการสร้าง OM ภายใน (ทรีของ SVGNode) ขึ้นมาจากอินพุทของสตรีม SVGT โดย SVGDocument จะมีทรีรูทของ SVGNode เป็นเมมเบอร์ ต่อจากนั้น SVGRaster จะเปลี่ยนออฟเจ็ค SVGNode ให้อยู่ในรูปกราฟิคแบบธรรมดาที่จาวารองรับและวาดไปบนออฟเจ็ค TinyPixelbuf ซึ่งอินเทอร์เฟส SVGImageProducer จะทำการแจ้งว่าพิกเซลใหม่ในออฟเจ็ค TinyPixelbuf พร้อมที่จะแสดงบนหน้าจอแล้ว

จากรูปจะมีขั้นตอนดังต่อไปนี้

1. ขั้นตอนการ init มีดังนี้

    1.1. ทำการสร้างบัฟเพอร์ของพิกเซล

```
TinyPixbuf buffer = new TinyPixbuf(width, height);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2. สร้างออฟเจ็ค SVGRaster

raster = new SVGRaster(buffer);

1.3. สร้าง SVGT image producer/consumer

imageProducer = new MIDPSVGImageProducer(raster);

imageProducer.setConsumer(this);

raster.setSVGImageProducer(imageProducer);

1.4. กำหนดอิรีเมนตร์ ImageLoader

SVGImageElem.setImageLoader(this);

2. ขั้นตอนการเฟส

2.1. สร้างเอกสาร SVGT เปล่า

SVGDocument doc = raster.createSVGDocument();

2.2. สร้างเฟสของ SVGT attribute ซึ่งต้องการกำหนดรายละเอียดของบัพเพอร์พิกเซล

SVGAttr attrParser = new SVGAttr(pixbuf.width, pixbuf.height);

2.3. สร้างเฟสของสตรีม SVGT

SVGParser parser = new SVGParser(attrParser);

2.4. ทำการเฟสอินพุทสตรีม SVGT ที่ชื่อว่า 'is' ไปบนเอกสาร SVGT ที่ชื่อว่า'doc'

parser.load(doc,is);

3. ขั้นตอนการ rasterizing

3.1. เซ็ตเอกสาร SVGT สำหรับวาด

raster.setSVGDocument(document);

3.2. ทำการ invalidate คลิบปัจจุบัน

raster.invalidate();

3.3. อัพเดตพิกเซลภายใต้คลิบปัจจุบัน ในขั้นตอนนี้ออฟเจ็ค SVGNode ทั้งหมดจะถูกวาดไปบนบัฟเฟอร์ของพิกเซล

raster.update();

3.4. ทำการเรียกอินเทอร์เฟส SVGImageProducer เพื่อบอกว่ามีพิกเซลใหม่

raster.sendPixels();

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การเขียนโปรแกรมรับค่าพิกัดจีพีเอสด้วย Java ME

การที่อุปกรณ์พกพาไม่ว่าจะเป็นพีดีเอหรือโทรศัพท์มือถือ จะทำการรับข้อมูลพิกัดจากระบบจีพีเอสได้นั้น อุปกรณ์พกพาจำเป็นที่จะต้องมีการเชื่อมต่อกับอุปกรณ์รับข้อมูลจีพีเอส (GPS Receiver)

อุปกรณ์รับข้อมูลจีพีเอสแต่ละชนิดจะมีวิธีการส่งข้อมูลมายังอุปกรณ์พกพาที่แตกต่างกันไป ซึ่งโดยส่วนใหญ่ข้อมูลจะถูกส่งผ่านมาทาง Bluetooth และทาง Serial Port ดังนั้น ผู้พัฒนาจำเป็นต้องทำการติดต่อกับอุปกรณ์รับข้อมูลจีพีเอส โดยทำการเชื่อมต่อไปยัง Bluetooth Port หรือ Serial Port โดยใช้จาวาแพคเกจดังต่อไปนี้

java.io

javax.microedition.io

3.2.1 การสถาปนาการเชื่อมต่อ

1. การเชื่อมต่อด้วย BlueTooth

สำหรับการเชื่อมต่อไปยังอุปกรณ์รับข้อมูลจีพีเอสชนิดที่ติดต่อกับอุปกรณ์พกพาผ่านทาง Bluetooth จำเป็นต้องใช้ Device Address ของอุปกรณ์รับข้อมูลมากำหนดเป็น URL สำหรับการเชื่อมต่อ โดย URL ประกอบไปด้วยสามส่วนดังนี้ รายการ URL (URL Scheme) ,Device Addess ของอุปกรณ์ Bluetooth และหมายเลข Port

String url = "btspp://0002c727fc56:1";

ทำการสร้างการเชื่อต่อจาก URL โดยใช้คลาส StreamConnection จากนั้นใช้คลาส InputStreamReader เพื่อเปิดการเชื่อมต่อโดยการเรียกเมธอดOpenInputStream ดังตัวอย่าง

StreamConnection connection = (StreamConnection) Connector.open(url, Connector.READ);

InputStreamReader reader = new InputStreamReader (connection.openInputStream());

2. การเชื่อต่อผ่าน Serial Port

สำหรับการเชื่อมต่อกับอุปกรณ์รับข้อมูลชนิดที่ติดตั้งภายในตัวอุปกรณ์โดยส่วนใหญ่แล้วจะมีการส่งข้อมูลมาทาง Serial Port ดังนั้นจึงต้องมีการใช้ URL เพื่อการเชื่อมต่อไปยัง Serial Port โดยมี รายการURL และค่าพารามิเตอร์ต่างๆดังต่อไปนี้

comm:<port identifier>[<optional parameters>]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.6 แสดงค่าพารามิเตอร์ของ URL String ที่ใช้เปิดการเชื่อต่อ Serial Port

พารามิเตอร์	ค่าปกติ	ความหมาย
baudrate	ขึ้นอยู่แต่ละ platform	ความเร็วการส่งข้อมูลของ Port
Bit per char	8	จำนวนของบิตต่อตัวอักษร (มีค่า 7 หรือ 8)
stopbits	1	จำนวนของ stop bit ต่อตัวอักษร

ตาราง 3.7 (ต่อ) แสดงค่าพารามิเตอร์ของ URL String ที่ใช้เปิดการเชื่อต่อ Serial Port

parity	none	ค่า parity (สามารถมีค่าเป็น odd , even หรือ none)
blocking	on	เมื่อเปิดการใช้งานจะรับข้อมูลจน buffer เต็ม
autocts	on	เมื่อเปิดการใช้งานจะรอ cts line ก่อนเขียนข้อมูล
autorts	on	เมื่อเปิดการใช้งานจะเปิด rts line ขณะที่ buffer ยังไม่เต็ม

```
String url = "comm:COM4;baudrate=4800;bitsperchar=8";
StreamConnection  myconnection = (StreamConnection) Connector.open(url, Connector.READ);
InputStreamReader reader = new
InputStreamReader(myconnection.openInputStream());
```

3.2.2 การอ่านข้อมูล

ตัวอักษรที่ถูกส่งมาจากอุปกรณ์รับข้อมูลจีพีเอสจะถูกอ่านอย่างเรียงกันเป็นลำดับ ผู้พัฒนาจะต้องทำการเก็บค่าไปจนกว่าจะพอตัวอักษร Return (ASCII รหัส 13) ซึ่งแทนความหมายการจบบรรทัดของประโยคข้อมูล และการนำ String ที่อ่านค่าได้ไปใช้งาน ควรมีการตัดตัวอักษรสุดท้ายที่เป็น Line Feed (ASCII รหัส 10) ออกไป ซึ่ง Output String ที่ได้จะคือ NMEA – 0183 ซึ่งจะนำไปวิเคราะห์และการตัดแบ่งคำต่อไป

```
String output;
Int input;
while ((input = reader.read()) != 13)
{
    output += (char) input;
}
output = output.substring(1, output.length() - 1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 เทคโนโลยี Java Servlet

Java Servlet เป็นแอพพลิเคชันที่ทำงานอยู่ฝั่งเซิร์ฟเวอร์ ซึ่งมีหลักการคล้ายกับเวปแอพพลิเคชันแบบ CGI (Common Gateway Interface) แต่ถูกพัฒนาให้ดีกว่าในหลายๆส่วน โดยส่วนแรกก็คือใช้ภาษาจาวาในการพัฒนา ซึ่งภาษาจาวานั้นใช้หลักการของObject Oriented ในการเขียน ทำให้สามารถลดความซับซ้อนของโครงสร้างโปรแกรม รวมไปถึงความสามารถในการนำส่วนของโปรแกรมที่เขียนไว้แล้วมาใช้ใหม่ได้ นอกจากนี้จาวายังเป็นภาษาที่เป็นลักษณะแบบ platform independent โดยจะทำการสร้าง 1 thread ต่อหนึ่ง request ซึ่งต่างจากเวปแอพพลิเคชันแบบ CGI ที่จะสร้าง 1 process ต่อหนึ่ง request ซึ่งจะทำให้เปลืองทรัพยากรมากกว่าและ process ในการรันก็จะช้ากว่าด้วยและจุดเด่นที่สำคัญของ Servlet ก็คือ API (Application Programming Interface) โดยระบบที่พัฒนาโดยใช้หลักการของ Servlet จะสามารถเรียกใช้ API ที่ทางจาวาให้มาได้ด้วย (javax.servlet.*, javax.servlet.http.*) ซึ่งจะช่วยทำให้การพัฒนาระบบดังกล่าวง่ายและเร็วยิ่งขึ้น

3.3.1 ขั้นตอนการทำงานของ Servlet Server

1. ในขั้นแรกนั้นไคลเอนต์(Web browser) จะทำการเชื่อมต่อกับเวปเซิร์ฟเวอร์และส่ง HTTP request มาให้เวปเซิร์ฟเวอร์
2. เวปเซิร์ฟเวอร์จะทำการรับ request จากไคลเอนต์และส่งต่อไปให้ Servlet container เพื่อทำการส่งต่อให้ servlet ตัวที่จัดการ request นั้นต่อไป โดยตัว Servlet container สามารถที่จะรันใน process เดียวกับโฮสเวปเซิร์ฟเวอร์ได้
3. Servlet container ทำการพิจารณาว่า servlet ตัวไหนที่เป็นตัวจัดการ request ที่ส่งมา โดยจะเรียก servlet นั้นผ่านทางออบเจ็ค request และ response
4. ตัว servlet นั้นจะใช้ออบเจ็ค request ในการหาว่ายูสเซอร์ที่รีโมทเข้ามาเป็นใคร, พารามิเตอร์ของ HTTP POST ที่อาจส่งมากับ request เป็นอะไรและข้อมูลอื่นๆที่เกี่ยวข้อง ต่อจากนั้น servlet จะทำงานตามที่ได้โปรแกรมไว้และแปลงข้อมูลเพื่อที่จะส่งกลับไปให้ไคลเอนต์ผ่านทางออบเจ็ค response
5. หลังจากที่ servlet จัดการ request เสร็จแล้ว ตัว servlet จะตรวจสอบให้แน่ใจก่อนว่าส่งข้อมูลไปให้ไคลเอนต์เสร็จเรียบร้อยแล้วจึงจะส่งการควบคุมกลับคืนไปให้โฮสเวปเซิร์ฟเวอร์ควบคุมต่อไป

3.3.2 เปรียบเทียบระหว่างเทคโนโลยี Servlets กับเทคโนโลยีอื่น

โดยทางเทคนิคแล้ว servlet อยู่ระหว่างโปรแกรมแบบ Common Gateway Interface (CGI)กับ server extension อย่างเช่น Netscape Server API (NSAPI) หรือ Apache Module ซึ่งตัว Servlet มีข้อดีที่ต่างจาก server extension อื่นตรงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โดยทั่วๆไปแล้ว servlet มีความเร็วในการทำงานที่มากกว่า CGI script เนื่องจากมีโมเดลการทำงานที่แตกต่างกัน
- ตัว servlet ใช้ API มาตรฐานซึ่งรองรับการทำงานบนเวปเซิร์ฟเวอร์หลายประเภท
- ตัว servlet มีความสามารถของภาษาจาวาทั้งหมด อีกทั้งยังง่ายต่อการพัฒนาและไม่ขึ้นกับแพลตฟอร์มใดๆอีกด้วย
- สามารถเข้าถึง API ที่มีขนาดใหญ่ซึ่งเป็นความสามารถของแพลตฟอร์มจาวา

3.3.3 Servlet Life Cycle

Servlet มีการกำหนดขั้นตอนการทำงานที่ดี โดยมีการกำหนดว่าจะโหลดและจอง, กำหนดค่าเริ่มต้น, ดูแล request จากไคลเอนต์และการออกจากเซอร์วิสด์อย่างไร โดยกระบวนการทำงานมีดังนี้คือ กำหนดค่าเริ่มต้น, ให้บริการและทำลายอินเตอร์เฟสของเมธอด javax.servlet.Servlet ซึ่งทุกๆ servlet จะต้องทำไม่ว่าจะเป็นทางตรงหรือทางอ้อมโดยผ่านทางคลาส GenericServlet หรือ HttpServlet

Loading and Instantiation

servlet container จะรับผิดชอบในการโหลดและการจองของตัว servlet ซึ่งจะเกิดขึ้นเมื่อ container เริ่มขึ้นหรือจนกว่า container จะกำหนดว่า servlet ต้องการบริการจากการร้องขอ

เมื่อ servlet เริ่มขึ้น ตัวคลาส servlet จะต้องมีการระบุตำแหน่งด้วย servlet container ซึ่ง servlet container จะใช้คลาสจาวาธรรมดาในการโหลดคลาสของ servlet ซึ่งในการโหลดนั้นอาจมาจากไฟล์ซิสเต็มภายใน, ไฟล์ซิสเต็มภายนอกหรือบริการทางเครือข่ายอื่นๆ

Initialization

หลังจากออบเจ็ค servlet ถูก instantiate แล้ว ตัว container จะต้องกำหนดค่าเริ่มต้นของservlet ก่อนที่จะจัดการ request จากไคลเอนต์ตัว container จะทำการกำหนดค่าเริ่มต้นให้ servlet ได้โดยการเรียกเมธอด init ในอินเตอร์เฟสของ servlet ด้วยออบเจ็คพิเศษซึ่งจะอิมพลีเมนท์อินเตอร์เฟสในส่วน ServletConfig

Request Handling

หลังจากที่ servlet ทำการกำหนดค่าเริ่มต้นแล้ว servlet container อาจจะใช้มันในการจัดการการร้องขอจากไคลเอนต์โดยการร้องขอที่เข้ามานั้นจะถูกดูแลโดยออบเจ็ค request ในชนิด ServletRequest ตัว servlet จะทำการตอบสนองการร้องขอโดยการเรียกเมธอดที่เรียกว่า ServletResponse ซึ่งออบเจ็คนี้จะทำการพาสพารามิเตอร์ไปยังเมธอดที่ให้บริการของอินเตอร์เฟสของ servlet

ในกรณีของการร้องขอของ HTTP ออบเจ็คจะถูกจัดการโดย container ในประเภท HttpServletRequest และ HttpServletResponse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thread Safety

ในการจัดเตรียมออบเจ็ค request และ response ไว้ไม่ได้เป็นการการันตีว่าเธรดจะปลอดภัย ซึ่งหมายความว่ามันสามารถสามารถใช้ได้ในเธรดด้านการร้องขอเท่านั้น

การอ้างอิงถึงออบเจ็ค request และ response ไม่สามารถเรียกใช้ได้จากเธรดอื่น ถ้าถูกเธรดสร้างโดยแอพพลิเคชันที่ใช้ออบเจ็ค container-managed อย่างเช่นออบเจ็ค request และออบเจ็ค response ซึ่งออบเจ็คเหล่านั้นจะถูกเข้าถึงเฉพาะภายในช่วงชีวิตของเซอร์วิสใน servlet เท่านั้น ซึ่งการที่เธรดมีช่วงชีวิตอยู่ในช่วงชีวิตของเซอร์วิสใน servlet เพราะว่าการเข้าถึงออบเจ็คเหล่านั้นหลังจากออบเจ็คเซอร์วิสจบไปแล้วอาจทำให้เกิดปัญหาได้

End of Service

ตัว servlet container ไม่ได้ถูกร้องขอให้เก็บ servlet ในการโหลดแบบเฉพาะเจาะจงในช่วงเวลาใดเวลาหนึ่ง ตัวอย่างของ servlet อาจจะถูกเก็บใน servlet container เป็นเวลาไม่ถึงช่วงเสี้ยววินาทีตลอดช่วงชีวิตของ servlet container

ในช่วงที่ servlet container จะสิ้นสุดลงนั้น servlet จะถูกย้ายออกไปจากเซอร์วิส มันจะเรียกเมธอด destroy ของอินเตอร์เฟสของ Servlet เพื่อที่จะให้ servlet ปล่อยทรัพยากรต่างๆที่มันกำลังใช้อยู่และทำการเซฟสถานะขณะนั้น ซึ่งก่อนที่ servlet container จะเรียกเมธอด destroy นั้นมันจะต้องรอให้เธรดที่กำลังรันอยู่ในเมธอดเซอร์วิสของ servlet ทำงานเสร็จเรียบร้อยก่อนแล้วจึงเรียกเมธอด destroy ขึ้นมา ซึ่งในขณะที่เรียกเมธอด destroy ขึ้นมาในนั้นอาจจะไม่รับ request ไปให้ servlet นั้น ถ้า container ต้องการให้ servlet นั้นรับ request อีกครั้งจะต้องทำการสร้างคลาสของ servlet นั้นขึ้นมาให้ก่อน จึงจะสามารถรับ request ได้อีกครั้ง

3.3.4 Apache Tomcat

Apache Tomcat เป็นเวปคอนเทนเนอร์หรือแอพพลิเคชันเซิร์ฟเวอร์ ซึ่งพัฒนาโดย Apache Software Foundation (ASF) ตัว Apache Tomcat รองรับ Java Servlet และ JavaServer Pages (JSP) โดย Apache Tomcat มีการพัฒนาให้ใช้ภาษาจาวาในการทำงานร่วมกับเวปเซิร์ฟเวอร์ อีกทั้งยังมีเครื่องมือที่ช่วยให้ปรับแต่งตัวเซิร์ฟเวอร์ให้ง่ายขึ้นอีกด้วย แต่ผู้ใช้ก็สามารถปรับแต่งตัวเซิร์ฟเวอร์ได้ด้วยการแก้ไขไฟล์คอนฟิกูเรชันซึ่งเป็นไฟล์ที่อยู่ในรูปแบบ XML ได้ ในปัจจุบันเวอร์ชันล่าสุดของ Apache Tomcat เป็นเวอร์ชัน 6.0.14 ซึ่งรองรับ Servlet 2.5 และ JSP 2.1

3.4 PostgreSQL

3.4.1 PostgreSQL คืออะไร

PostgreSQL คือ ระบบจัดการเชิงวัตถุ-สัมพันธ์ (Object-Relational Database Management System หรือ ORDBMS) ซึ่งปรับปรุงมาจากต้นแบบระบบฐานข้อมูล POSTGRES 4.2 ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มหาวิทยาลัยแคลิฟอเนีย ที่ เบอร์กเลย์ ภายใต้ความควบคุมของ Professor Michael Stonebreaker โดยได้รับเงินวิจัยสนับสนุนจาก the Defense Advanced Research Project Agency (DARPA), the Army Research Office (ARO), the National Science Foundation (NSF) และ ESL, Inc PostgreSQL เป็นโปรแกรมรหัสเปิด (Open-Source) สามารถนำไปใช้งานได้โดยไม่มีค่าใช้จ่าย และสนับสนุนมาตรฐาน SQL: 2003 และมีความสามารถต่างๆ เพิ่มขึ้นมาอีกมาก POSTGRES ได้นำเอาแนวคิดของ Object-Relational มาใช้ ซึ่งปัจจุบันก็เป็นต้นแบบของระบบฐานข้อมูลที่นำมาทำเพื่อการค้ามากมาย และเป็นธรรมดาที่ Relational Database Management Systems (RDBMS) จะมีการสนับสนุนรูปแบบของข้อมูลที่ประกอบไปด้วยกลุ่มของข้อมูลที่มีความสัมพันธ์กัน แต่ PostgreSQL ได้มีการเพิ่มเติมสิ่งที่น่าสนใจเพิ่มขึ้นทำให้ผู้ใช้งานสามารถจัดการการขยายระบบของตนเองได้ง่ายขึ้นดังนี้

- Inheritance
- Data Types
- Functions

นอกจากนี้ PostgreSQL ยังได้มีการเพิ่มเติมความสามารถให้ระบบมีประสิทธิภาพและมีความยืดหยุ่นเพิ่มขึ้นหลายอย่าง เช่น

- Constraints
- Triggers
- Rules
- Transactional Integrity

ซึ่งความสามารถเหล่านั้นนำไปให้ PostgreSQL ถือว่าเป็นฐานข้อมูลประเภท Object-Relational

3.5 สถาปัตยกรรมการทำงานของ PostgreSQL

สถาปัตยกรรมมาตรฐานสำหรับระบบที่ทำงานบนระบบเครือข่าย คือ สถาปัตยกรรมที่เรียกว่า Client-Server ซึ่งในสถาปัตยกรรมนี้ประกอบด้วย

- เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ซึ่งจะมี Process ทำงานอยู่ ซึ่งจากนี้ไปจะขอเรียกว่า Server Process
- เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอนต์ซึ่งจะมี Process ทำงานอยู่ ซึ่งจากนี้ไปจะขอเรียกว่า Client Process

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Server Process จะทำหน้าที่รอรับการติดต่อจาก Client Process และทำงานให้บริการแก่ Client Process โดยทั่วไป Server Process ยังแบ่งตามลักษณะช่วงเวลาที่ Client Process ติดต่อกับ Server Process ซึ่งสามารถออกเป็น 2 ประเภท คือ

- Iterative Server ใช้สำหรับที่สามารถประมาณช่วงเวลาได้ ตัวอย่างเช่น httpd process ในเวปเซิร์ฟเวอร์ซึ่งให้บริการช่วงสั้นๆ ในการส่งข้อมูล Web Page ให้แก่ Web Browser
- Concurrent Server ใช้กับระบบที่ไม่สามารถประมาณช่วงเวลานั้นได้ไคลเอนต์อาจติดต่อเป็นช่วงเวลาสั้นๆ หรือนานมากในการขอบริการ ระบบจัดการฐานข้อมูลทำงานในลักษณะนี้

PostgreSQL ทำงานภายใต้โมเดลของสถาปัตยกรรมแบบ Server Process โดย Server Process ทำงานในลักษณะ Concurrent Server ซึ่งมีขั้นตอนดังต่อไปนี้

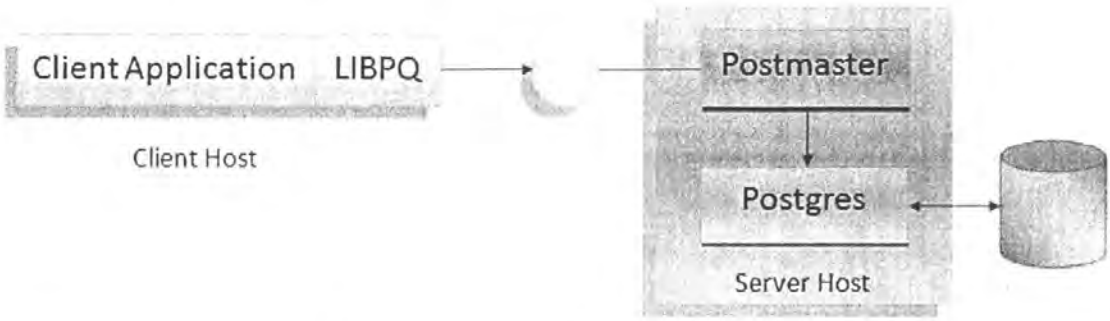
Server Process ที่ชื่อว่า Postmaster ถูกเรียกให้เริ่มต้นทำงานบนคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ ทำการเปิดช่องการติดต่อและแจ้งกับระบบปฏิบัติการที่จะรับคำขอบริการจากไคลเอนต์ที่พอร์ต ที่กำหนดไว้ โดยทั่วไป Postmaster กำหนด Port เริ่มต้นไว้ที่หมายเลข 5432

