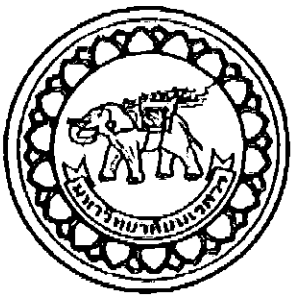


เกมออนไลน์บนมือถือด้วย J2ME

GAME ONLINE ON MOBILE BY J2ME

นายธวัชชัย จันทร์พุฒ รหัส 44362606

นายสุริยพงศ์ อัมพวา รหัส 44362804

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 25 มี.ค. 2553
เลขทะเบียน i 5000683
เลขเรียกหนังสือ ปร. ธ395ก
มหาวิทยาลัยนเรศวร
2547

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2547

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	เกมออนไลน์บนมือถือด้วย J2ME			
ผู้ดำเนินโครงการ	นาย ธวัชชัย	จันทร์พุฒ	รหัส	44362606
	นาย สุริยพงศ์	อัมพวา	รหัส	44362804
อาจารย์ที่ปรึกษา	อาจารย์ พงศ์พันธ์ กิจสนาโยธิน			
สาขาวิชา	วิศวกรรมคอมพิวเตอร์			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2547			

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(อาจารย์พงศ์พันธ์ กิจสนาโยธิน)

.....กรรมการ
(ดร.พนมขวัญ ริยะมงคล)

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

หัวข้อโครงงาน	เกมออนไลน์บนมือถือด้วย J2ME			
ผู้ดำเนินโครงงาน	นายธวัชชัย	จันทร์พุฒ	รหัส	44362606
	นายสุริยพงศ์	อัมพวา	รหัส	44362804
อาจารย์ที่ปรึกษา	อาจารย์พงศ์พันธ์ กิจสนาโยธิน			
สาขาวิชา	วิศวกรรมคอมพิวเตอร์			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2547			

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและพัฒนาโปรแกรมบนโทรศัพท์มือถือด้วยภาษาจาวา (JAVA) ในรูปแบบของ J2ME การพัฒนาออกมาในรูปแบบของเกมออนไลน์ที่มีผู้เล่นได้ 2 คนพร้อมกันบนเครือข่าย โดยใช้การติดต่อสื่อสารแบบซ็อกเก็ต (Socket) ระหว่างเครื่องแม