Postmaster จะเข้าสู่สภาวะ Sleep เพื่อรอการเรียกจาก Client Process โดยที่ Client Process จากเครื่องคอมพิวเตอร์ลูกข่าย (หรือแม้จากเซิร์ฟเวอร์เองก็ได้) ติดต่อขอรับบริการจากไลบรารี LIBPQ มาที่ Port ที่กำหนดไว้ ดังแสดงในรูป

รูปที่ 3.4 แสดง Client Process ขอบริการ Postmaster

เมื่อ Postmaster ได้รับการติดต่อจาก Client Process แล้วจะทำการสร้าง Process ลูก คือ postgres เพื่อให้บริการแทนตัวเองดังแสดงในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

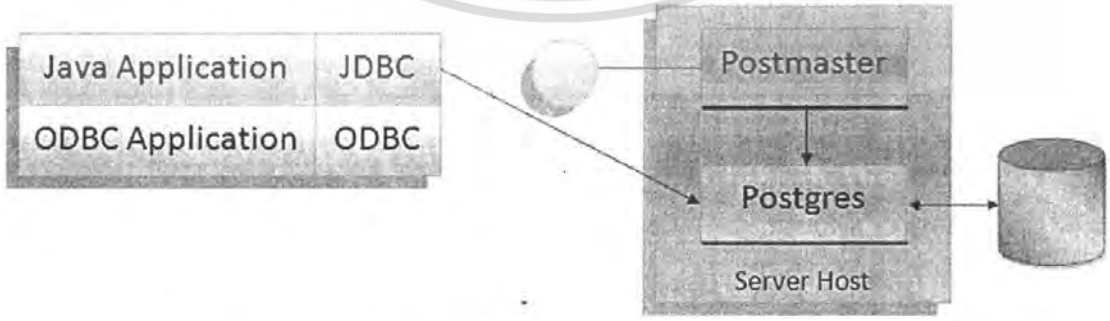


รูปที่ 3.5 แสดง Postmaster สร้าง Postgres Process

หลังจากนั้น Postmaster จะรอรับการขอบริการจาก Client Process อื่นต่อไป ในขณะที่ postgres process เมื่อให้บริการจนจบ ก็จะปิดตัวเองดังแสดงในรูป

รูปที่ 3.6 แสดง Postgres Process ให้บริการแก่ Client Process

PostgreSQL ทำงานในลักษณะ per-user process โดยทั้ง postmaster และ postgres process ทำงานด้วย user-id ของ postgres super user (postgres super user ไม่จำเป็นต้องเป็น user ที่ชื่อ postgres แต่เป็น user ที่ติดตั้ง PostgreSQL)



รูปที่ 3.7 แสดงการขอบริการเซิร์ฟเวอร์ผ่านอินเตอร์เฟส JDBC,ODBC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LIBPQ เป็นไลบรารีมาตรฐานสำหรับโปรแกรมไคลเอนต์(ที่เขียนด้วย C Program) ในการติดต่อกับเซิร์ฟเวอร์ PostgreSQL อย่างไรก็ตาม เราสามารถที่จะติดตั้งซอฟต์แวร์มาตรฐานอื่น เช่น ODBC หรือ JDBC ดังแสดงในรูป เพื่อให้ Client Program ต่างๆที่ใช้อินเตอร์เฟสมาตรฐานเหล่านี้สามารถติดต่อกับ PostgreSQL ได้เช่นกัน

3.6 การใช้เทคโนโลยี Java ติดต่อกับฐานข้อมูล

ในการติดต่อกับฐานข้อมูล PostgreSQL ด้วยการใช้ Java Servlet จะต้องมีการติดตั้งแพ็คเกจสำหรับติดต่อกับฐานข้อมูลหรือที่เรียกว่า JDBC Driver โดยจะต้องใส่ไว้ในตัวแอพพลิเคชันก่อน ซึ่งในที่นี้ใช้ JDBC4 เวอร์ชัน 8.2-507 เมื่อทำการติดตั้งแพ็คเกจเรียบร้อยแล้วจะสามารถเรียกใช้ไดรเวอร์ได้ ซึ่งขั้นตอนสำคัญในการเขียนโปรแกรมสำหรับติดต่อกับฐานข้อมูลด้วยภาษาจาวามีดังนี้

1. ทำการโหลดคลาสของไดรเวอร์

```
Class.forName("com.somejdbcvendor.TheirJdbcDriver");
Class.forName("org.postgresql.Driver");
```

2. สร้างการเชื่อต่อกับฐานข้อมูล

```
Connection conn = DriverManager.getConnection(
 "jdbc:somejdbcvendor:other data needed by some jdbc vendor",
 "myLogin",
 "myPassword" );

Connection conn =DriverManager.getConnection(
"jdbc:postgresql://localhost:5432/GPS_BASE_GAME",
"admin",
"jame");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สร้างสเตทเมนท์

```
Statement symt = cn.createStatement();    //สร้างแอปเจ็กต์ stmt สำหรับจัดการของ
                                          SQL
```

4. ประมวลผลด้วยภาษา SQL และรับผลลัพธ์กลับมาเป็น result set

```
String  query =  SQL;
ResultSet rs = stmt.executeQuery(query);  //สร้างออปเจ็กต์ rs เป็นกลุ่มเรคคอร์ด
                                          // บรรจุข้อมูล
Rs.next();                                //ข้อมูลในเรคอร์ดถัดไป
```

5. ปิดการเชื่อมต่อกับฐานข้อมูล

```
Stmt.close();
cn.close();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบระบบ

4.1 การออกแบบเกม

การออกแบบโครงงานชิ้นนี้นำหลักการของ Location – Based Service มาออกแบบเป็นโปรแกรมเกมเพื่อความบันเทิง โดยนำความรู้ทางด้านระบบสารสนเทศภูมิศาสตร์ และระบบระบุพิกัดด้วยจีพีเอสมาประยุกต์ใช้ในเกม

4.1.1 เนื้อเรื่องของเกม

ในอนาคตอันใกล้นี้ ด้วยปัญหาสภาวะโลกร้อนที่สะสมกันมาอย่างช้านานจนถึงขั้นวิกฤต ส่งผลให้น้ำแข็งที่ขั้วโลกทั้งสองต่างละลายจนไปหมดสิ้น แผ่นดินทวีปทั้ง 7 ต่างจมอยู่ใต้ท้องทะเลอันกว้างใหญ่ไพศาล อารยธรรมและเทคโนโลยีของมนุษย์ชาติล่มสลาย ผู้คนต่างใช้ชีวิตบนผืนทะเล และอยู่อาศัยบนเศษซากอารยธรรมที่หลงเหลือโผล่พ้นผืนทะเลขึ้นมา สิ่งเหล่านั้นคือตึกอาคารที่ถูกต่อเติมขึ้นไปเป็นเมืองต่างๆให้เหล่าผู้คนอยู่อาศัย ภายใต้ยุคแห่งความวุ่นวายนี้ สงครามและการแย่งชิงอาณาเขตเกิดขึ้นทุกหนทุกแห่ง โดยเรือที่ถูกสร้างขึ้น เหล่ากลุ่มคนที่ต่อสู้บนท้องทะเลด้วยเรือติดอาวุธเหล่านี้ถูกเรียกว่า “โจรสลัด”

ผู้เล่นจะสวมบทบาทเป็นหัวหน้าของกองเรือโจรสลัด ออกผจญภัยไปในท้องทะเล โดยมีการใช้แผนที่ที่สร้างขึ้นโดยอ้างอิงกับอาคารในโลกของความเป็นจริง และใช้ตำแหน่งของผู้เล่นจริงๆในขณะที่เล่น

4.1.2 วัตถุประสงค์ของเกม

ผู้เล่นจะต้องทำการแข่งขันกับผู้เล่นอื่นด้วยการวางแผนทำสงคราม สะสมสินค้าเพื่อทำการค้าขายระหว่างเมือง อัพเกรดเรือโจรสลัด และบังคับเรือของผู้เล่นไปต่อสู้กับเรือของผู้เล่นอื่น โดยจะมีคะแนนสะสมเมื่อมีการต่อสู้ชนะ

4.1.3 มุมมองภายในเกม

เป็นมุมมองบนหน้าจอของอุปกรณ์พอกเกตพีซีหรือโทรศัพท์มือถือ ขนาดหน้าจอ 240 x 320 พิกเซล ในลักษณะของแผนที่ประกอบไปด้วยข้อมูลผู้เล่นต่างๆ และวัตถุที่เคลื่อนที่บนแผนที่ ที่มีการเปลี่ยนตำแหน่งอยู่ตามเวลาจริง ประกอบกับจะมีหน้าจอของการทำกิจกรรมต่างๆบนแผนที่ หน้าจอการต่อสู้ระหว่างเรือ และหน้าจอการทำกิจกรรมต่างๆภายในเมือง

4.1.4 กฎ กติกา และการเล่นเกม

1. ผู้เล่นเป็นเจ้าของเรือโจรสลัดอย่างน้อยหนึ่งลำ โดยเริ่มแรก จะมีเรือมาให้ผู้เล่น 1 ลำ
2. ผู้เล่นทำการบังคับเรือให้ไปในสถานที่ต่างๆบนแผนที่ โดยจะอ้างอิงตำแหน่งของผู้เล่นจากตัวรับจีพีเอสบนอุปกรณ์พกพา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อเรือของผู้เล่นอยู่ใกล้สถานที่ที่เป็นเมือง ผู้เล่นสามารถทำกิจกรรมต่างๆภายในเมืองได้ เช่น ทำการซื้อขายสินค้าระหว่างเมือง โดยการรับซื้อสินค้าจากเมืองหนึ่งไปขายอีกเมืองหนึ่งเพื่อเกร็งกำไร การซ่อมเรือ การอัพเกรดเรือ

4. ผู้เล่นสามารถนำเรือไปโจมตีเรือของผู้เล่นอื่น ซึ่งเมื่อโจมตีเรือสำเร็จ ผู้เล่นสามารถ ยึดสินค้าที่บรรทุกอยู่บนเรือนั้นได้

5. ในการต่อสู้จะเป็นการต่อสู้กันโดยอัตโนมัติ ผลัดกันโจมตีสลับกันเป็นรอบๆไปจนกว่าเรือของฝ่ายใดฝ่ายหนึ่งจะมีค่าพลังเท่ากับศูนย์

6. ในการต่อสู้จะมีการคำนวณโอกาสการโจมตีโดนออกมาเป็นเปอร์เซ็นต์ และจะมีการสุ่มค่าตัวเลขออกมาตั้งแต่ค่า 0 ถึง 100 เพื่อใช้ในการกำหนดว่าการโจมตีนั้นๆโดนหรือไม่โดน ซึ่งค่าโอกาสสำเร็จนี้จะขึ้นอยู่กับความแม่นยำของอาวุธ ความแม่นยำของเรือ และความเร็วของเรือ โดยจะมีการคำนวณผลของการโจมตีออกมาอย่างอัตโนมัติ ในทุกรอบของการโจมตี

7. ผู้เล่นแต่ละคนจะมีแต้มคะแนนเพื่อใช้ในการแข่งขันจัดอันดับคะแนน โดยคะแนนของผู้เล่นจะเพิ่มขึ้นจากการใช้เรือต่อสู้ชนะ และจำนวนทรัพยากรที่ขายได้กำไรเป็นต้น เป็นต้น

4.1.5 รายละเอียดของวัตถุในเกม

สำหรับวัตถุในเกม จะประกอบไปด้วย เรือ เมือง อาวุธ และทรัพยากร โดยแต่ละวัตถุของเกมจะมีความแตกต่างกันไปด้วยค่าพารามิเตอร์ วัดค่าคุณสมบัติต่างๆของวัตถุนั้น และถูกใช้ในการเล่นเกมต่อไป

เรือหนึ่งลำมีรายละเอียดดังนี้

ตาราง 4.1 แสดงค่าพารามิเตอร์ของเรือ

พารามิเตอร์	**ความหมาย**	**หมายเหตุ**
Ship Name	ชื่อของเรือ	-
Position	ตำแหน่งของเรือ	พิกัดของเรือบนแผนที่
HP	ความคงทนของเรือ	มีค่าไม่เกิน HP Max จะแพ้การต่อสู้เมื่อค่านี้มีค่าเป็น 0
HP Max	ความคงทนของเรือสูงสุด	-
Defend	ความแข็งแกร่งของเรือ	ใช้ลดความเสียหายในการต่อสู้
Weight Limit	น้ำหนักที่เรือบรรทุกได้	ใช้ในการขนส่งสินค้าบนเรือ เรือบรรทุกสินค้าที่มีน้ำหนักรวมได้ไม่เกินค่านี้
Ship Accurate	ความแม่นยำ	ใช้ในการคำนวณโอกาสการโจมตีสำเร็จ
Speed	ความเร็ว	ใช้ในการคำนวณโอกาสการหลบการโจมตีสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สินค้าชนิดต่างๆมีดังต่อไปนี้

1. Food (อาหาร) เป็นสินค้าที่ถูกใช้ในการเดินเรือไปยังสถานที่ต่างๆ
2. Crew (ลูกเรือ) เป็นกึ่งแรงงานกึ่งสินค้าที่ถูกนำไปใช้ในการต่อสู้ระหว่างเรือ และการยึดเมือง
3. Ore (สินแร่) เป็นสินค้าที่ใช้ในการพัฒนาเมือง การสร้างอาวุธ และการสร้างเรือ
4. Jewel (อัญมณี) เป็นสินค้าที่ใช้ในการพัฒนาเมือง และ การค้าขายระหว่างเมือง

เมืองแต่ละเมืองมีรายละเอียดดังต่อไปนี้

ตาราง 4.2 แสดงค่าพารามิเตอร์ของเมือง

พารามิเตอร์	**ความหมาย**	**หมายเหตุ**
Town Name	ชื่อเมือง	-
Position	ตำแหน่งของเมือง	พิกัดของเมืองบนแผนที่
Food Grow Rate	อัตราการผลิตอาหาร	อัตราการผลิตสินค้าอาหารของเมือง
Crew Grow Rate	อัตราการผลิตลูกเรือ	อัตราการผลิตสินค้าลูกเรือของเมือง
Ore Grow Rate	อัตราการผลิตสินแร่	อัตราการผลิตสินค้าสินแร่ของเมือง
Jewel Grow Rate	อัตราการผลิตอัญมณี	อัตราการผลิตสินค้าอัญมณีของเมือง
Max Weight	ค่าน้ำหนักที่เมืองเก็บสินค้าได้	เก็บสินค้าไว้ในเมืองได้โดยน้ำหนักต้องไม่เกินค่านี้

อาวุธแต่ละชนิดมีรายละเอียดดังต่อไปนี้

ตาราง 4.3 แสดงค่าพารามิเตอร์ของอาวุธ

พารามิเตอร์	**ความหมาย**	**หมายเหตุ**
Weapon Name	ชื่อของอาวุธ	-
Atk Power	พลังโจมตี	ใช้คำนวณความเสียหายแก่เรือที่ถูกโจมตี
Accurate	ความแม่นยำ	ใช้ในการคำนวณโอกาสการโจมตีสำเร็จ
Max Ammo	จำนวนกระสุนสูงสุด	จำนวนครั้งที่สามารถใช้อาวุธชิ้นนี้ได้ในแต่ละครั้งของการต่อสู้
Critical Rate	อัตราการโจมตีรุนแรง	โอกาสที่จะสามารถโจมตีรุนแรงเป็นสองเท่า มีค่าไม่เกิน 100
Sale Value	ราคาขาย	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 การออกแบบกิจกรรมต่างๆในเกม

กิจกรรมต่างๆบนแผนที่และภายในเกมมีดังต่อไปนี้

การเดินเรือ

ในการเล่นเกม ผู้เล่นจะเสมือนบังคับเรือโจรสลัดที่แล่นอยู่ในทะเล โดยจะแสดงอยู่บนแผนที่ที่จำลองมาจากแผนที่จริง และถูกเก็บไว้บนพีดีเอ หรือโทรศัพท์มือถือ จากนั้น เมื่อผู้เล่นต้องการจะทำการเคลื่อนตำแหน่งของเรือโจรสลัดภายในเกมไปยังตำแหน่งที่ผู้เล่นต้องการ ผู้เล่นจะต้องทำการเดินทางไปยังตำแหน่งนั้นๆด้วยการเดินทางไปยังสถานที่จริง โดยตำแหน่งของเรือโจรสลัดจะเคลื่อนที่ตามค่าตำแหน่งพิกัดที่อ่านได้จากจีพีเอสทำให้เสมือนว่า ผู้เล่นได้ทำการเดินเรือโจรสลัดไปยังตำแหน่งต่างๆบนแผนที่ในเกม

การต่อสู้ระหว่างเรือ

เมื่อผู้เล่นบังคับเรือไปใกล้เรือของผู้เล่นอื่น ผู้เล่นสามารถสั่งให้เรือของผู้เล่นนั้นไปโจมตีเรือของผู้เล่นอื่นได้ โดยเมื่อการโจมตีเริ่มขึ้น จะเป็นการโจมตีสลับกันไปมาระหว่างเรือโจรสลัดสองลำ จะมีการคำนวณโอกาสการโจมตีโดนจาก ความเร็วของเรือ และความแม่นยำของอาวุธ มีการคำนวณค่าความเสียหายจากค่าพลังโจมตีของอาวุธ อัตราการเกิดการโจมตีที่รุนแรง และพลังป้องกันของเรือ เพื่อนำค่าความเสียหายที่ได้ ไปลดค่าความคงทนของเรือ(hp) สำหรับเรือโจรสลัดที่ถูกโจมตีจนค่าความคงทนเหลือ 0 ก่อนจะเป็นผู้แพ้ และถูกแย่งชิงทรัพยากรบนเรือไปยังเรือผู้ชนะ

การออกแบบอัลกอริทึมการต่อสู้

- จะทำการคำนวณค่าโอกาสโจมตีโดน (Hit) เป็นตัวเลข Integer ด้วยการคำนวณดังต่อไปนี้

$$Hit = Accurate - Speed(d) + Speed(a)$$

Hit = โอกาสโจมตีโดน

Accurate = ความแม่นยำของอาวุธที่ใช้โจมตี

Speed(d) = ความเร็วของเรือที่ป้องกัน

Speed(a) = ความเร็จของเรือที่โจมตี

- ทำการสุ่มค่าตัวเลขในของเขตจาก 0 ถึง 100 แล้วนำไปเปรียบเทียบกับค่าโอกาสโจมตีโดน (Hit) โดย ถ้า ค่าที่สุ่มได้ มากกว่า โอกาสโจมตีโดน (Hit) แล้วจะถือว่าการโจมตีครั้งนั้นพลาด (Miss) และ ถ้า ค่าที่สุ่มได้ น้อยกว่าหรือเท่ากับค่าโอกาสโจมตีโดน (Hit) แล้วจะถือว่าการโจมตีนั้นโดน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อโจมตีโดนจะมีการคำนวณค่าความเสียหาย (Dammage) โดยมีสูตรการคำนวณดังต่อไปนี้

Damage = Weapon Attack – Defend(d)*2

Damage = ค่าความเสียหาย ที่จะนำไปลดค่าความคงทน (HP) ของเรือ

Weapon Attack = ค่าพลังโจมตีของอาวุธ

Defend(d) = ค่าความแข็งแกร่งของเรือที่ป้องกัน

- เมื่อเรือโจมตีโดน จะมีการคำนวณโอกาสการเกิดการโจมตีอย่างรุนแรง โดยจะคำนวณจากค่า อัตราการโจมตีรุนแรง (Critical) นำไปเปรียบเทียบกับค่าที่สุ่มได้ในขอบเขต 0 ถึง 100 โดย ถ้า ค่าที่สุ่มได้ มากกว่า อัตราการโจมตีรุนแรง (Critical) แล้วจะถือว่าการโจมตีครั้งนั้นเป็นการโจมตีธรรมดาและถ้าค่าที่สุ่มได้น้อยกว่าหรือเท่ากับ อัตราการโจมตีรุนแรง (Critical) แล้ว ค่าความเสียหาย (Damage) จะเป็น 1.5 เท่า

กิจกรรมภายในเมือง

สำหรับตึกอาคารต่างๆที่เป็นสถานที่จริง จะถูกจำลองให้เป็นเมืองภายในเกม โดยเมื่อผู้เล่นทำการเดินเรือเข้าไปในบริเวณของตึก ผู้เล่นสามารถสั่งให้เรือเข้าไปเทียบท่าในเมืองได้ โดยภายในเมืองจะมีกิจกรรมให้ผู้เล่นทำดังต่อไปนี้

1 การซื้อและขายสินค้า

ภายในเมืองจะมีสินค้าขายในราคาที่ต่างกัน ผู้เล่นสามารถซื้อสินค้าเหล่านี้ แล้วนำไปขายอีกเมืองหนึ่งได้ โดยจะได้ผลกำไรจากส่วนต่างของราคาสินค้าของเมืองสองเมือง ซึ่งสินค้าแต่ละชนิด จะมีฐานราคากลาง และราคาที่ขาย ดังต่อไปนี้

ตาราง 4.4 แสดงสูตรคำนวณราคาขายสินค้า

ชนิดของสินค้า	ราคากลาง	สูตรคำนวณราคาขาย
Food	20	ราคากลาง - Food Grow Rate * 2
Jewel	130	ราคากลาง - Jewel Grow Rate * 6
Ore	30	ราคากลาง – Ore Grow Rate * 3
Crew	50	ราคากลาง

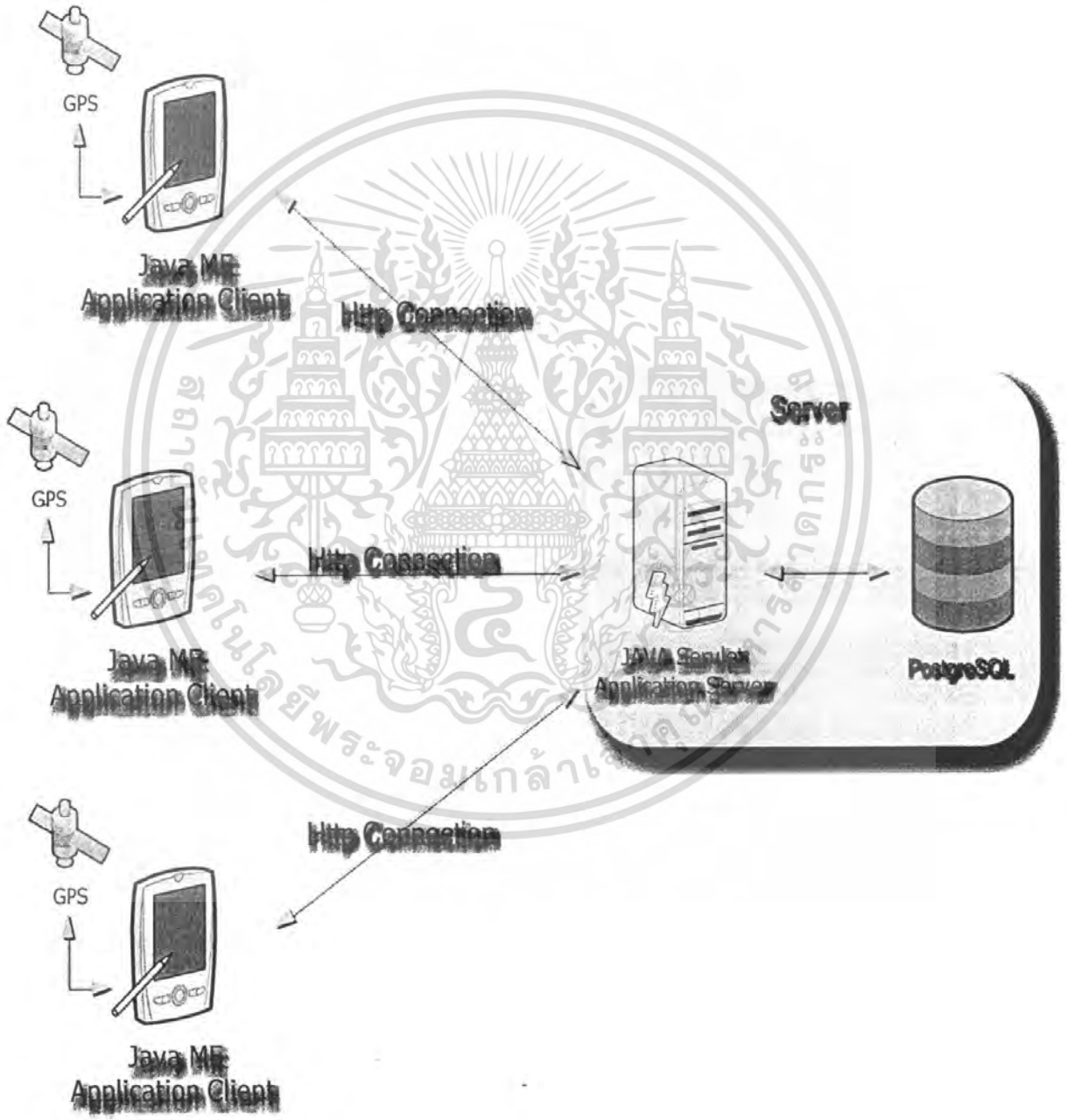
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 การซ่อมและอัพเกรดเรือ

ภายในเมืองจะมีอู่ซ่อมเรือ ให้ผู้เล่นซ่อมแซมความเสียหายของเรือ และสามารถเพิ่มความสามารถต่างๆของเรือได้

4.2 การออกแบบระบบ

4.2.1 การออกแบบสถาปัตยกรรมของระบบ

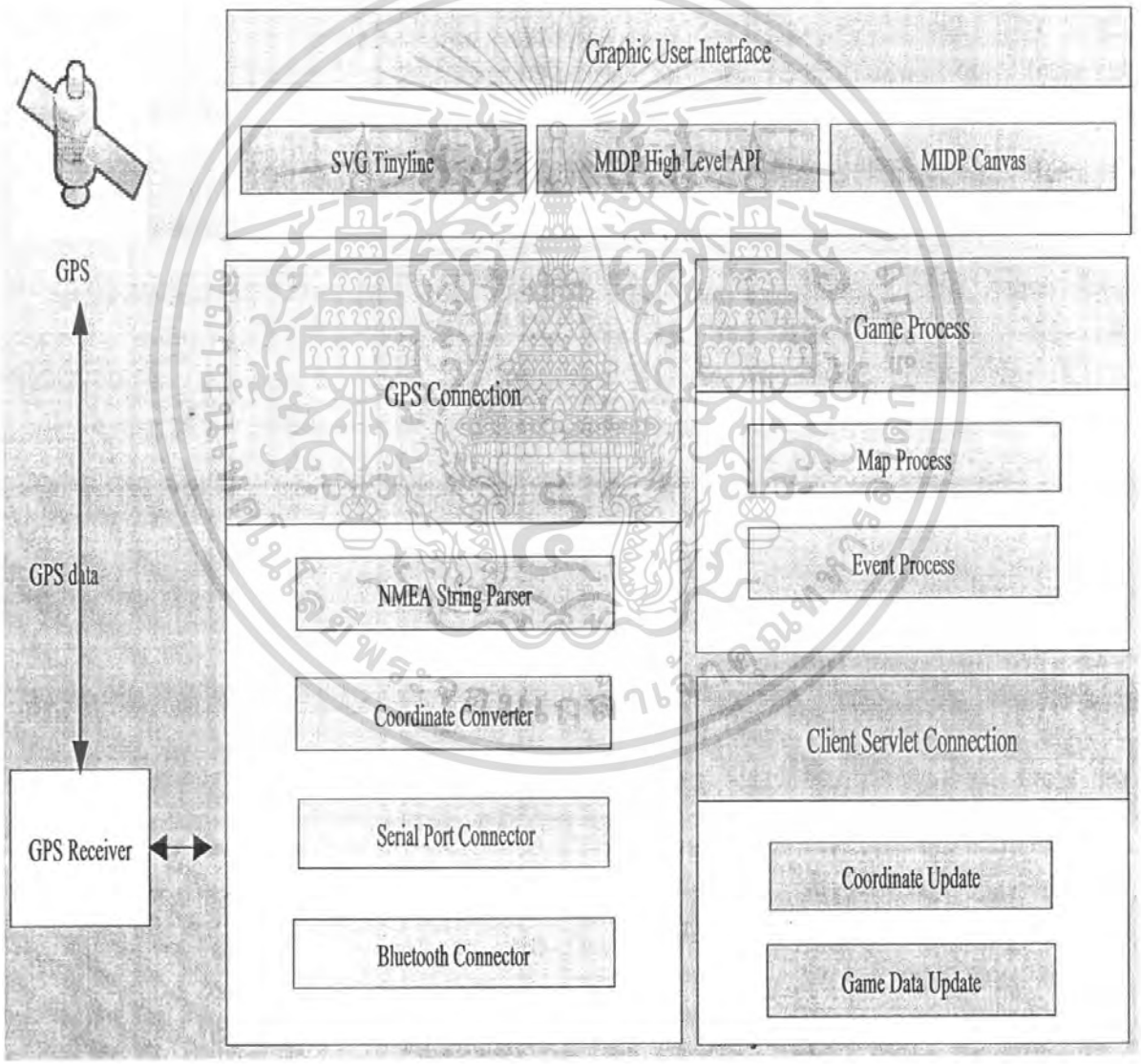


รูปที่ 4.1 แสดงสถาปัตยกรรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเกม จะแบ่งออกเป็นสองส่วนหลัก คือ ฝั่งไคลเอนต์ และเซิร์ฟเวอร์โดยทั้งสองส่วนติดต่อกันด้วย โปรโตคอล HTTP ใช้ในการแลกเปลี่ยนข้อมูล ทางส่วนไคลเอนต์จะพัฒนาโดยใช้ Java ME ทำหน้าที่หลักในการแสดงผลกราฟิกของเกม การระบุตำแหน่งของผู้เล่นโดยทำการเชื่อมต่อไปยังจีพีเอสการเรียกฟังก์ชัน Servlet เพื่อขอบริการข้อมูลต่างๆจากฝั่งเซิร์ฟเวอร์รวมถึงการส่งข้อมูลไปจัดเก็บลงยังฐานข้อมูล ซึ่งระบบฐานข้อมูลที่ใช้จะเป็นระบบฐานข้อมูล PostgreSQL ใช้ในการเก็บข้อมูลต่างๆ เช่น ข้อมูลตำแหน่งของผู้เล่น ข้อมูลตำแหน่งของสถานที่ต่างๆภายในเกม และข้อมูลรายละเอียดของตัวละครต่างๆในเกม เป็นต้น

4.2.2 สถาปัตยกรรมของ Client



รูปที่ 4.2 แสดงโครงสร้างของฝั่ง Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของไคลเอนต์ประกอบไปด้วย 4 ส่วนหลักๆ โดยแต่ละส่วนจะมีหน้าที่การทำงานดังต่อไปนี้

1. Graphic User Interface

ทำหน้าที่แสดงผลทางกราฟิกของเกม และทำหน้าที่ในการรับค่าคำสั่งต่างๆจากผู้เล่นในขณะเล่นเกม โดยแบ่งออกเป็นองค์ประกอบย่อยที่มีหน้าที่แตกต่างกันดังต่อไปนี้

SVG Tinyline เป็น JavaME Tool ที่ถูกนำมาใช้ในการแสดงผลภาพแผนที่ Vector ทำหน้าที่วาดแผนทีจากการอ่านไฟล์แผนที่ที่จัดเก็บไว้ภายในเครื่อง ซึ่งแผนที่จะถูกเก็บไว้ในลักษณะของไฟล์ SVG อีกทั้ง SVG Tinyline ยังมีหน้าที่ในการวาดกราฟิก 2D ของวัตถุต่างๆลงบนแผนที่ ไม่ว่าจะเป็น การวาดเรือโจรสลัดของผู้เล่น การวาดเรือโจรสลัดของผู้เล่นอื่นที่กำลังเล่นเกมอยู่ รวมถึงการระบุตำแหน่งของตึกสถานที่ ที่ถูกใช้เป็นสถานที่กิจกรรมภายในเกม เป็นต้น

MIDP High Level API เป็นแพกเกจ ที่ใช้สร้าง User Interface ระดับสูงของโปรไฟล์ MIDP โดยแพกเกจนี้จะถูกใช้ในการแสดงผลกราฟิกภายในเกมที่ไม่ซับซ้อน ส่วนแสดงผลที่ต้องการรับค่า Key Pad จากผู้เล่น เช่น Form, Text field, Choice group หรือ List เป็นต้น

MIDP Canvas เป็น Class ของโปรไฟล์MIDP ถูกใช้ในการแสดงผลจอภาพที่เป็นการวาดกราฟิกในระดับล่าง ถูกนำมาใช้ให้กราฟิกเกมมีความสวยงาม เช่น ใช้ในการวาดรูปภาพจากไฟล์ภาพต่างๆ การวาดภาพเคลื่อนไหว การวาดฉากBackground เป็นต้น

2 GPS Connection ทำหน้าที่ในการระบุตำแหน่งของผู้เล่นด้วยระบบจีพีเอสแบ่งออกเป็นองค์ประกอบย่อยที่มีหน้าที่แตกต่างกันดังต่อไปนี้

Serial Port Connector ทำหน้าที่เชื่อมต่อไปยัง Local Data Port ของเครื่องพีดีเอ หรือโทรศัพท์มือถือ ทำหน้าที่ในการอ่านข้อมูลจากตัวรับข้อมูลจีพีเอสที่อยู่ในตัวเครื่อง ซึ่งข้อมูลที่รับได้จะเป็นในรูปของฟอร์แมต NMEA

Bluetooth Connector ทำหน้าที่เชื่อมต่อกับ GPS Receiver ที่เป็นเครื่องแยกจากเครื่องพีดีเอ โดยทำหน้าที่เปิด Connection ผ่าน Bluetooth port อ่านข้อมูลในรูปของฟอร์แมต NMEA เช่นเดียวกันกับ Serial Port Connector

NMEA String Parser ทำหน้าที่แปลงข้อมูล String NMEA ให้เป็นข้อมูลพิกัดทางภูมิศาสตร์ ทำหน้าที่วิเคราะห์ชนิดของข้อมูล NMEA และตัดเอาใจความของข้อมูลพิกัด สามารถนำไปใช้ในการระบุตำแหน่งของผู้เล่นหรือเรือของผู้เล่นได้ โดยจะประกอบไปด้วย ฟังก์ชันในการตัดสตริง และส่วนวิเคราะห์ข้อมูล และเก็บข้อมูลพิกัด

Coordinate Converter ทำหน้าที่แปลงพิกัด Lattitude และ Longtitude ให้เป็นพิกัดที่ใช้ในแผนที่เกม ซึ่งแผนที่ในเกมจะใช้เป็นพิกัด UTM ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Game Data Process ทำหน้าที่รวบรวมข้อมูล และจัดการประมวลผลข้อมูลต่างๆของเกม ไม่ว่าจะเป็นการประมวลผล Event ต่างๆ ประมวลผลการเคลื่อนที่ของวัตถุบนแผนที่ หรือการคำนวณข้อมูลต่างๆภายเกม

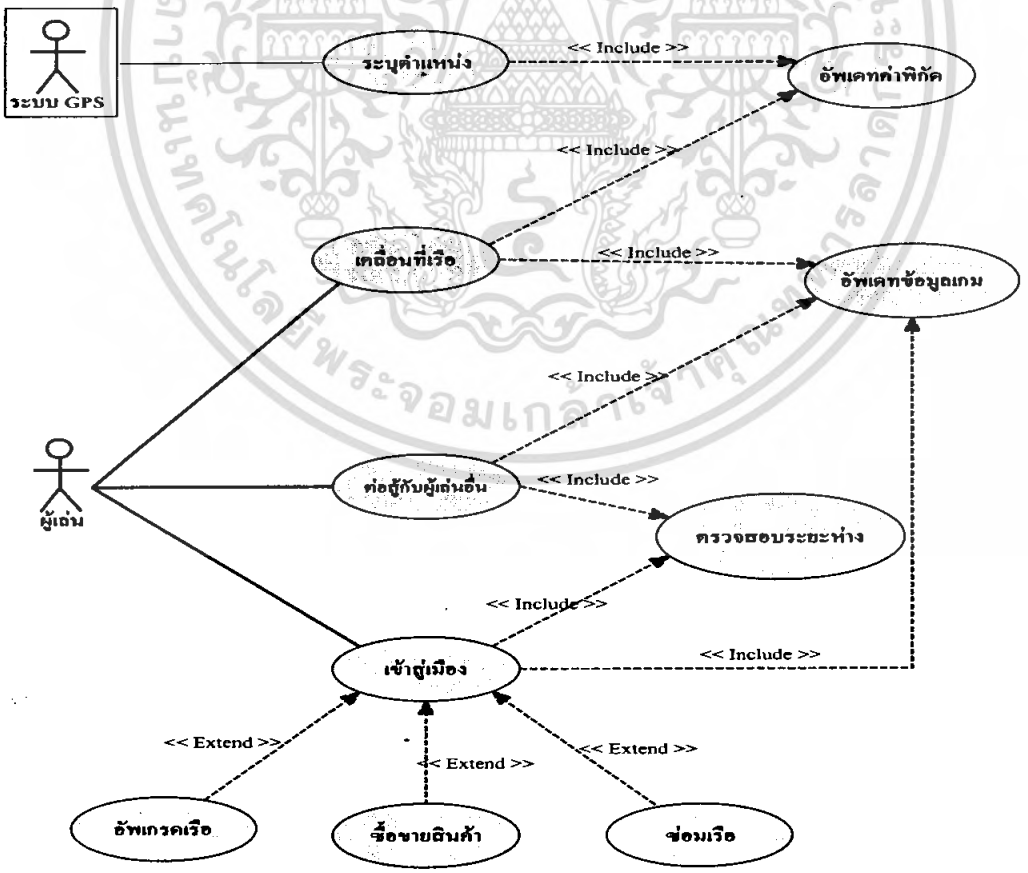
4 Servlet Connector ทำหน้าที่ในการเชื่อมต่อกับ Servlet ที่ Run อยู่ในทางฝั่งเซิร์ฟเวอร์ แบ่งออกเป็นสองส่วนหลักๆมีหน้าที่แตกต่างกันดังต่อไปนี้

Coordinate Update ทำหน้าที่ในการส่งพิกัดของผู้เล่นไปอัพเดตยังฐานข้อมูล อีกทั้งยังทำหน้าที่ในการโหลดค่าพิกัดของวัตถุต่างๆบนแผนที่จากฐานข้อมูลมาใช้ในการอัพเดทตำแหน่งบนแผนที่ต่อไป

Game Data Update ทำหน้าที่ในการร้องขอข้อมูลเกม ไม่ว่าจะเป็นข้อมูลผู้เล่น หรือข้อมูลรายละเอียดของวัตถุต่างๆในเกม การรับข้อมูลมาจัดเก็บลงคลาส หรือการส่งค่าจากคลาสไปจัดเก็บลงฐานข้อมูลเป็นต้น

4.3 การวิเคราะห์ระบบ

4.3.1 Use Case Diagram



รูปที่ 4.3 แสดง Use Case Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายของแต่ละ Use Case มีดังต่อไปนี้

Use Case : ระบุตำแหน่ง

Primary Actor : ระบบGPS

Description :

การระบุตำแหน่งของผู้เล่นด้วยจีพีเอสซึ่งจะเป็นการเปิดการเชื่อมต่อเพื่ออ่านค่าจาก GPS Receiver รวมถึงการแปลงพิกัด และการเก็บค่าพิกัดไว้ใช้ในการอัพเดทตำแหน่งของผู้เล่นต่อไป

Pre - Condition : เมื่อผู้เล่น Login เข้าสู่เกม

Main Flow : 1. สร้างการเชื่อมต่อกับ Serial port หรือ Bluetooth Port
2. รับข้อมูล NMEA มาวิเคราะห์หาข้อมูลพิกัด Latitude Longitude
3. แปลงค่าพิกัด Latitude Longitude เป็น x,y (UTM)
4. เก็บค่าพิกัดไว้ และอ่านค่าพิกัดใหม่ต่อไป

Use Case: เคลื่อนที่เรือ

Primary Actor: Player

Description :

ในการเล่นเกม เมื่อผู้เล่นต้องการเคลื่อนที่เรือในเกม ผู้เล่นจะต้องทำการเดินไปยังตำแหน่งที่ต้องการ โดยเมื่อจีพีเอสทำการระบุตำแหน่งแล้ว ตัวเกมจะทำการอัพเดทตำแหน่งเรือในเกมตามตำแหน่งที่ผู้เล่นนั้นได้เดินทางไปจริงๆ

Pre - Condition : เมื่อผู้เล่นเคลื่อนที่

Main Flow: 1. ขอข้อมูลตำแหน่งล่าสุดของผู้เล่น
2. วาดตำแหน่งของผู้เล่นลงบนแผนที่
3. ผู้เล่นเคลื่อนที่ไปยังตำแหน่งใหม่
4. รับค่าพิกัดจาก GPS
5. วาดตำแหน่งใหม่ของผู้เล่นลงบนแผนที่
6. ส่งข้อมูลตำแหน่งใหม่ของผู้เล่นไปอัพเดทยังฐานข้อมูล

Use Case: ต่อสู้กับเรือผู้เล่นอื่น

Primary Actor: Player

Description :

การต่อสู้จะเป็นในลักษณะของการโจมตีสลับกันไปมาระหว่างเรือสองลำ โดยเรือแต่ละลำจะค่าพารามิเตอร์แตกต่างกันตามการอัพเกรด และมีอาวุธที่ใช้ในการโจมตีแตกต่างกัน

Pre – Condition : เมื่อตำแหน่งของเรือ อยู่ใกล้ตำแหน่งของเรือศัตรู และผู้เล่นกดโจมตี

Main Flow: 1. ผู้เล่นสั่งคำสั่งให้เรือโจมตี เรือผู้เล่นอื่น
2. รับค่าพิกัดจากจีพีเอสแล้วส่งไปตรวจสอบระยะห่างกับ Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โหลดข้อมูลเรือทั้งของผู้เล่น และเรือศัตรู
4. แสดงรายละเอียดข้อมูลเรือ และฉากต่อสู้
5. คำนวณการโจมตี และค่าความเสียหายต่างๆ
6. แสดงภาพการโจมตี และแสดงค่าความเสียหาย
7. แสดงผลการต่อสู้ และอัพเดทข้อมูลเรือไปยังฐานข้อมูล

Use Case: เข้าสู่เมือง

Primary Actor: Player

Description :

การนำเรือของผู้เล่นไปจอดภายในเมือง เพื่อทำกิจกรรมต่างๆที่มีอยู่ภายในเมืองนั้นๆ ซึ่งตำแหน่งของเมืองจะถูกนำมาจากตำแหน่งของอาคารสถานที่ ของสถานที่จริง

Pre – Condition : ผู้เล่นเดินทางเข้าใกล้อาคารที่ถูกจำลองเป็นเมืองในเกม และกดคำสั่งจอดเรือ

Main Flow:
1. ผู้เล่นเลือกคำสั่งเข้าสู่เมือง
2. รับค่าพิกัดจากจีพีเอสแล้วส่งไปตรวจสอบระยะห่างกับ Server
3. โหลดข้อมูลผู้เล่น เรือของผู้เล่น และข้อมูลเมือง
4. แสดงรายละเอียดเมือง และรายการกิจกรรมต่างๆภายในเมือง

Use Case: ซื้อขายสินค้า

Primary Actor: -

Description :

กิจกรรมซื้อ และขายสินค้าชนิดต่างๆระหว่างเมือง โดย แต่ระเมืองจะมีราคาสินค้าที่แตกต่างกัน เป็นกิจกรรมที่ทำให้ผู้เล่นสามารถหาผลกำไรเป็นเงินในเกมได้ด้วยการเดินทางไปซื้อและขายสินค้าจากเมืองหนึ่ง ไปยังอีกเมืองหนึ่ง

Pre – Condition : เข้าสูเมือง

Main Flow:
1. ผู้เล่นเข้าสู่เมืองและเลือกกิจกรรมการซื้อขาย
2. แสดงรายการสินค้า จำนวนสินค้าคงเหลือ และราคาสินค้า
3. คำนวณและตรวจสอบการซื้อขายสินค้าจากข้อมูลเมือง
4. เมื่อซ้อขายสำเร็จ อัพเดทข้อมูลสินค้า และข้อมูลผู้เล่น ไปยังฐานข้อมูล
5. อัพเดทรายการจำนวนสินค้าที่แสดง

Use Case: ซ่อมเรือ

Primary Actor: -

Description :

การซ่อมแซมเรือที่เสียหาย

Pre – Condition : เข้าสู่เมือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Main Flow: 1. ผู้เล่นเข้าสู่เมืองและเลือกซ่อมเรือ

2. แสดงราคาค่าซ่อมเรือ และรายละเอียดของเรือ

3. เมื่อผู้เล่นยืนยันการซ่อมเรือ อัพเดทข้อมูลเรือและผู้เล่นไปยังฐานข้อมูล

Use Case: อัพเกรดเรือ

Primary Actor: -

Description :

การเพิ่มประสิทธิภาพของพารามิเตอร์ต่างๆของเรือ เพื่อนำไปใช่ในการต่อสู้

Pre – Condition : เข้าสู่เมือง

Main Flow: 1. ผู้เล่นเข้าสู่เมืองและเลือกอัพเกรดเรือ

2. แสดงรายละเอียดเรือปัจจุบัน และรายการการอัพเกรด

3. เมื่อผู้เล่นยืนยันการอัพเกรดเรือ อัพเดทข้อมูลเรือและข้อมูลผู้เล่นไปยังฐานข้อมูล

Use Case: อัพเดทค่าพิกัด

Primary Actor: -

Description :

การนำค่าพิกัดที่อ่านค่าได้จากจีพีเอสนำไปเปลี่ยนแปลงพิกัดของเรือผู้เล่น และทำการส่งค่าพิกัดล่าสุดไปเก็บยังฐานข้อมูล รวมถึงการรับค่าพิกัดของวัตถุอื่นๆมาวาดลงแผนที่ของเกม

Pre – Condition : -

Main Flow: 1 รับค่าพิกัดจากจีพีเอส

2. ส่งข้อมูลพิกัดล่าสุดผู้เล่นไปยังเซิร์ฟเวอร์

3. ร้องขอข้อมูลพิกัดอื่นๆมายัง Client

4. เปลี่ยนตำแหน่งของวัตถุชนิดต่างๆบนแผนที่

Use Case: อัพเดทข้อมูลเกม

Primary Actor: -

Description :

การส่งข้อมูลของเกมไปเก็บยังฐานข้อมูล หลังจากมีการเปลี่ยนแปลงข้อมูลของผู้เล่น ข้อมูลเรือ หรือข้อมูลของเมืองเป็นต้น

Pre – Condition :

จบกิจกรรมภายในเกม

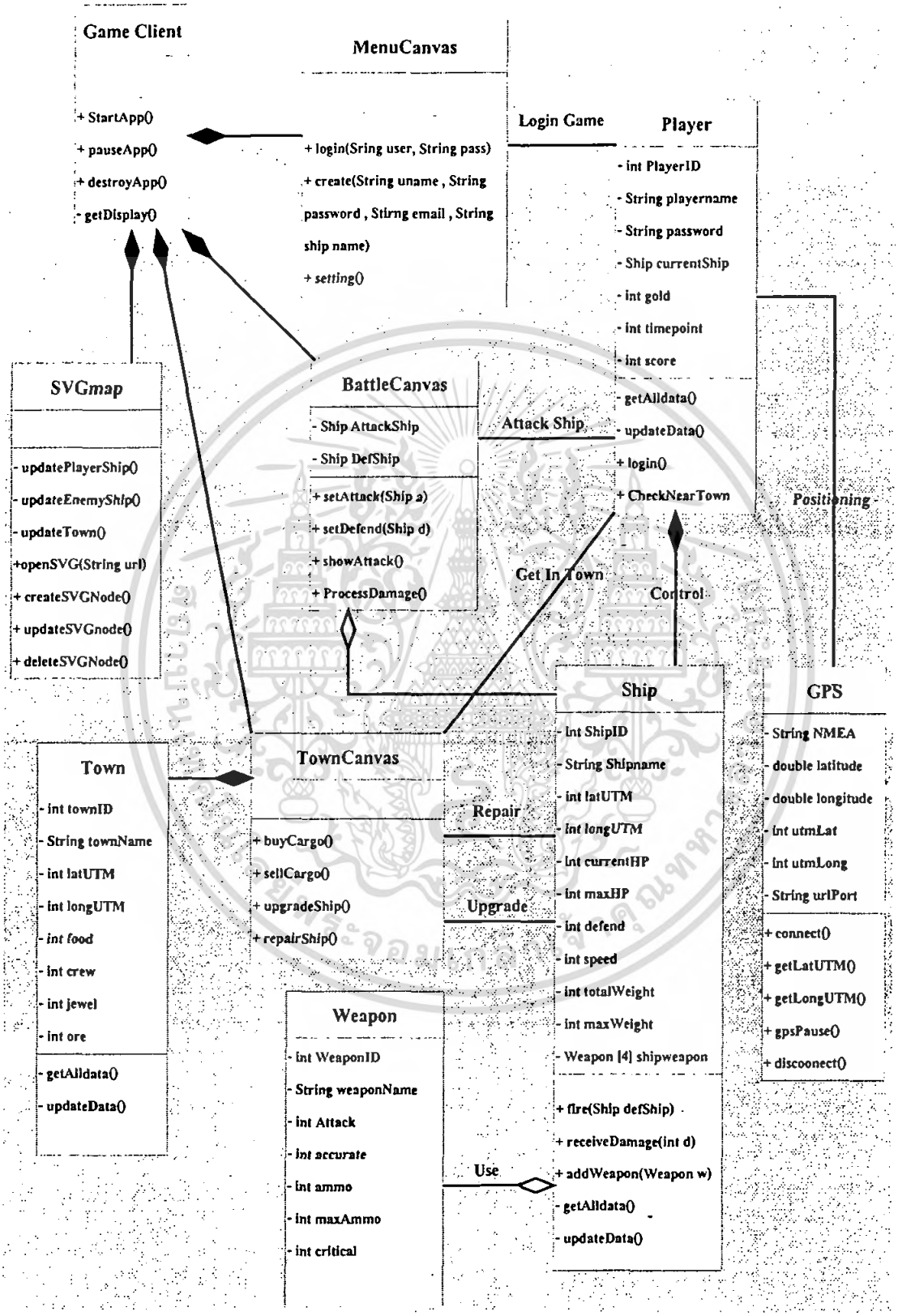
Main Flow: 1. จะทำการเชื่อมต่อไปยัง Servlet

2. อ่านข้อมูลและเก็บค่า

3. นำข้อมูลมาตรวจสอบกับเงื่อนไขการทำงานต่างๆของกิจกรรมในเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 Design Class Diagram

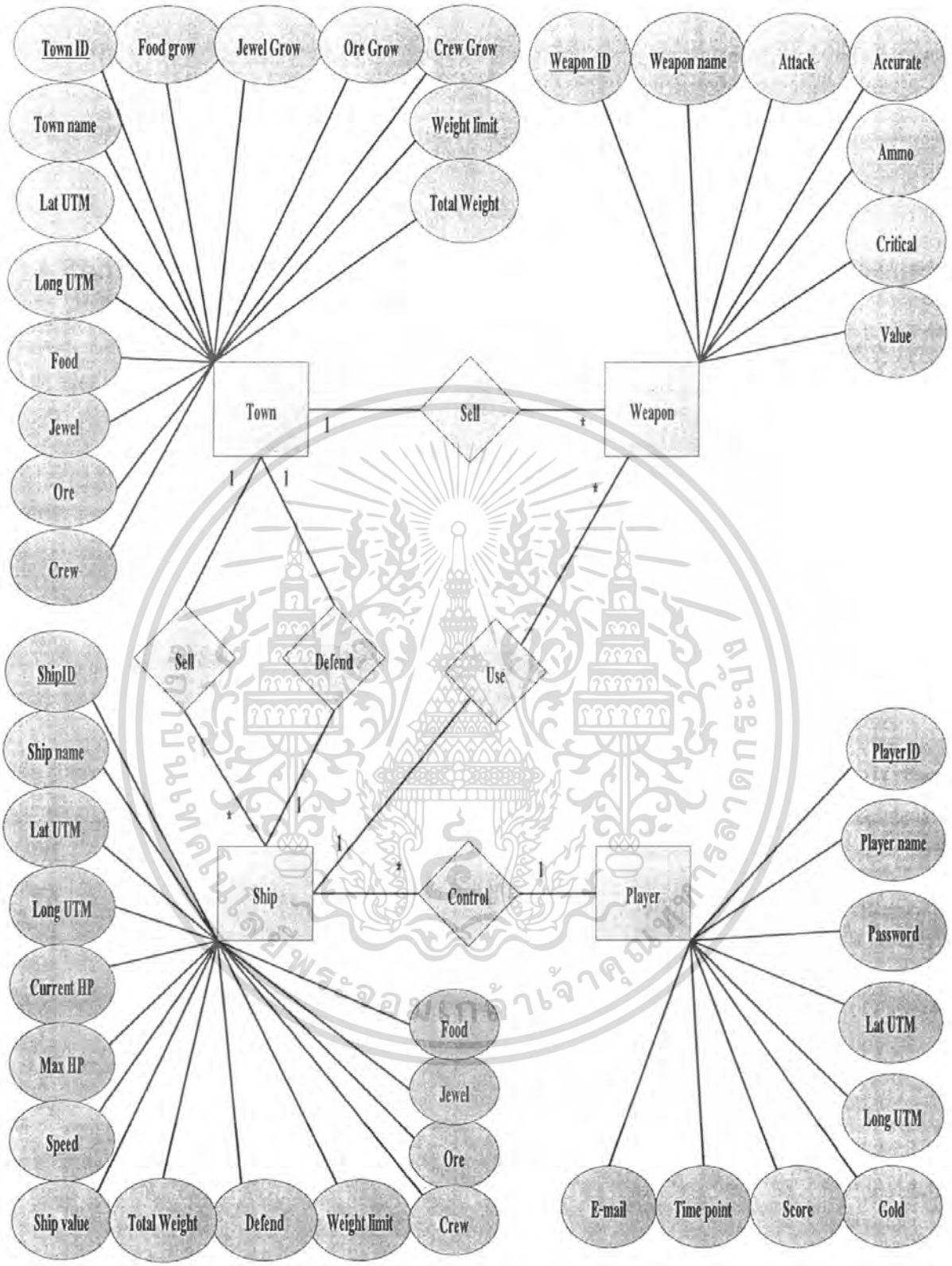


รูปที่ 4.4 แสดง Class Diagram ของ Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบฐานข้อมูลดัง ER Diagram ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดง ER Diagram ของระบบ

4.3.3 รายละเอียดของตารางข้อมูลต่างๆ

ตาราง ship ใช้ในการจัดเก็บข้อมูลรายละเอียดของผู้เล่นทั้งหมดภายในเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 4.5 แสดงส่วนประกอบของตาราง ship

Name	Type	Description
playerid	int	id ของผู้เล่น
playername	String(20)	ชื่อของผู้เล่นเกมที่ใช้เข้าเกม
password	String(20)	รหัสผ่านของผู้เล่นที่ใช้เข้าเกม
utmlat	Double	ค่าตำแหน่ง Latitude ใน UTM ของผู้เล่น
utmlong	Double	ค่าตำแหน่ง Longitude ใน UTM ของผู้เล่น
gold	Int	เงินของผู้เล่นที่มีอยู่ภายในเกม
score	Int	ค่า score ของผู้เล่นภายในเกม
timepoint	Int	ค่าที่ใช้ในการพัฒนาเมือง โดยจะเพิ่มขึ้นตามเวลา
e-mail	String(50)	อีเมล์ของผู้เล่น
currentshipid	Int	เรือปัจจุบันที่ผู้เล่นบังคับอยู่

ตาราง playership ใช้เก็บความสัมพันธ์ระหว่างผู้เล่นกับเรือของผู้เล่น

ตาราง 4.6 แสดงส่วนประกอบของตาราง playership

Name	Type	Description
playerid	Int	id ของผู้เล่นที่อยู่ในตาราง player
Shipid	Int	id ของเรือที่อยู่ในตาราง ship

ตาราง ship ใช้ในการเก็บค่าต่างๆของเรือทุกลำที่อยู่ในตัวเกม

ตาราง 4.7 แสดงส่วนประกอบของตาราง ship

Column	Type	ความหมาย
Shipid	Int	id ของเรือที่อยู่ภายในเกม
shipname	String(20)	ชื่อของเรือที่อยู่ภายในเกม
utmlat	Double	ตำแหน่ง Latitude ของ UTM ของเรือ
utmlong	Double	ตำแหน่ง Longitude ของ UTM ของเรือ
currenthp	Int	พลังชีวิตของเรือที่เหลืออยู่ของเรือ
maxhp	Int	พลังชีวิตสูงสุดของเรือ
Speed	Int	ความว่องไวของเรือในการเล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

defend	Int	ความแข็งแรงของเรือ
maxweight	Int	น้ำหนักสูงสุดที่เรือลำนั้นจะสามารถบรรทุกได้
Value	Int	ราคาขายของเรือ
totalweight	Int	น้ำหนักรวมของสินค้าและลูกเรือที่บรรทุกอยู่
Food	Int	จำนวนอาหารที่บรรทุกอยู่บนเรือ
Jewel	Int	จำนวนเพชรที่บรรทุกอยู่บนเรือ
Ore	Int	จำนวนแร่ที่บรรทุกอยู่บนเรือ
Crew	Int	จำนวนลูกเรือที่อยู่บนเรือ

ตาราง shipweapon ใช้เก็บความสัมพันธ์ระหว่างเรือกับอาวุธของเรือ

ตาราง 4.8 แสดงส่วนประกอบของตาราง shipweapon

Column	Type	ความหมาย
Shipid	Int	Id ของเรือที่อยู่ในตาราง ship
weaponid	Int	Id ของอาวุธที่อยู่ในตาราง weapon

ตาราง town ใช้ในการเก็บข้อมูลต่างๆของเมืองที่ใช้ในการเล่นเกม

ตาราง 4.9 แสดงส่วนประกอบของตาราง ship

Column	Type	ความหมาย
townid	Int	Id ของเมืองในเกม
townname	String(20)	ชื่อเมืองภายในเกม
utmlat	Double	ตำแหน่ง Latitude ของ UTM ของเมือง
utmlong	Double	ตำแหน่ง Longitude ของ UTM ของเมือง
foodgrow	Int	อัตราการเพิ่มของอาหารภายในเมือง
jewelgrow	Int	อัตราการเพิ่มของเพชรภายในเมือง
oregrow	Int	อัตราการเพิ่มของลูกเรือภายในเมือง
crewgrow	Int	อัตราการเพิ่มของลูกเรือภายในเมือง
weightlimit	Int	น้ำหนักมากสุดที่เมืองสามารถเก็บได้
defenderid	Int	Id ของผู้เล่นที่เป็นเจ้าของ
totalweight	Int	น้ำหนักรวมทั้งหมดของทรัพยากรและลูกเรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		ภายในเมือง
food	Int	อาหารภายในเมือง
jewel	Int	เพชรภายในเมือง
ore	Int	แร่ภายในเมือง
crew	Int	ลูกเรือภายในเมือง

ตาราง township ใช้เก็บความสัมพันธ์ระหว่างเรือกับเมือง โดยจะทำการเก็บไอดีของเรือกับไอดีของเมือง เพื่อให้ทราบว่าเมืองนั้นๆมีเรือลำไหนจอดเทียบท่าอยู่

ตาราง 4.10 แสดงส่วนประกอบของตาราง township

Column	Type	ความหมาย
townid	Int	Id ของเมืองของตาราง town
shipid	Int	Id ของเรือของตาราง ship

ตาราง townweapon ใช้เก็บความสัมพันธ์ระหว่างเมืองกับอาวุธ โดยจะทำการเก็บไอดีของเมืองกับไอดีของอาวุธ เพื่อให้ทราบว่าเมืองนั้นๆมีอาวุธอันไหนอยู่บ้าง

ตาราง 4.11 แสดงส่วนประกอบของตาราง townweapon

Column	Type	ความหมาย
Townid	Int	Id ของเมืองของตาราง town
weaponid	Int	Id ของอาวุธของตาราง weapon

ตาราง weapon ใช้เก็บข้อมูลต่างๆที่สำคัญและเกี่ยวข้องกับอาวุธภายในเกม

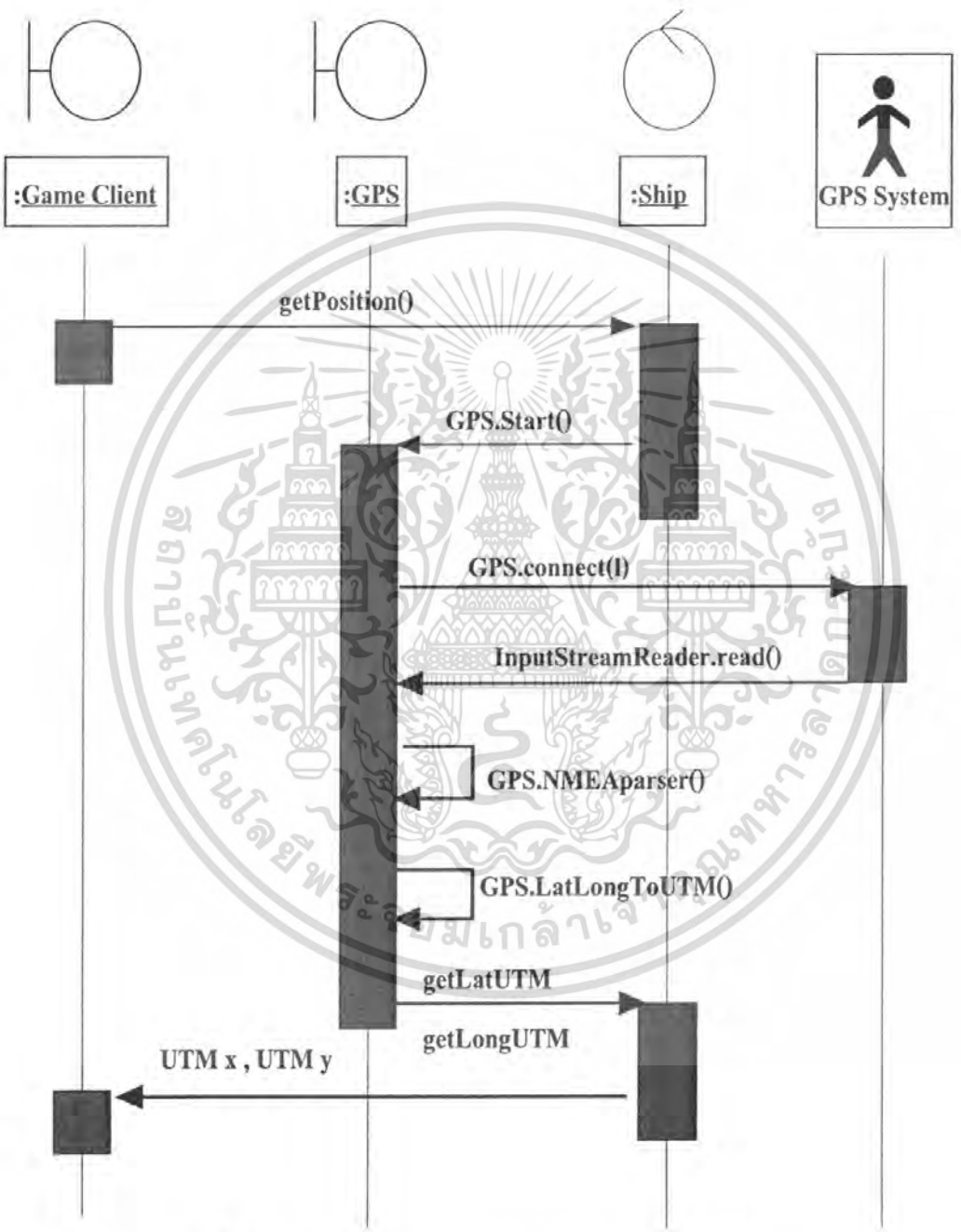
ตาราง 4.12 แสดงส่วนประกอบของตาราง weapon

Column	Type	ความหมาย
weaponid	Int	Id ของอาวุธภายในเกม
weaponname	String(20)	ชื่ออาวุธภายในเกม
Attack	Int	อัตราการโจมตีของอาวุธนั้น
accurate	Int	ความแม่นยำของอาวุธ
Ammo	Int	อัตราการป้องกันของอาวุธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

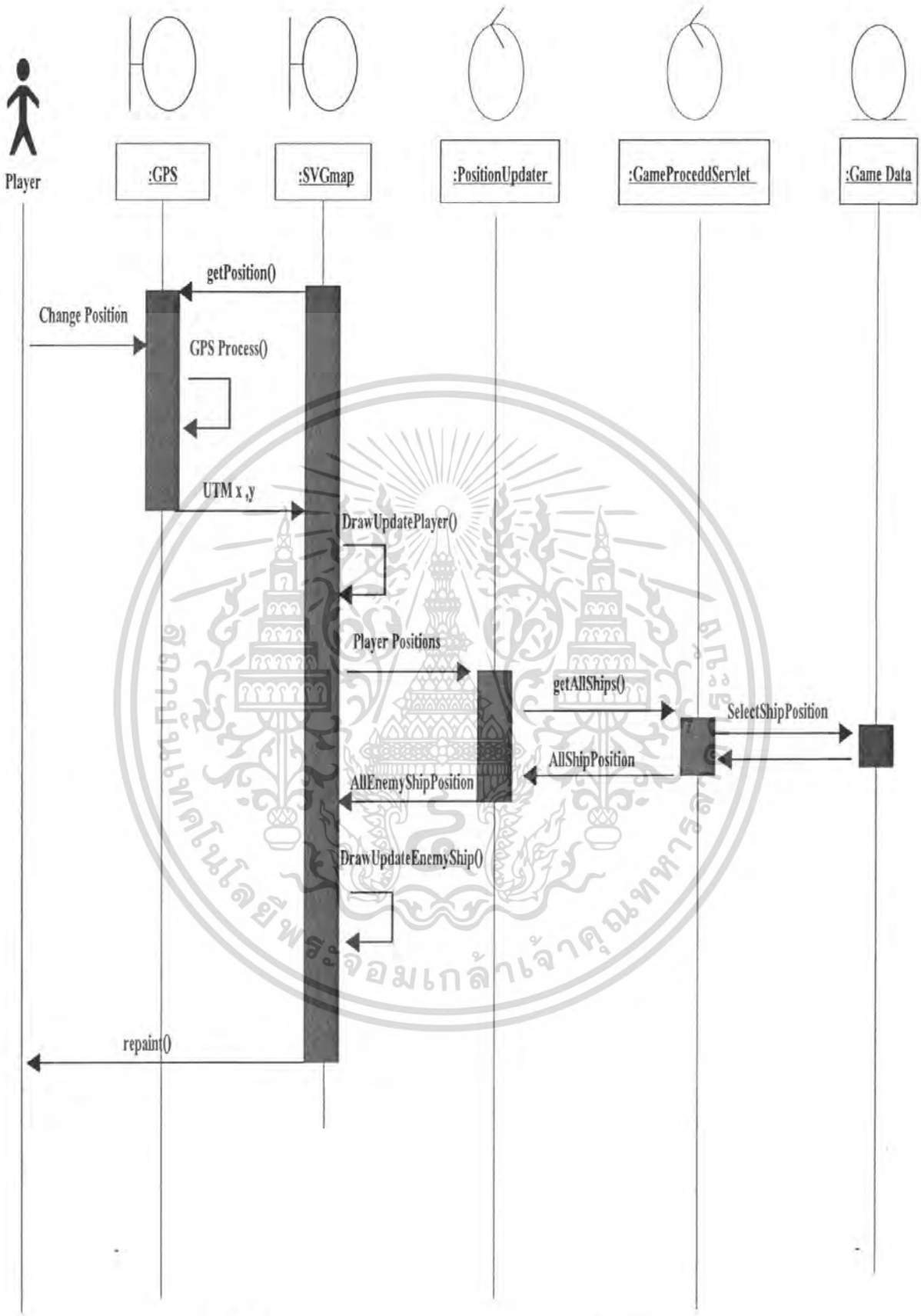
Critical	Int	โอกาสการเกิดการโจมตีที่มากกว่าปกติ
Value	Int	ราคาขายของอาวุธนั้น

4.3.4 การออกแบบลำดับการทำงานของระบบ



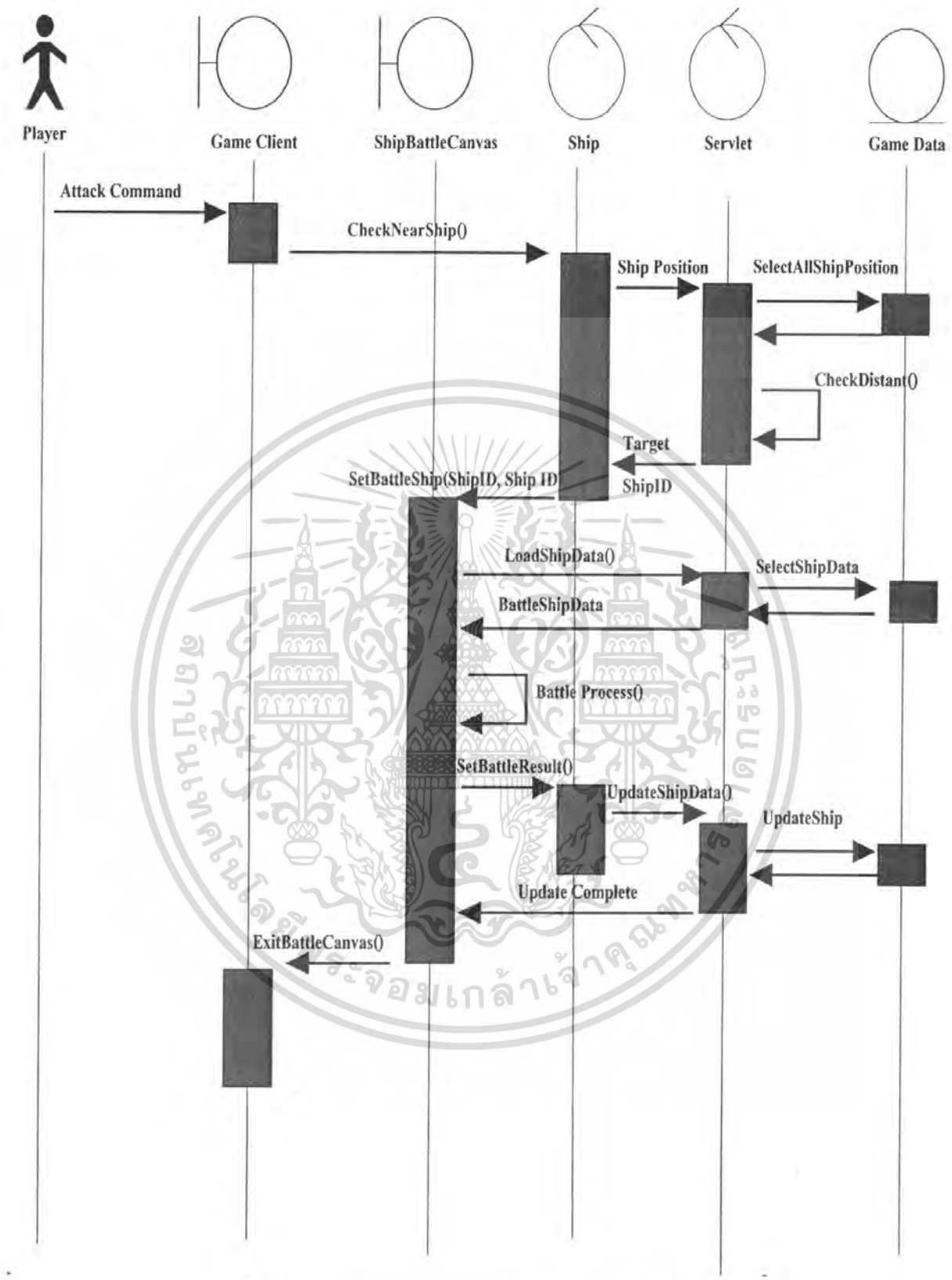
รูปที่ 4.6 แสดง Sequenial Diagram ของการระบุตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดง Sequenial Diagram ของการเดินเรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



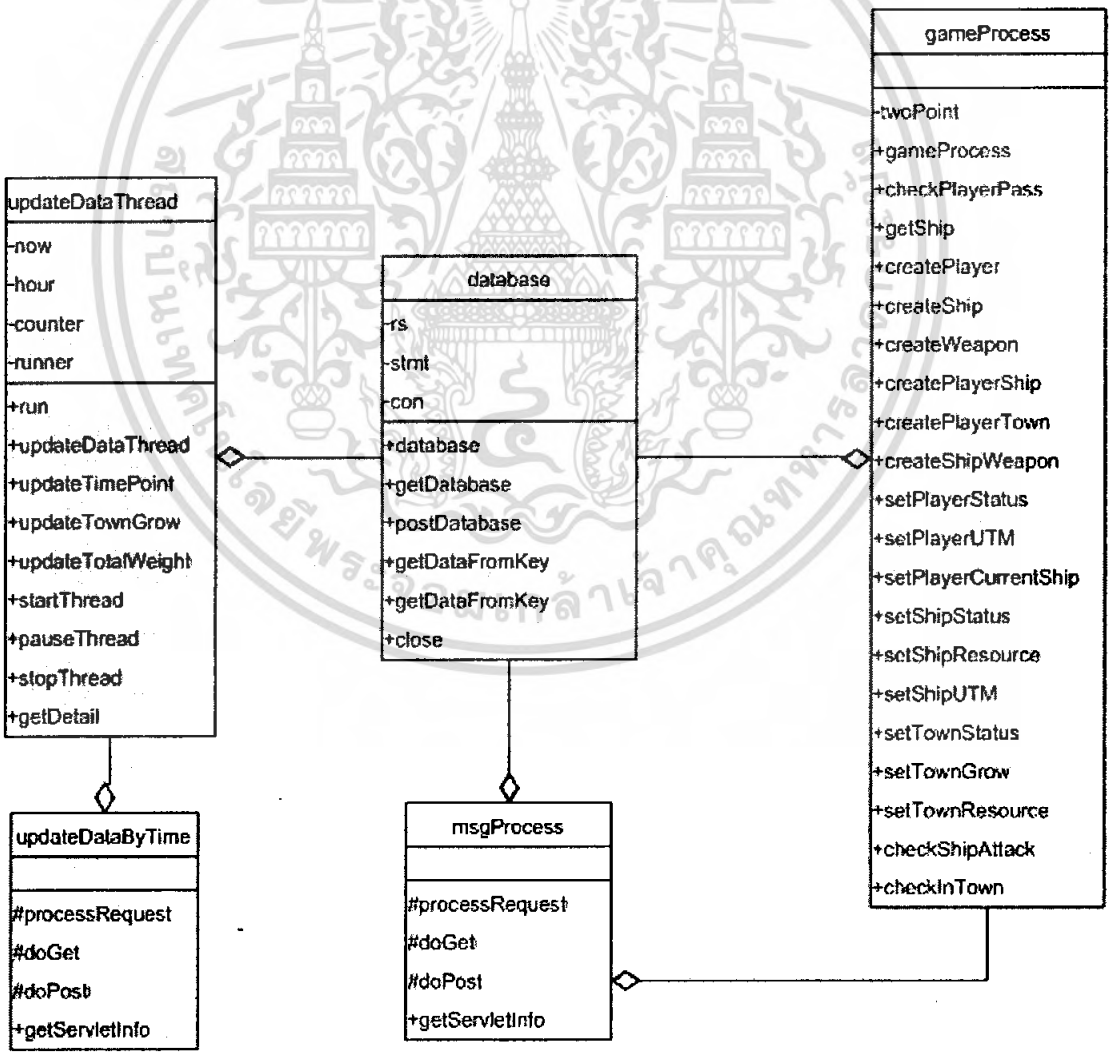
รูปที่ 4.8 แสดง Sequenial Diagram ของการโจมตีเรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การออกแบบการทำงานส่วนเซิร์ฟเวอร์

ในส่วนของเซิร์ฟเวอร์นั้นได้พัฒนาเป็น web application server โดยใช้ Apace Tomcat ในการจัดการส่วนของเซิร์ฟเวอร์และใช้เทคโนโลยี Java Servlet ในการจัดการเรื่องการประมวลผล นอกจากตัว Servlet จะจัดการส่วนประมวลผลแล้วยังทำหน้าที่เป็นตัวส่งข้อมูลที่จำเป็นในการเล่นเกมให้แก่ไคลเอนต์ด้วย โดยตัวเซิร์ฟเวอร์จะจัดการ request จากไคลเอนต์ผ่านทาง HTTP Connection และส่งข้อมูลตามที่ไคลเอนต์ร้องขอมากลับไป โดยใช้ Java Servlet เป็นตัวดำเนินการรอรับและตอบสนองกับไคลเอนต์ซึ่งในการติดต่อกับฐานข้อมูลนั้นใช้ JDBC driver เป็นตัวกลางในการเชื่อมต่อ และในส่วนของฐานข้อมูลนั้นใช้ postgreSQL เป็น DBMS ไว้การจัดเก็บข้อมูลของผู้เล่นและข้อมูลที่จำเป็นต่างๆในการเล่นเกม

4.4.1 Class diagram ของส่วนเซิร์ฟเวอร์

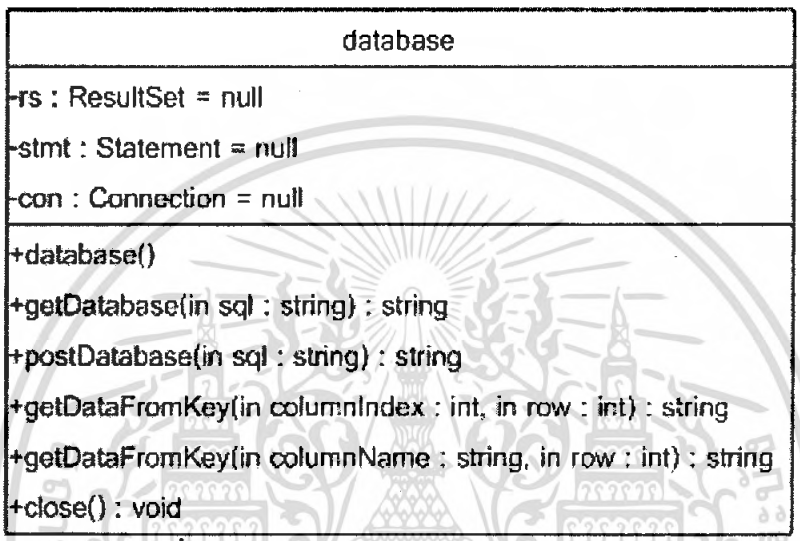


รูปที่ 4.9 แสดง class diagram ของ Servlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.1.1 คลาส database

เป็นคลาสที่ใช้จัดการในส่วนของการรับและส่งข้อมูลระหว่างตัว Servlet กับฐานข้อมูล ในการดึงข้อมูลที่ต้องการมาจากฐานข้อมูลนั้นจะใช้ฟังก์ชัน getDatabase ในการนำข้อมูลที่ query ได้มาเก็บไว้ในออฟเจ็ค ResultSet และใช้ฟังก์ชัน getDataFromKey ในการดึงข้อมูลที่ต้องการออกมาเป็นตัวๆไป ส่วนในการใส่ข้อมูลลงฐานข้อมูลนั้นจะเป็นหน้าที่ของฟังก์ชัน postDatabase ในการจัดการ



รูปที่ 4.10 แสดงรายละเอียดของคลาส database

■ ฟังก์ชัน get Database

เป็นฟังก์ชันสำคัญที่ใช้ในการดึงข้อมูลที่ query ได้ทั้งหมดไปใส่ในออฟเจ็ค ResultSet เพื่อให้ผู้ใช้เรียกดูข้อมูลที่ต้องการผ่านทางฟังก์ชัน getDataFromKey ต่อไป โดยโครงสร้างของตัวฟังก์ชันมีดังนี้

```
String getDatabase(String sql)
Parameter = sql //เป็นคำสั่ง sql ในการดึงข้อมูลที่ต้องการจากฐานข้อมูล
Return    = status  //สถานะว่าดึงข้อมูลได้หรือไม่ได้
```

■ ฟังก์ชัน getDataFromKey

เป็นฟังก์ชันที่ใช้ในการดึงข้อมูลที่ต้องการจากออฟเจ็ค ResultSet ในตำแหน่งที่ row และแถวที่ column นั้นๆ โดยในการเรียกฟังก์ชันนี้หนึ่งครั้งจะได้ข้อมูลมาหนึ่งตัวเท่านั้นและหากไม่มีข้อมูลในตำแหน่งที่ต้องการจะรีเทิร์นเป็น '$'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

String getDataFromKey(int column, int row)		
Parameter	= column	//เป็นตำแหน่ง Column ที่ต้องการดึงข้อมูลมา
	= row	//เป็นตำแหน่งของ row ที่ต้องการดึงข้อมูลมา
Return	= value	//ข้อมูลตำแหน่งที่ row และแถวที่ column

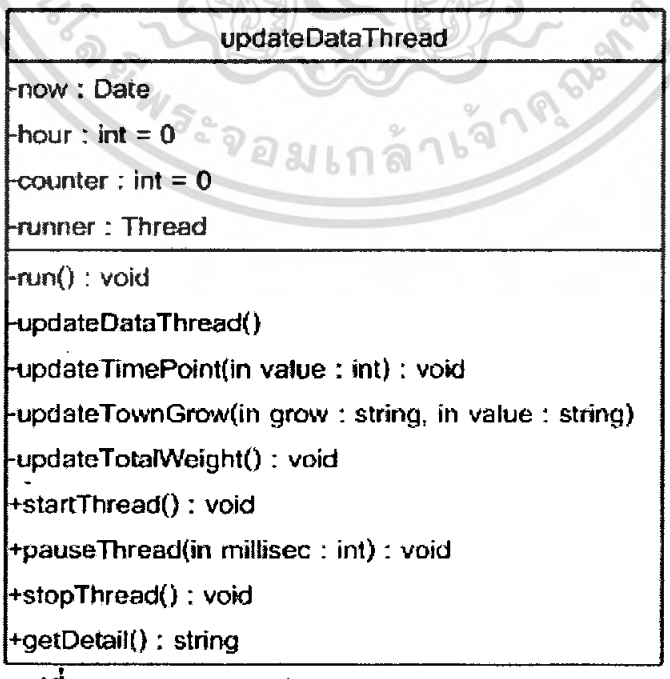
- ฟังก์ชัน postDatabase

เป็นฟังก์ชันที่ใช้ในการใส่หรือลบข้อมูลจากฐานข้อมูล ซึ่งใช้งานร่วมกับคำสั่ง create ,updateและ delete โดยโครงสร้างของฟังก์ชันมีดังนี้

String postDatabase(String sql)		
Parameter	= sql	//เป็นคำสั่ง sql ในการใส่หรือลบข้อมูลจากฐานข้อมูล
Return	= status	//เป็นสถานะว่าใส่หรือลบข้อมูลได้หรือไม่

4.4.1.2 คลาส updateDataThread

เป็นคลาสที่ใช้ในการเพิ่มค่าของข้อมูลต่างๆที่ใช้ในตัวเกมตามเวลาที่กำหนดไว้ ซึ่งค่าที่ต้องอัพเดตได้แก่ค่า timepoint ของผู้เล่นทุกคนและทรัพยากรต่างๆภายในเมือง ซึ่งคลาสนี้ได้สร้างเป็นเทรดอีกเทรดหนึ่งทำงานไปพร้อมกับ Servlet ส่วนอื่นๆ ซึ่งฟังก์ชันที่สำคัญๆมีดังนี้



รูปที่ 4.11 แสดงรายละเอียดของคลาส updateDataThread

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ ฟังก์ชัน updateTimePoint

เป็นฟังก์ชันที่ใช้ในการอัพเดตค่า Time Point ของผู้เล่นทุกคนภายในเกม โดยตัวโครงสร้างของฟังก์ชันนี้มีดังนี้

```
void updateTimePoint(int value)
Parameter = value          //เป็นจำนวน Time Point ที่เพิ่ม
Return    = ไม่มี
```

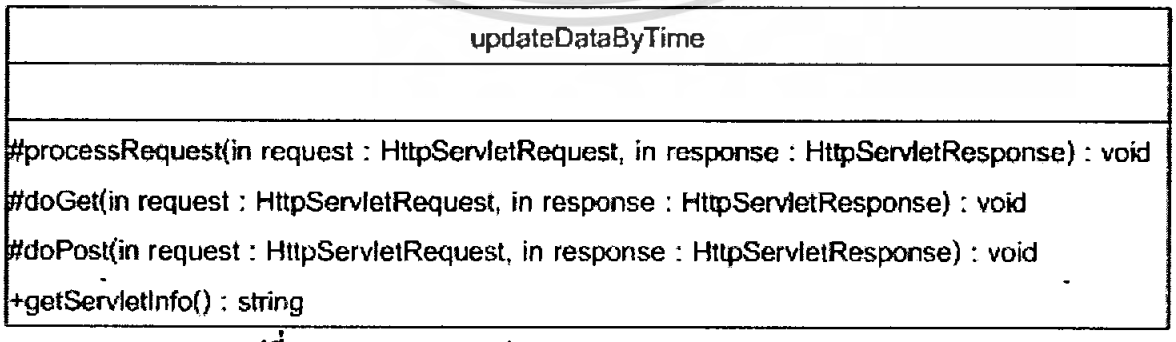
■ ฟังก์ชัน updateTotalWeight

เป็นฟังก์ชันที่ใช้ในการอัพเดตค่า Totalweight ของเมืองทั้งหมดในตาราง Town ของฐานข้อมูล โดยจะนำน้ำหนักของทรัพยากรต่างๆมาบวกกันและทำการใส่ไปใน column นี้

```
void updateTotalWeight(String grow, String value)
Parameter = grow     //เป็นชื่อของ cloumn อัตราการเติบโตของทรัพยากร
          = value    //เป็นชื่อทรัพยากรที่ต้องการเพิ่ม
Return    = ไม่มี
```

4.4.1.3 คลาส updateDataByTime

เป็นคลาสของ Servlet ที่ใช้ในการรับการร้องขอของผู้ผ่านทาง HTTP Connection ซึ่งเปรียบเสมือนเป็น Interface สำหรับจัดการคลาส updateDataThread โดยในการจัดการนั้นผู้ใช้จะส่งพารามิเตอร์ที่ชื่อว่า Status มาด้วย ซึ่งค่าของตัวแปรจะเป็นตัวควบคุมการทำงาน โดยค่า Status มีค่าดังนี้คือ “Start”, “Destroy” และ “GetStatus” ซึ่งหมายความของแต่ละข้อความมีดังนี้



รูปที่ 4.12 แสดงรายละเอียดของคลาส updateDataByTime

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การทำงานเมื่อ Status เป็น Start

เป็นข้อความที่ผู้ใช้สั่งให้ thread ของออฟเจ็ค updateDataThread ทำงานเพื่อทำการอัพเดตค่าข้อมูลของตัวเกมค่าต่างๆตามเวลาที่ได้กำหนดไว้

- การทำงานเมื่อ Status เป็น GetStatus

เป็นข้อความที่ผู้ใช้ร้องขอข้อมูลรายละเอียดต่างๆที่จะเป็นของออฟเจ็ค updateDataThread อย่างเช่นเวลาของเซิร์ฟเวอร์ในปัจจุบัน, จำนวนครั้งที่อัพเดตค่าต่างๆของข้อมูล เป็นต้น

- การทำงานเมื่อ Status เป็น Destroy

เป็นการเรียกฟังก์ชัน stopThread ของออฟเจ็ค updateDataTread ดดยจะเป็นการทำลาย thread ที่รันอยู่ใน updateDataThread

4.4.1.4 คลาส gameProcess

เป็นคลาสที่ใช้ในการจัดการของฐานข้อมูลที่ใช้ภายในเกม โดยมีหน้าที่หลักๆคือ นำข้าข้อมูลที่ผู้เล่นต้องการจากฐานข้อมูลมาแปลงให้อยู่ในฟอร์มที่ไคลเอนต์เข้าใจแล้วส่งออกข้อมูลนั้นให้ไคลเอนต์ต่อไปหรือหน้าที่อีกส่วนนึงคือรับข้อมูลจากไคลเอนต์มาอัพเดตข้อมูลในฐานข้อมูล ซึ่งฟังก์ชันที่สำคัญของคลาสนี้มีดังนี้

gameProcess
-twoPoint(in x1 : string, in x1, in y1 : int, in x2 : int, in y2 : int) : double
+gameProcess()
+checkPlayerPass(in user : string, in pass : string) : Boolean
+getShip(in shipID : String[0..*], in shipTown : String[0..*]) : string
+createPlayer(in id : int, in name : string, in pass : string, in email : string, in currentShip : string) : string
+createShip(in id : int, in name : string) : string
+createWeapon(in id : int) : string
+createPlayerShip(in playerID : int, in shipID : int) : string
+createPlayerTown(in playerID : int, in townID : int) : string
+createShipWeapon(in shipID : int, in weaponID : int) : string
+setPlayerStatus(in id : string, in gold : string, in score : string, in timepoint : string) : string
+setPlayerUTM(in id : string, in lat : string, in long : string) : string
+setPlayerCurrentShip()
+setShipStatus(in id : string, in currenthp : string, in speed : string, in defend : string, in maxweight : string, in value : string) : string
+setShipResource(in totalweight : string, in food : string, in jewel : string, in ore : string, in crew : string) : String[0..*]
+setShipUTM(in id : string, in lat : string, in long : string) : string
+setTownStatus(in id : string, in defenderid : string, in weightlimit : string) : string
+setTownGrow(in id : string, in food : string, in jewel : string, in ore : string, in crew : string) : string
+setTownResource(in id : string, in food : string, in jewel : string, in ore : string, in crew : string, in totalweight : string) : string
+checkShipAttack(in shipid : string) : string
+checkInTown(in playerID : string) : string

รูปที่ 4.13 แสดงรายละเอียดของคลาส playerProcess

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟังก์ชัน checkPlayerPass

เป็นฟังก์ชันที่ใช้เช็ค username และ password ว่าถูกต้องหรือไม่ ถ้ามีจะรีเทิร์นเป็น True ถ้าไม่ถูกจะรีเทิร์น false ซึ่งตัวโครงสร้างของฟังก์ชันนี้มีดังนี้

```
Boolean checkPlayer( String user, String pass)
Parameter  = user        //ชื่อผู้เล่น
           = password    //พาสเวิร์ดของผู้เล่น
Return     = Status      //ถ้ามี user และ password ที่ใส่ไปจะเป็น True ถ้าไม่มีจะเป็น false
```

- ฟังก์ชัน getShip

ฟังก์ชันนี้มีหน้าที่ในการส่งข้อมูลของเรือที่มีตำแหน่งอยู่บนแผนที่ โดยจะมีการส่ง shipid, utmlat และ utmlong ไปให้ไคลเอนต์โดยข้อมูลที่ส่งไปให้ไคลเอนต์นั้นจะอยู่ในรูปข้อมูลที่ต่อกัน โดยใช้เครื่องหมาย “,” คั่นระหว่างข้อมูล ซึ่งตัวฟังก์ชันมีโครงสร้างดังนี้

```
String getShip(String[] shipID,String[] shipTown)
Parameter  = shipID      //เป็นไอดีของเรือของผู้เล่น
           = shipTown    //เป็นไอดีของเรือที่ขายอยู่ในเมือง
Return     = String      //สตริงของข้อมูลที่ต่อกัน
```

- ฟังก์ชัน insertPlayer

เป็นฟังก์ชันที่ใช้ในการ insert ข้อมูลของผู้เล่นลงในฐานข้อมูล โดยมีรายละเอียดของฟังก์ชันดังต่อไป

```
String insertPlayer(int id, String Name, String Pass, String Email, int currentShip)
Parameter  = id          //ไอดีของผู้เล่น
           = Name        //ชื่อของผู้เล่น
           = Pass        //พาสเวิร์ดของผู้เล่น
           = Email       //อีเมล์ของผู้เล่น
           = currentShip //เรือปัจจุบันของผู้เล่น
Return     = String      //ผลของการ insert
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟังก์ชัน insertShip

เป็นฟังก์ชันที่ใช้ในการ insert ข้อมูลของเรือลงในฐานข้อมูล โดยมีรายละเอียดของฟังก์ชันดังต่อไปนี้

```
String insertShip(int shipID,String shipName)
Parameter   = shipID      //ไอดีของเรือ
            = shipName    //ชื่อของเรือ
Return      = String      //ผลของการ insert
```

- ฟังก์ชัน insertWeapon

เป็นฟังก์ชันที่ใช้ในการ insert ข้อมูลของอาวุธลงในฐานข้อมูล โดยมีรายละเอียดของฟังก์ชันดังต่อไปนี้

```
String insertWeapon(int weaponid)
Parameter   = weaponid    // ไอดีของอาวุธ
Return      = String      //ผลของการ insert
```

- ฟังก์ชัน insertPlayerShip

เป็นฟังก์ชันที่ใช้ในการ insert ข้อมูลความสัมพันธ์ระหว่างผู้เล่นกับเรือ โดยมีรายละเอียดของฟังก์ชันดังต่อไปนี้

```
String insertPlayerShip(int playerID,int shipID)
Parameter   = playerID    //ไอดีของผู้เล่น
            = shipID      //ไอดีของเรือ
Return      = String      //ผลของการ insert
```

- ฟังก์ชัน insertPlayerTown

เป็นฟังก์ชันที่ใช้ในการ insert ข้อมูลความสัมพันธ์ระหว่างผู้เล่นกับเมือง โดยมีรายละเอียดของฟังก์ชันดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
String insertPlayerTown(int playerID,int townID)
Parameter   = playerID      //ไอดีของผู้เล่น
            = townID        //ไอดีของเมือง
Return      = String        //ผลของการ insert
```

- ฟังก์ชัน insertPlayerWeapon

เป็นฟังก์ชันที่ใช้ในการ insert ข้อมูลความสัมพันธ์ระหว่างผู้เล่นกับอาวุธ โดยมีรายละเอียดของฟังก์ชันดังต่อไปนี้

```
String insertPlayerWeapon(int playerID,int weaponID)
Parameter   = playerID      //ไอดีของผู้เล่น
            = weaponID      //ไอดีของอาวุธ
Return      = String        //ผลของการ insert
```

- ฟังก์ชัน insertShipWeapon

เป็นฟังก์ชันที่ใช้ในการ insert ข้อมูลความสัมพันธ์ระหว่างเรือกับอาวุธ โดยมีรายละเอียดของฟังก์ชันดังต่อไปนี้

```
String insertShipWeapon(int shipID,int weaponID)
Parameter   = shipID        //ไอดีของผู้เล่น
            = weaponID      //ไอดีของอาวุธ
Return      = String        //ผลของการ insert
```

- ฟังก์ชัน insertTownShip

เป็นฟังก์ชันที่ใช้ในการ insert ข้อมูลความสัมพันธ์ระหว่างเรือกับเมือง โดยมีรายละเอียดของฟังก์ชันดังต่อไปนี้

```
String insertTownShip(int townID,int shipID)
Parameter   = townID        //ไอดีของเมือง
            = shipID        //ไอดีของเรือ
Return      = String        //ผลของการ insert
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟังก์ชัน updatePlayerUTM

เป็นฟังก์ชันที่ใช้ในเปลี่ยนตำแหน่งของผู้เล่นที่รับมาจากไคลเอนต์ไปยังฐานข้อมูล โดยมีรายละเอียดของฟังก์ชันดังต่อไปนี้

```
String updatePlayerUTM(String playerid,String utmlat,String utmlong)
Parameter  = playerid     //ไอดีของผู้เล่น
           = utmlat       //ค่าตำแหน่ง utmlat ใหม่ของผู้เล่น
           = utmlong      //ค่าตำแหน่ง utmlong ใหม่ของผู้เล่น
Parameter  = String       //ผลของการเปลี่ยนแปลงค่า
```

- ฟังก์ชัน updateShipUTM

เป็นฟังก์ชันที่ใช้ในเปลี่ยนตำแหน่งของเรือที่รับมาจากไคลเอนต์ไปยังฐานข้อมูล โดยมีรายละเอียดของฟังก์ชันดังต่อไปนี้

```
String updateShipUTM(String shipid,String utmlat,String utmlong)
Parameter  = shipid       //ไอดีขอเรือ
           = utmlat       //ค่าตำแหน่ง utmlat ใหม่ของเรือ
           = utmlong      //ค่าตำแหน่ง utmlong ใหม่ของเรือ
Parameter  = String       //ผลของการเปลี่ยนแปลงค่า
```

- ฟังก์ชัน shipAttack

เป็นฟังก์ชันที่ใช้ในการหาเรือของศัตรูเพื่อทำการต่อสู้ โดยตำแหน่งเรือของศัตรูนั้นจะต้องห่างจากเรือของผู้เล่นไม่เกิน 50 เมตร ซึ่งรายละเอียดของฟังก์ชันมีดังต่อไปนี้

```
String shipAttack(String shipid)
Parameter  = shipid       //ไอดีเรือของผู้เล่น
Return     = String       //ถ้าสามารถโจมตีได้จะรีเทิร์นอีและตำแหน่งของเรือ
                          //ถ้าไม่จะรีเทิร์น",,,"
```

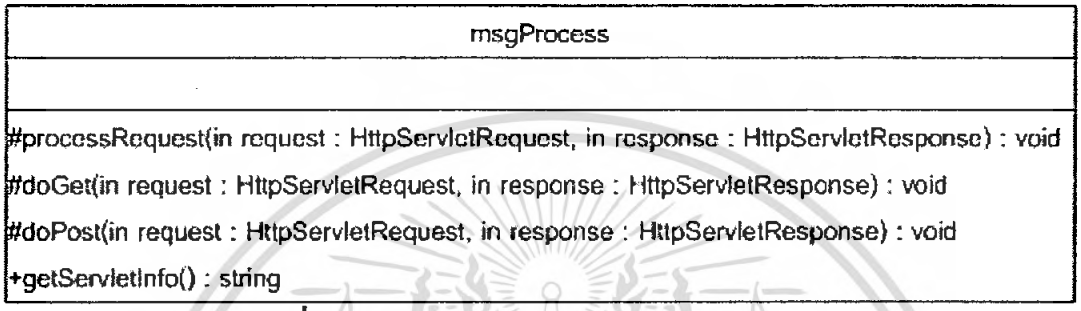
- ฟังก์ชัน inTown

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นฟังก์ชันที่ใช้ในการเช็คการเข้าเมืองของเรือโดยเรือลำนั้นต้องอยู่ห่างจากเมืองไม่เกิน 50 เมตรจึงจะถือว่าเข้าเมือง ซึ่งรายละเอียดของฟังก์ชันมีดังต่อไปนี้

4.4.1.5 คลาส msgProcess

เป็นคลาสของ Servlet ที่ทำหน้าที่ในการจัดการกับ message ที่มาจากไคลเอนต์โดยส่วนใหญ่แล้วจะเป็น message เกี่ยวกับการส่งข้อมูลที่จำเป็นในการเล่นเกมให้ไคลเอนต์และ message เกี่ยวกับการอัพเดตค่าต่างๆภายในเกม ซึ่ง message ที่สำคัญมีดังต่อไปนี้



รูปที่ 4.14 แสดงรายละเอียดของคลาส msgProcess

- เมสเสจ Log In

เป็นเมสเจสที่ทำหน้าที่ตรวจสอบความถูกต้องของผู้เล่นที่ทำการ log in โดยจะตรวจว่าชื่อผู้เล่น (username)และรหัสผ่าน (password) ถูกต้องหรือไม่ ถ้าถูกต้องจะส่งข้อมูลของผู้เล่นกลับไปให้ไคลเอนต์แต่ถ้าไม่ถูกต้องจะส่งข้อความ "False" กลับไป

- พารามิเตอร์ที่ส่งมา
  - msg = "Login"
  - PlayerName = username ของผู้เล่น
  - Password = password ของผู้เล่น
- Return
  - ข้อมูลของผู้เล่น

- เมสเสจ Sign Up

เป็นเมสเจสที่จัดการเกี่ยวกับการลงทะเบียนเล่นเกมของผู้เล่นใหม่ โดยเริ่มแรกนั้นระบบจะทำการตรวจสอบก่อนว่าชื่อของผู้เล่นซ้ำหรือไม่ ถ้าไม่ซ้ำจะทำการเพิ่มข้อมูลของผู้เล่นเข้าไปในฐานข้อมูล แต่ถ้าซ้ำจะส่งข้อความบอกให้ผู้เล่นกรอกข้อมูลใหม่อีกที

- พารามิเตอร์ที่ส่งมา
  - msg = "SignUp"
  - name = ชื่อที่ใช้ลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- password = รหัสผ่าน
- email = อีเมล์ของผู้เล่น
- shipname = ชื่อเรือของผู้เล่น

- Return
  - ผลของการลงทะเบียน

■ เมสเสจ Ship

เป็นเมสเสจที่จะส่งตำแหน่งของเรือของผู้เล่นมาเพื่อทำการอัพเดตลงฐานข้อมูล พร้อมทั้งให้เซิร์ฟเวอร์ตำแหน่งเรือของศัตรูทั้งหมดที่อยู่บนแผนที่กลับไปให้ไคลเอนต์

-พารามิเตอร์ที่ส่งมา

- msg = "Ship"
- playerid = ไอดีของผู้เล่น
- utmlat = ค่า utmlat ของผู้เล่น
- utmlong = ค่า utmlong ของผู้เล่น

- Return
  - ไอดีและตำแหน่งของเรือศัตรูทั้งหมด

■ เมสเสจ shipAttack

เป็นเมสเสจสำหรับเช็คว่ามีเรือของศัตรูที่อยู่ใกล้พอที่จะโจมตีได้มั้ย โดยกำหนดระยะห่างไม่เกิน 50 หน่วย UTM ถ้ามีเรืออยู่ในระยะนี้ตัวเซิร์ฟเวอร์จะส่งไอดีและตำแหน่งของเรือศัตรูไปให้ไคลเอนต์แต่ถ้าไม่มีเรืออยู่ในระยะที่กำหนดจะส่งข้อความว่า "False" กลับไปให้ client

- พารามิเดอร์ที่ส่งมา
  - msg = "shipAttack"
  - shipid = ไอดีของเรือ
- Return
  - ไอดีและตำแหน่งของเรือศัตรู

■ เมสเสจ townPosition

เป็นเมสเสจที่จะส่งตำแหน่งของเมืองทั้งหมดในแผนไปให้ไคลเอนต์

-พารามิเตอร์ที่ส่งมา

- msg = "townPosition"
- Return
  - ไอดีและตำแหน่งของเมืองทั้งหมด

■ เมสเสจ inTown

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นเมสเสจที่ใช้สำหรับเช็คว่าเรือของผู้เล่นอยู่ใกล้กับเมืองไหนมั้ย โดยระยะห่างระหว่างเรือกับเมืองต้องห่างกันไม่เกิน 50 UTM ซึ่งถ้าเรือของผู้เล่นอยู่ในระยะที่กำหนดเซิร์ฟเวอร์จะส่งไอดีของเมืองที่อยู่ในระยะที่กำหนดกลับไป แต่ถ้าไม่มีเมืองใดอยู่ในระยะที่กำหนดจะส่งข้อความ “False” กลับไป

- พารามิเตอร์ที่ส่งมา
  - msg = “inTown”
- Return
  - ไอดีและตำแหน่งของเมือง

■ เมสเสจ playerShip

เป็นเมสเสจสำหรับร้องขอไอดีของเรือที่ผู้เล่นเป็นเจ้าของทั้งหมด โดยไคลเอนต์จะมีการส่งตำแหน่งของผู้เล่นขณะนั้นมาด้วย เพื่อที่จะอัพเดตข้อมูลตำแหน่งของผู้เล่นลงฐานข้อมูล

- พารามิเตอร์ที่ส่งมา
  - msg = “playerShip”
  - playerid = ไอดีของผู้เล่น
  - utmlat = ค่า utmlat ของผู้เล่น
  - utmlong = ค่า utmlong ของผู้เล่น
- Return
  - ไอดีของเรือทั้งหมดที่ผู้เล่นเป็นเจ้าของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทดลองและสรุปผลการทดลอง

ในบทนี้จะแสดงการทดลองในส่วนต่างๆของตัวเกมและระบบ ว่าตัวเกมที่พัฒนาขึ้นมานั้น หน้าจอ อินเตอร์เฟสติดต่อกับผู้เล่นมีลักษณะการทำงานอย่างไร โดยจะเป็นการทดสอบการเล่นและจับภาพจากหน้าจอของอุปกรณ์พกพาจากการทดสอบเล่นจริง

5.1 อินเตอร์เฟสพื้นฐานภายในเกม

5.1.1 หน้าจอเมนูตอนเริ่มเกม

รูปที่ 5.1 แสดงหน้าจอแสดงเมนูตอนเริ่มเกม

ประกอบด้วยรายละเอียดดังต่อไปนี้

- Sign In เป็นส่วนที่ให้ผู้เล่นที่ลงทะเบียนแล้วเข้าเล่นเกม
- Create ID เป็นส่วนลงทะเบียนสำหรับผู้เล่นใหม่
- Option เป็นส่วนที่ใช้ปรับรายละเอียดต่างๆของตัวเกมไม่ว่าจะเป็นไอพีของเซิร์ฟเวอร์หรือการเปลี่ยนพอร์ตของจีพีเอส เป็นต้น
- Exit Game เป็นส่วนที่ใช้สำหรับออกจากเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 หน้าจอหลักภายในเกม

รูปที่ 5.2 แสดงหน้าจอหลักของตัวเกม

ส่วนประกอบของหน้าจอหลัก

1. แถบแสดงสินค้าต่างๆของตัวเรือ
2. ส่วนที่ใช้ในการแสดงแผนที่ที่ใช้ในการเล่น
3. แถบแสดงคำสั่งต่างๆที่ใช้ในการเล่นเกม เช่นการซูมเข้า/ออก, คำสั่งโจมตีเรือศัตรู, คำสั่งเทียบท่าเรือกับเมือง หรือคำสั่งเลือกเรือของผู้เล่นลำอื่น เป็นต้น
4. แถบแสดงรายละเอียดของผู้เล่น เช่นชื่อผู้เล่น, ชื่อเรือที่ผู้เล่นกำลังบังคับอยู่, คะแนนของผู้เล่น, จำนวนเงิน, Time Point และพลังชีวิตเรือที่ผู้เล่นบังคับ เป็นต้น

5.1.3 สัญลักษณ์ต่างๆภายในเกม

อาหาร

ลูกเรือ

แร่

เพชร

ทองใช้สำหรับซื้อของ

Time Point

เรือของผู้เล่น(สีฟ้า)

เรือของศัตรู(สีแดง)

เมืองท่า

5.1.4 วิธีการซูมเข้าและซูมออกแผนที่ภายในเกม

ในการทำการซูมเข้า/ออกของแผนที่นั้น ทำได้โดยเลือก "+" เพื่อทำการซูมเข้าหรือ "-" เพื่อทำการซูมออก แล้วกด OK เพื่อทำการซูมเข้า/ออก

วิธีการเลื่อนแผนที่ภายในเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเลื่อนแผนที่ภายในเกมนั้นสามารถทำได้โดยการเอาปากกาจิ้มบริเวณหน้าจอตรงแผนที่แล้วทำการเลื่อนไปซ้าย, ขวา, ขึ้นหรือลงแล้วอาปากกาขึ้น แผนที่ภายในเกมจะเลื่อนไปทางขวา,ซ้าย, ลงหรือขึ้นตามลำดับ

5.2 การทดลองการเคลื่อนที่ด้วยจีพีเอส

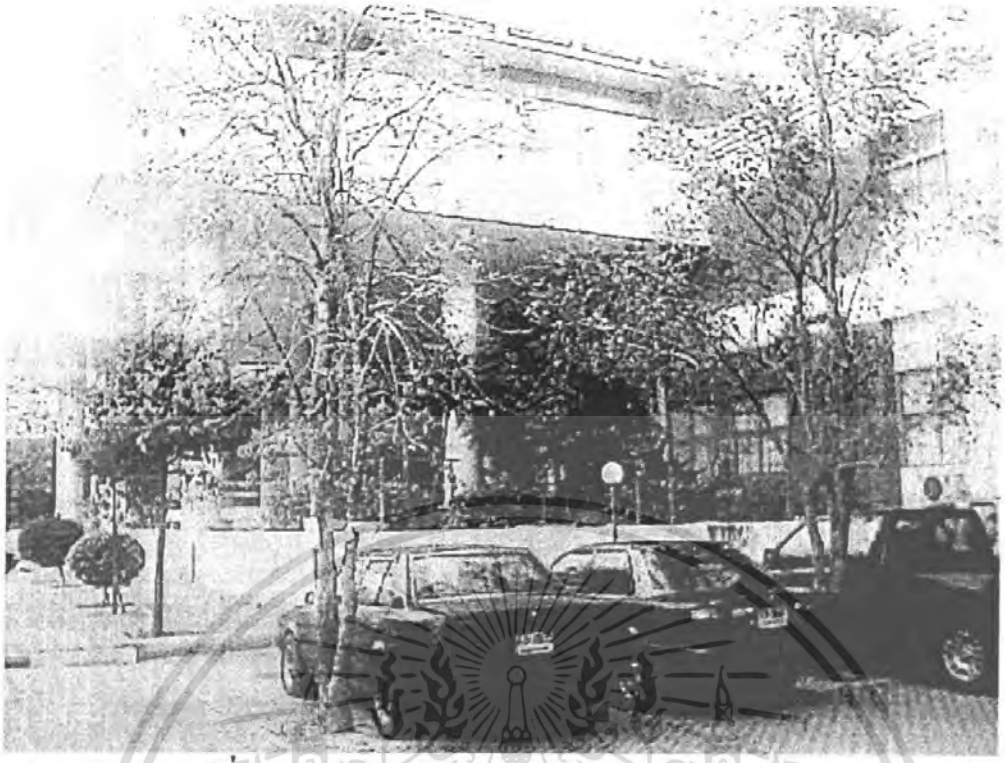
เป็นการทดสอบการทำงานของระบบระบุตำแหน่งในเกม โดยจะทำการเข้าเล่นเกมและเคลื่อนที่จากตำแหน่งหนึ่งไปยังตำแหน่งหนึ่ง โดยมีการวางแผนการทดลองดังต่อไปนี้

1. ผู้เล่นเข้าเกม ณ บริเวณ หน้าตึกปฏิบัติการ Electronic และทำการรอสัญญาณจีพีเอส ที่บริเวณนั้น

รูปที่ 5.3 แสดงจุดเข้าสู่เกมบริเวณอาคารปฏิบัติการ Electronic

2. ผู้เล่นทำการเคลื่อนที่จากบริเวณหน้าอาคารปฏิบัติการ Electronic ไปตามถนน จนถึงหน้าบริเวณตึก ECC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 แสดงปลายทางการเดินทดสอบบริเวณตึก ECC

3. ทำการบันทึกผลการทดลอง โดยการบันทึกจอภาพของเกม และการติดตามค่าในฐานข้อมูล ผลการทดลองที่ได้มีดังต่อไปนี้

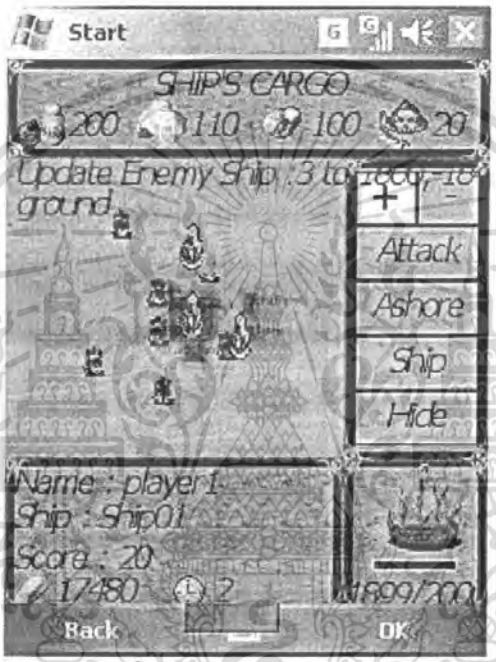
รูปที่ 5.5 แสดงหน้าจอแสดงผลภายในเกม ณ จุดเริ่มการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	shipid [PK] integer	shipname text	utmlat double precis	utmlong double precis	currenthp integer	maxhp integer	speed integer	defend integer	maxweight integer	value integer
1	1	Ship01	2020	-18530	1899	2000	60	120	1000	0
2	2	Ship02	2055	-18450	1200	1200	50	100	1000	2000

รูปที่ 5.6 แสดงหน้าฐานข้อมูลตำแหน่ง ณ จุดเริ่มการทดสอบ

เมื่อเข้าสู่เกมแล้ว รูปเรือของผู้เล่นจะปรากฏขึ้นจา ตำแหน่งดังภาพ (รูปเรือสีฟ้า) และส่งค่าไปบันทึกที่ฐานข้อมูลเรือ จากนั้นเมื่อเปลี่ยนตำแหน่งจากหน้าตึกปฏิบัติการ Electronic ไปยังบริเวณตึก ECCแล้วพบว่า การแสดงผลหน้าจอมีการเปลี่ยนแปลงดังภาพต่อไปนี้



รูปที่ 5.7 แสดงการเปลี่ยนแปลงหน้าจอการทดสอบการเคลื่อนที่

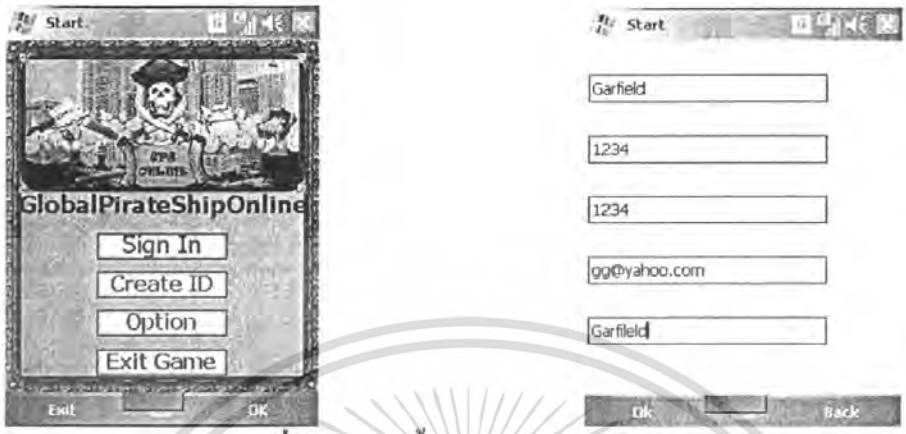
	shipid [PK] integer	shipname text	utmlat double precis	utmlong double precis	currenthp integer	maxhp integer	speed integer	defend integer
1	1	Ship01	2100	-18530	1899	2000	60	120
2	2	Ship02	2055	-18450	1200	1200	50	100

รูปที่ 5.8 แสดงการเปลี่ยนแปลงฐานข้อมูลตำแหน่งของการทดสอบการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การทดลองระบบการลงทะเบียนเกม

5.3.1 ขั้นตอนการลงทะเบียนสำหรับผู้เล่น



รูปที่ 5.9 แสดงขั้นตอนการลงทะเบียน

1. ทำการเลือก Create ID ที่เมนูหลัก เพื่อทำการลงทะเบียนเข้าเล่นเกม
2. กรองรายละเอียดของผู้เล่นต่างๆดังนี้
   - Playername ชื่อผู้เล่นที่ใช้ในการเข้าเล่นเกม
   - Password รหัสผ่านสำหรับเข้าเล่นเกม
   - E-mail อีเมล์ของผู้เล่น
   - Ship Name ชื่อเรือของผู้เล่นที่ใช้ในการเล่นเกม
3. กด OK เพื่อทำการยืนยันการลงทะเบียน

เมื่อผู้เล่นทำกายืนยันการลงทะเบียนของฝั่งไคลเอนต์แล้ว จะส่งข้อมูลของผู้เล่นไปให้เซิร์ฟเวอร์เพื่อทำการสร้างข้อมูลพื้นฐานของผู้เล่นในดาต้าเบสดังต่อไปนี้

	playerid [PK] integer	playername text	password text	utmlat double precis	utmlong double precis	gold integer	score integer	timepoint integer	e-mail text	currentshipid integer
1	1	Lord_Of_Wars	9321	0	0	0	0	0	crash_me@hotm	1
2	2	MaYu	3953723	0	0	0	0	0	mayu@yahoo.c	2
3	3	BlackPearl	1234	2500	-18300	0	0	0	long_doo@hotm	4
4	4	Ghost	4321	0	0	0	0	0	pepejang@hotm	5
5	5	Garfield	1234	0	0	0	0	0	gg@yahoo.com	7

รูปที่ 5.10 แสดงข้อมูลของผู้เล่นที่สร้างขึ้นในตาราง player

ในส่วนของตาราง player นั้น นอกจากจะมีการเก็บข้อมูลของผู้เล่นที่ส่งมาแล้ว ยังมีใส่ค่าข้อมูลพื้นฐานสำหรับให้ผู้เล่นใช้ในการเล่นเกมด้วยดังเช่น ตำแหน่งของผู้เล่น, จำนวนเงิน, ค่าคะแนนและไอดีของเรือของผู้เล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	shi [PK	shipname text	utmlat double	utmlong double pi	currenthp integer	maxhp integer	speed intege	defend integer	maxweight integer	value intege	totalweight integer	food integ	jewel intege	ore integ	crew integ
1	1	Ghost_Ship	1980	-18560	200	200	50	20	1000	2000	505	200	200	100	5
2	2	MaMa	0	0	200	200	50	20	1000	2000	0	0	0	0	5
3	3	MaZaKa	1940	-18420	200	200	50	20	1800	1400	700	350	120	200	30
4	4	Black_Pearl	2020	-18370	200	200	50	20	1000	2000	950	340	280	260	70
5	5	Ghost	2200	-18472	200	200	50	20	1000	2000	595	250	180	120	45
6	6	Black_pearl2	2060	-18400	0	500	50	30	2400	2800	600	300	150	100	50
7	7	Garfileld	0	0	200	200	50	20	1000	2000	5	0	0	0	5
8	8	Garfileld2	1920	-18472	300	300	50	20	1000	2000	600	300	150	70	80

รูปที่ 5.11 แสดงข้อมูลเรือของผู้เล่นที่สร้างขึ้นในตาราง ship

ในส่วนของตาราง ship จะมีการใส่รายละเอียดเรือของผู้เล่น โดยผู้เล่นใหม่ที่พึ่งลงทะเบียนนั้น จะมีเรือให้หนึ่งลำ ซึ่งค่าสเตตัสของเรือของผู้เล่นใหม่ จะมีค่าเท่านั้นหมดทุกลำดังรูปข้างต้น

	playerid [PK] integer	shipid [PK] integer
1	1	1
2	2	2
3	2	3
4	3	4
5	3	6
6	4	5
7	5	7
8	5	8

รูปที่ 5.12 แสดงข้อมูลที่ถูกสร้างขึ้นในตาราง playership

ในส่วนของตาราง playership จะมีการเก็บความสัมพันธ์ระหว่างผู้เล่นในตาราง player และเรือของผู้เล่นในตาราง ship ดังรูปข้างต้น

5.4 การทดลองระบบการ Sign In

5.4.1 ขั้นตอนการ Sign In สำหรับผู้เล่น

ในการเข้าเกมนั้นผู้เล่นจำเป็นต้องมี username และ password สำหรับลงทะเบียนก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.13 แสดงขั้นตอนการ Sign In สำหรับผู้เล่น

1. ทำการเลือกเมนู Sign In จากเมนูตอนเริ่มต้นเกมแล้วกด OK

รูปที่ 5.14 แสดงหน้าจอสำหรับกรอก username และ password

2. ทำการกรอก username และ password โดยจะต้องกรองให้ตรงกับที่ได้ทำการลงทะเบียนไว้ เมื่อกรอกเสร็จเรียบร้อยแล้วก็กด Login ถ้าผู้เล่นกรอก username และ password ถูกต้องจะสามารถเข้าเกมได้ แต่ถ้าไม่ถูกต้องผู้เล่นจะถูกส่งมากรอก username และ password ใหม่อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 การทดลองระบบต่อสู้

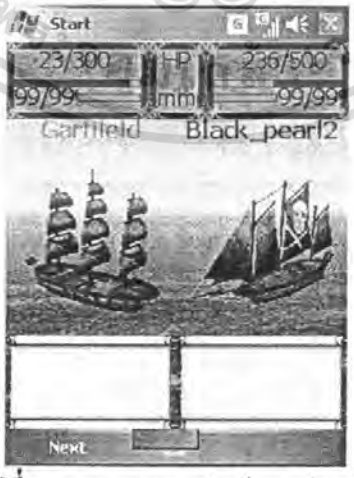
ในการต่อสู้กับผู้เล่นอื่นนั้น จะเป็นการเลือกเรือของผู้เล่นอื่นที่อยู่ใกล้ที่สุดมาเพื่อทำการต่อสู้ด้วย โดยเรือของผู้เล่นที่ถูกเลือกมานั้นจะต้องอยู่ห่างจากเรือของผู้เล่นไม่เกิน 50 หน่วย UTM จึงจะสามารถต่อสู้ด้วยได้ ถ้าไม่มีเรือของผู้เล่นอื่นอยู่ในระยะที่กำหนดจะไม่มีการต่อสู้กัน

4.4.1 การทดลองระบบต่อสู้ระหว่างเรือของผู้เล่นกับเรือของศัตรู

1. ทำการเลือก Attack ตรงเมนูในหน้าจอแผนที่ภายในเกม แล้วกด OK

รูปที่ 5.15 แสดงหน้าต่อสู้ระหว่างเรือ

2. เมื่อเลือกแล้วถ้ามีเรือของผู้เล่นอื่นอยู่ใกล้ๆ หน้าจอจะเปลี่ยนไปเป็นหน้าต่อสู้ระหว่างเรือของผู้เล่นกับเรือของศัตรู แต่ถ้าไม่มีเรือของผู้เล่นอื่นอยู่ใกล้จะไม่เป็นไปยังหน้าจอต่อสู้



รูปที่ 5.16 แสดงการต่อสู้ระหว่างเรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อเข้าไปหน้าจอต่อสู้แล้วนั้น ตัวเกมจะแสดงการต่อสู้ระหว่างเรือ โดยจะสลับการโจมตีเป็นตาๆไปจนกว่าเรือลำใดมีพลังชีวิตเป็น 0 ซึ่งผู้เล่นจะต้องกด next เพื่อดูการโจมตีต่อไป จนกว่าจะมีเรือลำใดแพ้ไป

	shipid [PK] ir	shipname text	utmlat double	utmlong double p	currenthp integer	maxhp integer	speed integer	defend integer	maxweight integer	value integ	totalweight integer	food integ	jewel integ	ore inte	crew integ
1	1	Ghost_Ship	1980	-18560	200	200	50	20	1000	2000	505	200	200	100	5
2	2	MaMa	0	0	200	200	50	20	1000	2000	0	0	0	0	5
3	3	MaZaKa	1940	-18420	200	200	50	20	1800	1400	700	350	120	200	30
4	4	Black_Pearl	2020	-18370	200	200	50	20	1000	2000	950	340	280	260	70
5	5	Ghost	2200	-18472	200	200	50	20	1000	2000	595	250	180	120	45
6	6	Black_pearl2	2060	-18400	0	500	50	30	2400	2800	600	300	150	100	50
7	7	Garfileld	2060	-18400	23	200	50	20	1000	2000	700	350	120	200	90
8	8	Garfileld2	1920	-18472	300	300	50	20	1000	2000	600	300	150	70	80

รูปที่ 5.17 แสดงข้อมูลเรือทั้งสองลำบนตาราง ship

เมื่อการต่อสู้ระหว่างเรือของผู้เล่นกับเรือของศัตรูเสร็จสิ้นแล้ว ตัวไคลเอนต์พลังชีวิตที่เหลือของเรือทั้งสองลำไปให้เซิร์ฟเวอร์

5.6 ทดลองระบบการซื้อขายทรัพยากรภายในเมือง

ในการซื้อขายสินค้าในเมืองนั้น ผู้เล่นต้องทำการเข้าไปในเมืองที่ผู้เล่นต้องการค้าขายด้วยก่อนก่อน ซึ่งกฎในการเข้าเมืองนั้นเรือของผู้เล่นจะต้องอยู่ห่างจุดศูนย์กลางของเมืองนั้นๆไม่เกิน 50 เมตรจึงจะสามารถเข้าเมืองได้ ถ้าผู้เล่นอยู่ห่างจากเมืองมากกว่าที่กำหนดผู้เล่นจะไม่สามารถเข้าเมืองได้

5.6.1 ขั้นตอนการซื้อขายทรัพยากรภายในเมือง

รูปที่ 5.18 แสดงการเข้าสู่เมือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

shipid [PK] int	shipname text	utmlat double	utmlong double pr	currenthp integer	maxhp integer	speed integer	defend integer	maxweight integer	value integer	totalweight integer	food integer	jewel integer	ore integer	crew integer
1	Ghost_Ship	1980	-18560	200	200	50	20	1000	2000	505	200	200	100	5
2	Garfileld	2031	-18442	300	300	50	20	1000	2000	600	300	150	70	80
3	MaZaKa	1940	-18420	200	200	50	20	1800	1400	700	350	120	200	30

รูปที่ 5.19 แสดง attribute ของตาราง ship ก่อนการเข้าเมือง

playerid [PK] inte	playername text	password text	utmlat double	utmlong double precis	gold integer	score integer	timepoint integer	e-mail text	currentshipid integer
1	Lord_Of_Wars	9321	1856	-18546	3254	2305	657	crash_me@hotm	1
2	MaYu	3953723	1978	-18305	2430	1250	540	mayu@yahoo.cc	10
3	BlackPearl	1234	2500	-18300	2378	2345	230	long_doo@hotm	4
4	Ghost	4321	1965	-18527	3577	1780	258	pepejang@hotm	5
5	Garfield	1234	2084	-18420	2400	3400	570	gg@yahoo.com	2

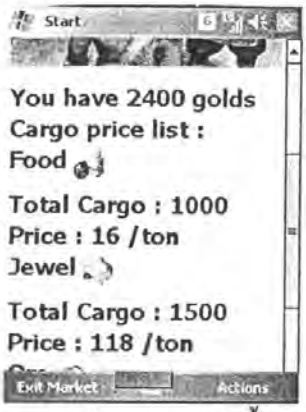
รูปที่ 5.20 แสดง attribute ของตาราง player ก่อนการเข้าเมือง

1. เมื่อเรือของผู้เล่นอยู่ในรัศมีของเมือง ให้ผู้เล่นเลือกเมนู Ashore แล้วกด OK เพื่อทำการเข้าเมือง

รูปที่ 5.21 แสดงหน้าจอภายในเมือง

2. เมื่อเข้าไปในเมืองแล้ว ทำการเลือกเมนู Market ตรง Actions ทางด้านซ้าย เพื่อทำการเข้าไปในหน้าการซื้อขายทรัพยากร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.22 แสดงหน้าตาของหน้าซื้อขายทรัพยากร

	townid [PK] int	townname text	utmlat double	utmlong double p	foodgrow integer	jewelgrow integer	oregrow integer	crewgrow integer	weightlimit integer	defenderid integer	totalweight integer	food integ	jewel intege	ore integ	cre inte
1	1	town1	1968	-18502	2	1	2	1	10000	3	4700	1000	1500	2000	200
2	2	town2	2062	-18479	2	1	1	1	10000	1	4700	1000	1500	2000	200
3	3	town3	1976	-18460	1	2	1	1	10000	2	4700	1000	1400	2000	300
4	4	town4	2307	-18514	2	1	1	1	10000	6	4600	1000	1400	2200	200
5	5	town5	1946	-18306	2	2	2	2	10000	4	4700	800	1600	2200	200

รูปที่ 5.23 แสดงตาราง town ในขณะที่ผู้เล่นเข้าเมือง

3. เมื่อเข้ามาในหน้าซื้อขายทรัพยากรแล้วจะมีการบอกจำนวนเงินที่ผู้เล่นมีอยู่, สินค้าที่มีขายในเมืองนั้น, จำนวนสินค้าในแต่ละชนิดที่มีอยู่ภายในเมืองและราคาในการซื้อขายสินค้าต่างๆ โดยผู้เล่นสามารถทำการซื้อหรือขายสินค้าก็ได้

รูปที่ 5.24 แสดงหน้าจอการซื้อ(รูปซ้าย) และขาย(รูปขวา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

playerid [PK] inte	playername text	password text	utmlat double p	utmlong double precis	gold integer	score integer	timepoint integer	e-mail text	currentshipid integer
1	Lord_Of_Wars	9321	1856	-18546	3254	2305	657	crash_me@hotm	1
2	MaYu	3953723	1978	-18305	2430	1250	540	mayu@yahoo.cc	10
3	BlackPearl	1234	2500	-18300	2378	2345	230	long_doo@hotm	4
4	Ghost	4321	1965	-18527	3577	1780	258	pepejang@hotm	5
5	Garfield	1234	2084	-18420	1860	3400	570	gg@yahoo.com	2

รูปที่ 5.25 แสดง attribute ของตาราง player หลังการทดลองซื้อ

สังเกตว่าหลังจากที่ผู้เล่นทำการซื้อเสร็จสิ้นแล้ว จำนวนเงินของผู้เล่นลดลงไป 540 (20*27=จำนวน*ราคา) ซึ่งสอดคล้องกับการซื้อที่ทำภายในเกม

shipid [PK] int	shipname text	utmlat double	utmlong double pr	currenthp integer	maxhp integer	speed integer	defend integer	maxweight integer	value integer	totalweight integer	food integer	jewel integer	ore integer	crew integer
1	Ghost_Ship	1980	-18560	200	200	50	20	1000	2000	505	200	200	100	5
2	Garfileld	2031	-18442	300	300	50	20	1000	2000	620	300	150	90	30
3	MaZaKa	1940	-18420	200	200	50	20	1800	1400	700	350	120	200	30

รูปที่ 5.26 แสดง attribute ของตาราง ship หลังการทดลองซื้อ

สังเกตว่าจำนวน Ore และน้ำหนักรวมของทรัพยากรภายในเรือของผู้เล่นเพิ่มขึ้น 20 ซึ่งแสดงให้เห็นว่าระบบทำงานได้ถูกต้อง

townid [PK] int	townname text	utmlat double	utmlong double p	foodgrow integer	jewelgrow integer	oregrow integer	crewgrow integer	weightlimit integer	defenderid integer	totalweight integer	food inte	jewel integ	ore inte	crew integ
1	town1	1968	-18502	2	1	2	1	10000	3	4700	1000	1500	200(	200
2	town2	2062	-18479	2	1	1	1	10000	1	4680	1000	1500	198(	200
3	town3	1976	-18460	1	2	1	1	10000	2	4700	1000	1400	200(	300
4	town4	2307	-18514	2	1	1	1	10000	6	4600	1000	1400	220(	200
5	town5	1946	-18306	2	2	2	2	10000	4	4700	800	1600	220(	200

รูปที่ 5.27 แสดง attribute ของตาราง town หลังการทดลองซื้อ

สังเกตว่าจำนวน Ore และน้ำหนักรวมของทรัพยากรภายในเมืองลดลงอย่างละ 20 ซึ่งสอดคล้องกับการกระทำของผู้เล่น

playerid [PK] inte	playername text	password text	utmlat double p	utmlong double precis	gold integer	score integer	timepoint integer	e-mail text	currentshipid integer
1	Lord_Of_Wars	9321	1856	-18546	3254	2305	657	crash_me@hotm	1
2	MaYu	3953723	1978	-18305	2430	1250	540	mayu@yahoo.cc	10
3	BlackPearl	1234	2500	-18300	2378	2345	230	long_doo@hotm	4
4	Ghost	4321	1965	-18527	3577	1780	258	pepejang@hotm	5
5	Garfield	1234	2084	-18420	2180	3400	570	gg@yahoo.com	2

รูปที่ 5.28 แสดง attribute ของตาราง player หลังการทดลองขาย

หลังจากที่ทำการทดลองขายจำนวนเงินของผู้เล่นเพิ่มจาก 1860 ไปเป็น 2180 ซึ่งเพิ่มขึ้น 320 (20*16) ซึ่งสอดคล้องกับการกระทำของผู้เล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

shipid [PK] int	shipname text	utmlat double	utmlong double pr	currenthp integer	maxhp integer	speed integer	defend integer	maxweight integer	value integer	totalweight integer	food integer	jewel integer	ore integer	crew integer
1	Ghost_Ship	1980	-18560	200	200	50	20	1000	2000	505	200	200	100	5
2	Garfileld	2031	-18442	300	300	50	20	1000	2000	600	280	150	90	80
3	MaZaKa	1940	-18420	200	200	50	20	1800	1400	700	350	120	200	30

รูปที่ 5.29 แสดง attribute ของตาราง ship หลังการทดลองขาย

จากการสังเกตจะเห็นว่าจำนวน food และน้ำหนักรวมของทรัพยากรลดลงอย่างละ 20 ซึ่งตรงกับข้อมูลที่ผู้เล่นได้กระทำ

townid [PK] int	townname text	utmlat double	utmlong double p	foodgrow integer	jewelgrow integer	oregrow integer	crewgrow integer	weightlimit integer	defenderid integer	totalweight integer	food integ	jewel intege	ore intec	crew integ
1	town1	1968	-18502	2	1	2	1	10000	3	4700	1000	1500	2000	200
2	town2	2062	-18479	2	1	1	1	10000	1	4700	1020	1500	1980	200
3	town3	1976	-18460	1	2	1	1	10000	2	4700	1000	1400	2000	300
4	town4	2307	-18514	2	1	1	1	10000	6	4600	1000	1400	2200	200
5	town5	1946	-18306	2	2	2	2	10000	4	4700	800	1600	2200	200

รูปที่ 5.30 แสดง attribute ของตาราง town หลังการทดลองขาย

จากการสังเกตจะเห็นว่าจำนวน food และน้ำหนักของทรัพยากรภายในเมืองลดลงไป 20 ซึ่งแสดงให้เห็นว่าระบบทำงานได้อย่างถูกต้อง

4. ทำการเลือกทรัพยากรที่ต้องการซื้อหรือขายและใส่จำนวนที่ต้องการซื้อหรือขาย ในการซื้อนั้นจะต้องคำนึงถึงจำนวนเงินที่ต้องใช้ในการซื้อ จำนวนทรัพยากรที่สามารถซื้อได้สูงสุด และน้ำหนักของทรัพยากรที่สามารถบรรทุกได้ของเรือ ส่วนการขายนั้นต้องคำนึงถึงจำนวนทรัพยากรที่ต้องใช้ในการขายและน้ำหนักของทรัพยากรที่เมืองสามารถมีได้ เมื่อเสร็จเรียบร้อยแล้วให้ทำการ confirm

5.7 ทดลองระบบการซ่อมแซมเรือ

เมื่อเรือของผู้เล่นเสียหายจากการโจมตี จะต้องไปทำการซ่อมแซมที่เมืองท่า โดยจะต้องเข้าไปในเมืองก่อนจึงจะสามารถซ่อมแซมเรือได้ ในการซ่อมแซมเรือนั้นจะต้องเสียค่าใช้จ่ายในการซ่อมด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.31 แสดงหน้าจอแสดงเมนูภายในเมือง

เมื่อผู้เล่นทำการเข้าไปในเมืองแล้ว เลือกเมนู Repair ตรง Actions เพื่อทำการซ่อมแซมเรือ ถ้าเรือไม่เสียหายจากการต่อสู้ จะไม่เสียเงินในการซ่อม

5.8 ทดลองระบบการอัพเกรดเรือและอาวุธ

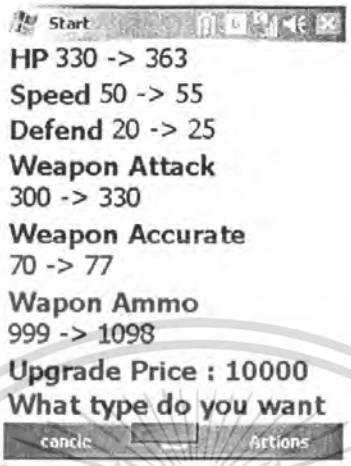
ภายในเกมนั้นผู้เล่นสามารถทำการอัพเกรดเรือและอาวุธของผู้เล่นได้ ในการอัพเกรดนั้นจะเสียค่าใช้จ่ายในการอัพเกรดครั้งละ 10,000 โดยผู้เล่นสามารถทำการอัพเกรดได้ที่เมืองท่า

5.8.1 ขั้นตอนการอัพเกรดเรือและอาวุธ

รูปที่ 5.32 แสดงเมนูภายในเมือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เมื่อผู้เล่นไปในเมืองแล้ว ให้เลือก Upgrade ที่เมนู Actions เพื่อทำการอัพเกรดเรือและอาวุธ



รูปที่ 5.33 แสดงหน้าจอสำหรับอัพเกรดเรือและอาวุธ

playerid [PK] integer	playername text	password text	utmlat double precis	utmlong double precis	gold integer	score integer	timepoint integer	e-mail text	currentshipid integer
2	MaYu	3953723	1978	-18305	2430	1250	540	mayu@yahoo.c	10
3	BlackPearl	1234	2500	-18300	2378	2345	230	long_doo@hotm	4
4	Ghost	4321	1965	-18527	3577	1780	258	pepejang@hotm	5
5	Garfield	1234	2084	-18420	21800	3400	570	gg@yahoo.com	2

รูปที่ 5.34 แสดงAttribute ต่างๆในตาราง player ก่อนการอัพเกรด

shipid [PK] ir	shipname text	utmlat double	utmlong double p	currenthp integer	maxhp integer	speed integer	defend integer	maxweight integer	value integ	totalweight integer	food integ	jewel integ	ore inte	crew integ
1	Ghost_Ship	1980	-18560	200	200	50	20	1000	2000	505	200	200	100	5
2	Garfileld	2031	-18442	330	330	50	20	1000	2000	600	280	150	90	80
3	MaZaKa	1940	-18420	200	200	50	20	1800	1400	700	350	120	200	30
4	Black_Pearl	2020	-18370	200	200	50	20	1000	2000	950	340	280	260	70
5	Ghost	2200	-18472	200	200	50	20	1000	2000	595	250	180	120	45

รูปที่ 5.35 แสดงAttribute ต่างๆในตาราง ship ก่อนการอัพเกรด

weaponid [PK] integer	weaponname text	attack integer	accurate integer	ammo integer	critical integer	value integer
1	normalgun	350	75	1432	5	20000
2	normalgun	750	50	700	5	3000
3	normalgun	550	85	1270	5	14000
4	normalgun	300	70	999	5	4000
5	normalgun	450	85	1800	5	22000
6	normalgun	300	140	1109	6	16000
7	normalgun	350	75	999	5	7000
8	normalgun	550	75	1012	5	12000

รูปที่ 5.36 แสดง Attribute ต่างๆในตาราง weapon ก่อนการอัพเกรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ในการอัพเกรดนั้นผู้เล่นสามารถทำการเลือกได้ว่าจะทำการอัพเกรดความสามารถทางด้านไหน ซึ่งในการอัพเกรดแต่ละครั้งนั้น จะเสียค่าใช้จ่าย 10,000 หน่วย โดยความสามารถที่ผู้เล่นสามารถอัพเกรดได้มีให้เลือกดังนี้

- HP หรือพลังชีวิตของเรือ
- Speed หรือความเร็วของเรือ
- Defend หรือค่าพลังป้องกันของเรือ
- Weapon Attack หรือค่าพลังโจมตีของอาวุธ
- Weapon Accurate หรือค่าความแม่นยำของอาวุธ
- Weapon Ammo หรือค่าความทนทานของอาวุธ

3. ทำการทดลองโดยการอัพเกรดพลังชีวิต(HP)ของเรือของผู้เล่น และตรวจสอบค่าต่างๆ ในฐานข้อมูลเพื่อดูว่าระบบการอัพเกรดทำงานได้ถูกต้องหรือไม่

playerid [PK] integer	playername text	password text	utmlat double precis	utmlong double precis	gold integer	score integer	timepoint integer	e-mail text	currentshipid integer
2	MaYu	3953723	1978	-18305	2430	1250	540	mayu@yahoo.c	10
3	BlackPearl	1234	2500	-16300	2378	2345	230	long_doo@hotm	4
4	Ghost	4321	1965	-18527	3577	1780	258	pepejang@hotm	5
5	Garfield	1234	2084	-18420	11800	3400	570	gg@yahoo.com	8

รูปที่ 5.37 แสดง Attribute ต่างๆของตาราง player หลังการอัพเกรด

จากการสังเกตภายในตารางจะเห็นว่าเงินของผู้เล่นลดไป 10,000 ตามการทำงานที่ได้ออกแบบไว้

shipid [PK] in	shipname text	utmlat double	utmlong double p	currenthp integer	maxhp integer	speed intege	defend integer	maxweight integer	value intege	totalweight integer	food integ	jewel integ	ore inte	crew integ
1	Ghost_Ship	1980	-18560	200	200	50	20	1000	2000	505	200	200	100	5
2	Garfileld	2031	-18442	300	363	50	20	1000	2000	600	280	150	90	80
3	MaZaKa	1940	-18420	200	200	50	20	1800	1400	700	350	120	200	30
4	Black_Pearl	2020	-18370	200	200	50	20	1000	2000	950	340	280	260	70
5	Ghost	2200	-18472	200	200	50	20	1000	2000	595	250	180	120	45
6	Black_pearl2	2060	-18400	0	500	50	30	2400	2800	600	300	150	100	50
8	Garfileld2	1920	-18472	300	300	50	20	1000	2000	600	300	150	70	80
10	MaMa	0	0	200	200	50	20	1000	2000	0	0	0	0	5

รูปที่ 5.38 แสดงAttribute ต่างๆของตาราง ship หลังการอัพเกรด

จากการสังเกตจะเห็นว่าค่า maxhp ของเรือเปลี่ยนจาก 330 เป็น 363 ซึ่งตรงกับหน้าจอที่แสดงไว้ให้ผู้เล่นดู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
บทวิจารณ์และสรุป

จากที่ได้ทำการศึกษาและพัฒนาโครงงานชิ้นนี้มาตลอดปีการศึกษา เห็นว่าโครงงานชิ้นนี้เป็นการพัฒนาแอพพลิเคชันที่รองรับเทคโนโลยีใหม่บนอุปกรณ์พกพาในที่มีหลากหลายในปัจจุบัน โดยจะเป็นแนวทางในการประยุกต์ระบบสารสนเทศทางภูมิศาสตร์ การระบุตำแหน่งด้วยระบบจีพีเอส รวมเข้ากับการพัฒนาแอพพลิเคชันเกมบนอุปกรณ์พกพา ออกมาเป็นเกมที่ใช้การแสดงผลกราฟิก และสามารถเชื่อต่อกับเครือข่ายเพื่อให้สามารถเล่นแบบออนไลน์ ให้สามารถสร้างความบันเทิงรูปแบบใหม่ในตลาดเกมบนอุปกรณ์พกพาได้ต่อไปในอนาคต

6.1 สิ่งที่ได้จากโครงงาน

6.1.1. สามารถออกแบบและพัฒนาเกมบนอุปกรณ์พกพาให้สามารถแสดงผลกราฟฟิก และเชื่อมต่อระบบเครือข่ายไปยังเซิร์ฟเวอร์ ได้ด้วยเทคโนโลยีจาว่า

6.1.2. สามารถออกแบบและพัฒนาแอพพลิเคชันบนเซิร์ฟเวอร์ จนสามารถให้บริการต่างๆ และเชื่อมต่อไปยังระบบฐานข้อมูล PostgreSQL

6.1.3. เข้าใจถึงกระบวนการทำงานของระบบระบุตำแหน่งด้วยจีพีเอส เพื่อสามารถนำไปประยุกต์ใช้กับแอพพลิเคชันที่พัฒนาบนอุปกรณ์พกพาได้

6.1.4. ได้ทำการศึกษาเทคโนโลยีต่างๆที่เกี่ยวข้องกับ ระบบสารสนเทศภูมิศาสตร์ และระบบระบีตำแหน่งจีพีเอส เพื่อนำไปประยุกต์ใช้กับการให้บริการทางสารสนเทศต่างๆได้ต่อไป

6.2 ปัญหาและอุปสรรคที่พบขณะดำเนินโครงงาน

6.2.1. ปัญหาข้อจำกัดของระบบ ซึ่งการพัฒนาด้วย Java ME บนอุปกรณ์พกพา ให้สามารถติดต่อกับตัวรับข้อมูลจีพีเอสได้นั้น จำเป็นต้องมีการเลือกใช้ Java Virtual Machine ให้สามารถรันแอพพลิเคชันเพื่อ เชื่อมต่อกับพอร์ตข้อมูลต่างๆได้ จึงทำให้เกิดปัญหาโดย Java Virtual Machine ที่มากับอุปกรณ์พกพา ไม่สามารถเชื่อมต่อกับพอร์ตข้อมูลได้ เลยทำให้ต้องทำการเปลี่ยน Java Virtual Machine ที่ใช้

6.2.2. ปัญหาการแสดงภาพแผนที่ SVG บน Java ME ที่จำเป็นต้องใช้หน่วยความจำที่มากในการอ่านข้อมูลจากไฟล์ ทำให้จำเป็นต้องพัฒนาแอพพลิเคชันโดยคำนึงถึงหน่วยความจำที่ใช้อย่างระมัดระวัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.3. ในการทดสอบการทำงานของตัวรับข้อมูลจีพีเอส ไม่สามารถทดสอบกับตัวจำลองการทำงาน (Emulator) เนื่องจากในการทดสอบจำเป็นต้องใช้กับอุปกรณ์และสถานที่จริง จึงต่อการหาทดสอบข้อผิดพลาดของโปรแกรม

6.3 ข้อเสนอแนะ

6.3.1. ในการระบุตำแหน่งด้วยจีพีเอส ถูกจำกัดด้วยตำแหน่งที่รับข้อมูล เช่นเมื่อตัวรับข้อมูลถูกบังด้วยผนังอาคาร ทำให้ไม่สามารถรับข้อมูลจากดาวเทียมได้ จึงมีความจำเป็นต้องออกมายังสถานที่โล่งแจ้ง เพื่อให้สามารถรับข้อมูลได้

6.3.2. การพัฒนาแอพพลิเคชันด้วย Java ME มีข้อจำกัดอยู่หลายอย่าง โดยเฉพาะข้อจำกัดทางด้านหน่วยความจำ ดังนั้นในการพัฒนาแอพพลิเคชันที่ใช้การแสดงผลทางกราฟิกระดับสูงจำเป็นต้องคำนึงถึงขนาดของไฟล์ภาพที่ใช้แสดงผล

6.4 แนวทางการพัฒนาต่อ

6.4.1. สามารถเพิ่มรายละเอียดของเกมลงไปเพื่อเพิ่มความหลากหลายของกิจกรรมต่างๆบนแผนที่ และเพื่อเพิ่มความสนุกสนานและความน่าสนใจของเล่นเกมเพิ่มยิ่งขึ้น

6.4.2. การนำแผนที่นอกจากเขตสถาบันมาประยุกต์ใช้กับเกมเพื่อขยายขอบเขตการเล่นเกมให้กว้างขวางยิ่งขึ้น

6.4.3. พัฒนาประสิทธิภาพ และความแม่นยำของการระบุตำแหน่งด้วยระบบจีพีเอส และการแปลงหน่วยพิกัดให้มีความแม่นยำยิ่งขึ้น

6.4.4. พัฒนาการระบบฐานข้อมูลให้เป็นฐานข้อมูลเชิงภูมิศาสตร์ เพื่อในการรับและส่งข้อมูลแผนที่ทางภูมิศาสตร์ได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

วิสุทธิ์ แซ่ตั้ง. 2546. "Open Source DBMS PostGreSQL" สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)

Dale DePriest. 2007. "**NMEA data**." [Online]. Available :
http://www.gpsinformation.org/dale/nmea.htm.

Daryl Wilding-McBride. 2007. "**Java Development on PDAs: Building Applications for PocketPC and Palm Devices.**" USA

Ludwig-Maximilians. 2004. **"Developers Training Material : the Java APIs for Bluetooth."** University Munich.

Peter H. Dana. 1994. " **The Global Positioning System.**" [Online].Available:
http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html.

PostgreSQL Global Development Group. **"PostgreSQL"** [Online] Available :
http://www.postgresql.org/docs/

Sun Microsystem. 2007. **"Java Comunity Process."** [Online].Available :
http://jcp.org/en/home/index

Sun Microsystem. **"Java Servlet Technology"** [Online].Available :
http://java.sun.com/products/servlet.

The Apache Software Foundation. **"Apache Tomcat 6.0"** [Online].Available :
http://tomcat.apache.org/tomcat-6.0-doc/introduction.html.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Netbean IDE 6.0

การพัฒนา Midlet ด้วย Netbean IDE 6.0

NetBeans เป็นเครื่องมือที่ใช้ในการช่วยพัฒนาโปรแกรมด้วยภาษา Java ที่มีประสิทธิภาพอย่างมาก สามารถใช้งานง่าย ทำให้พัฒนาระบบงานได้อย่างรวดเร็ว ที่สำคัญที่สุดคือเป็นเครื่องมือที่แจกฟรี โดยมี Sun Micro System เป็นผู้สนับสนุนโครงการ

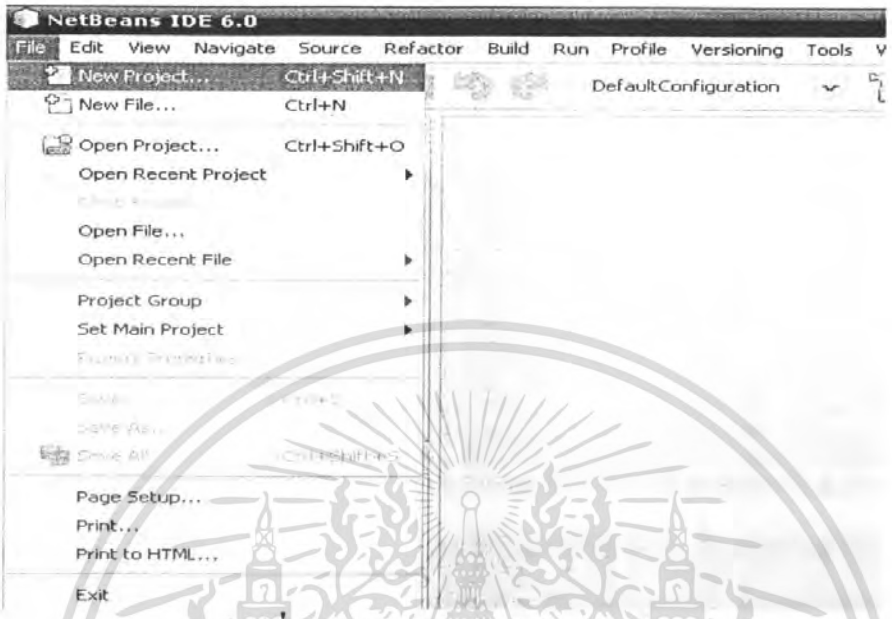
NetBeans IDE สามารถ Download ได้ที่ www.netbean.org

รูปที่ ก.1 แสดง NetBean IDE 6.0 เวอร์ชันล่าสุดที่ในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

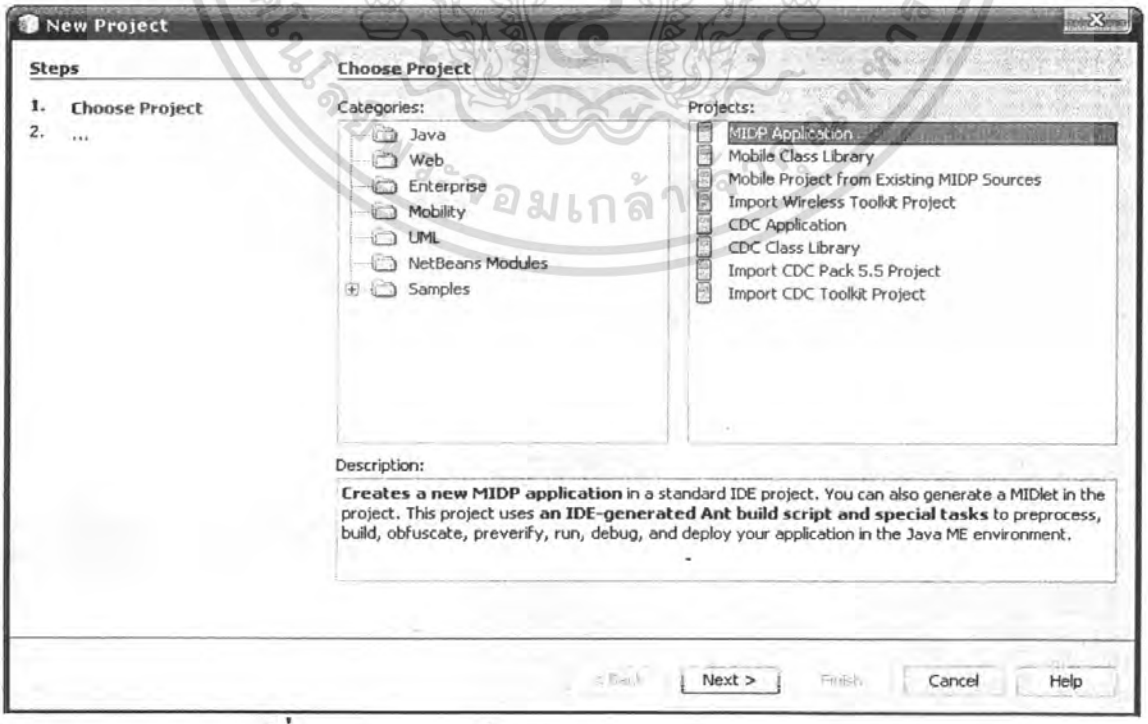
การพัฒนา Midlet ด้วย Netbean IDE 6.0 มีขั้นตอนดังต่อไปนี้

1. ทำการสร้างโปรเจกโดยเลือกที่ File -> New Project



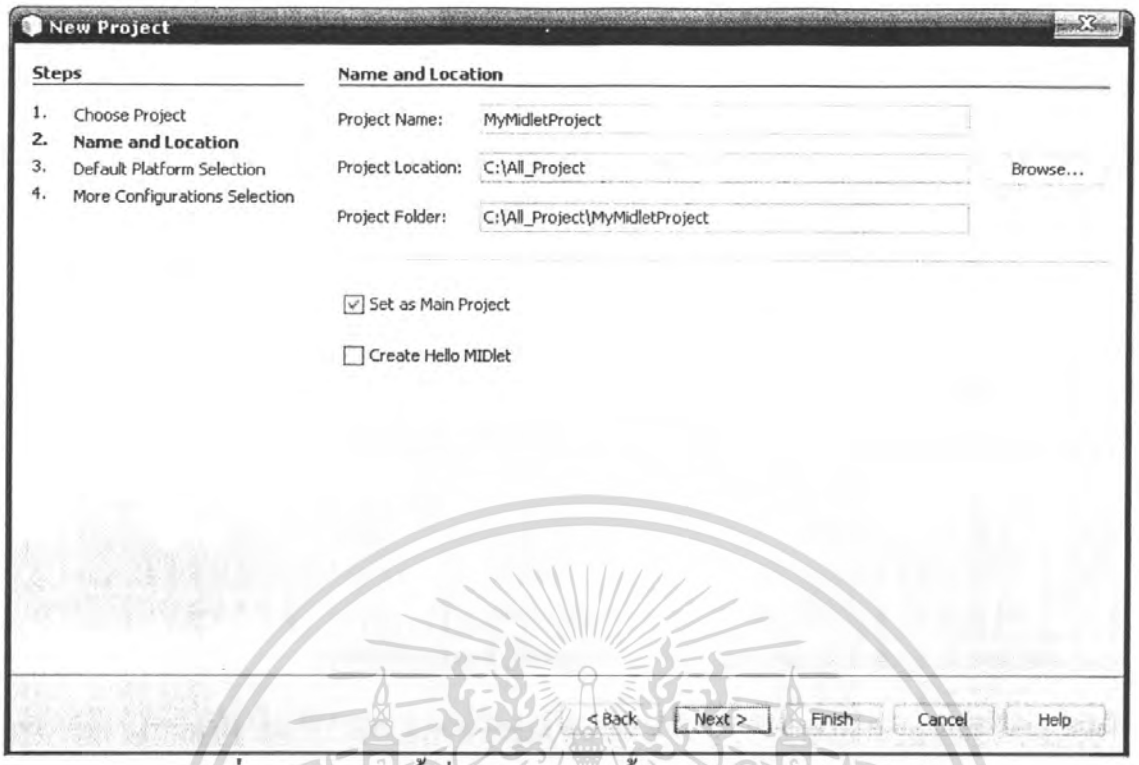
รูปที่ ก.2 แสดงการสร้างโปรเจกใหม่

2. เลือกไปที่แถบ Categories ไปที่ Mobility และเลือกชนิดของโปรเจกไปที่ MIDP Application เพื่อสร้างโปรเจกที่ใช้โปรไฟล์ MIDP ในการพัฒนา

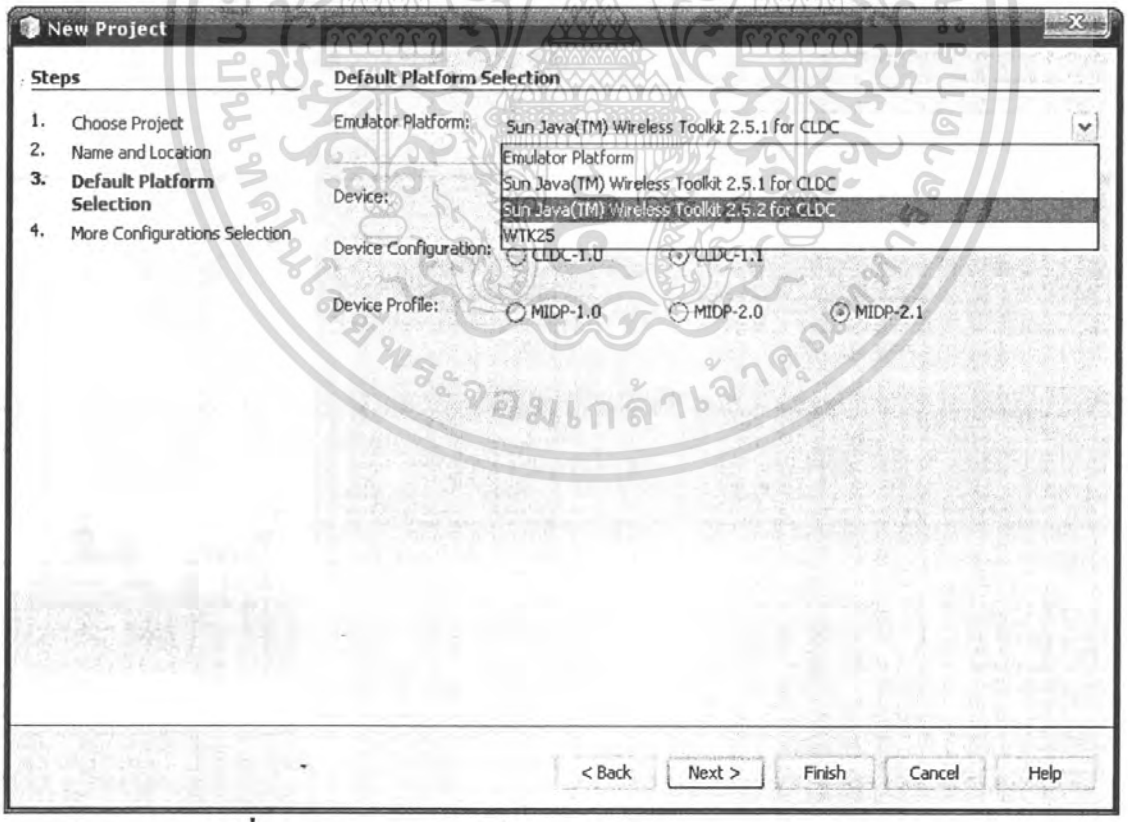


รูปที่ ก.3 แสดงการสร้างโปรเจกชนิด MIDP Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



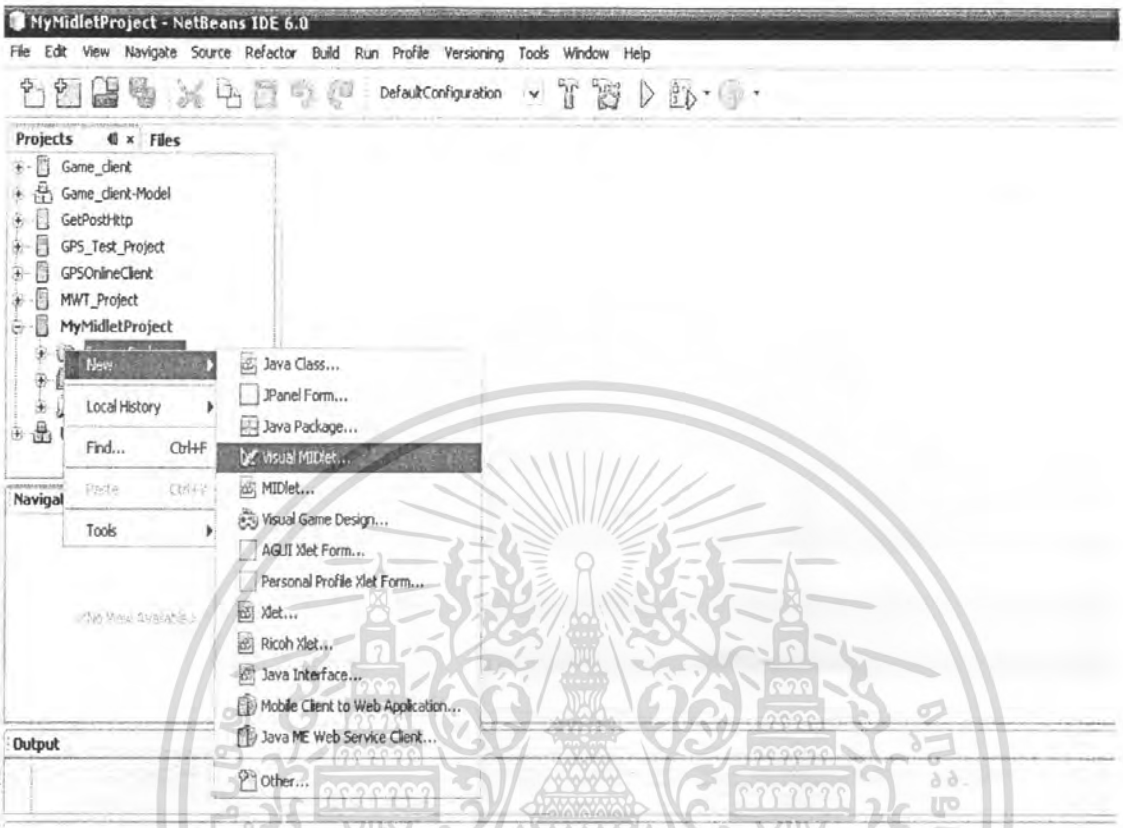
รูปที่ ก.4 แสดงการตั้งชื่อโปรเจก และสิ้นสุดการสร้างโปรเจกใหม่



รูปที่ ก.5 แสดงการเลือกใช้ Emulator ในการทดสอบการทำงาน

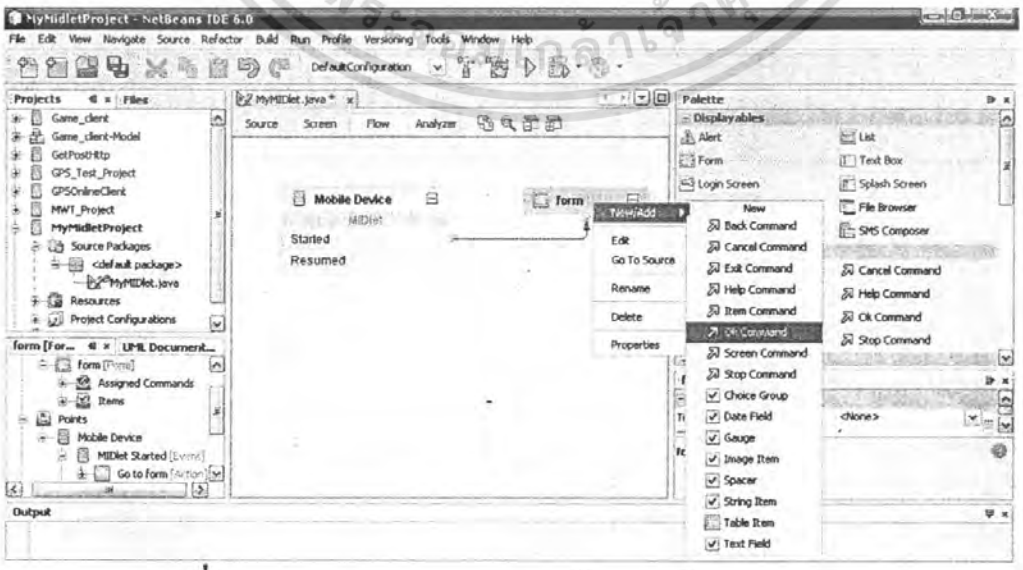
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การสร้าง Midlet นั้นสามารถเลือกใช้ Virtual Midlet เป็นตัวช่วยในการสร้างได้



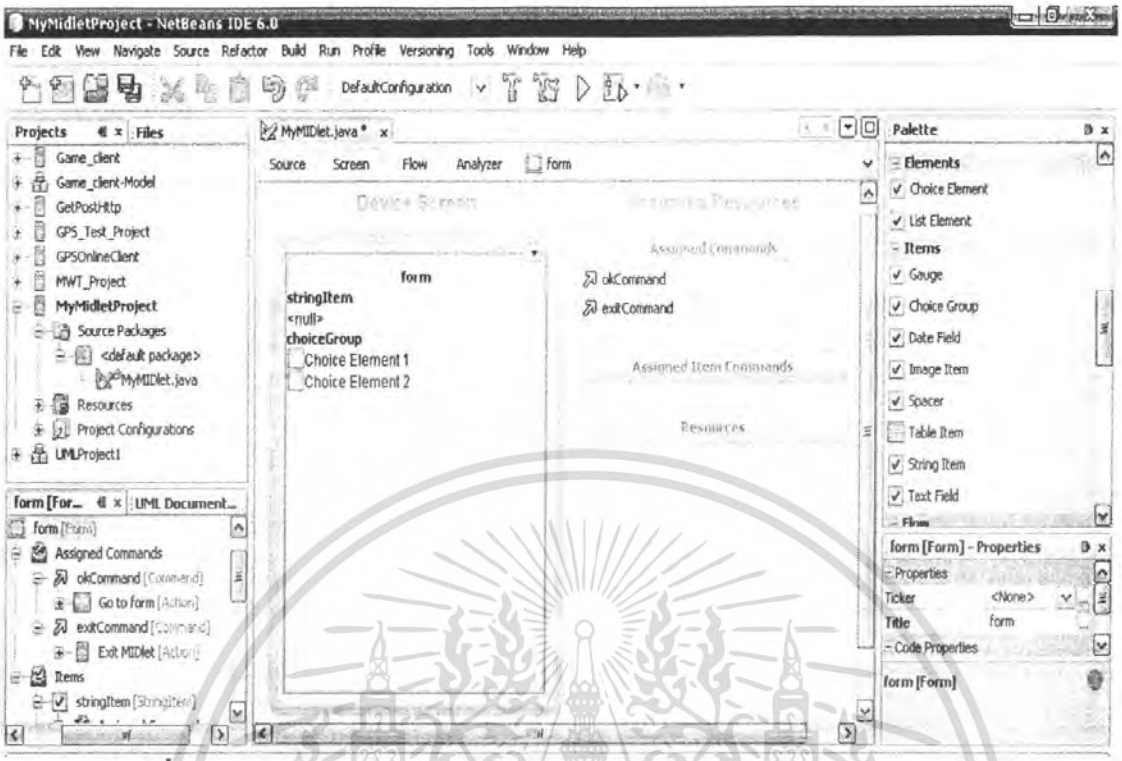
รูปที่ ก.6 แสดงการสร้าง Virtual Midlet

4. เมื่อทำการสร้าง Virtual Midlet แล้วทำให้สามารถใช้งาน Palatte ต่างๆได้ โดยไม่จำเป็นต้องเขียนโค๊ดเองใหม่ทั้งหมด



รูปที่ ก.7 แสดงการออกแบบ Midlet ด้วยการใช้ Virtual Midlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.8 แสดงการออกแบบองค์ประกอบต่างๆของ Form ด้วย Virtual Midlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้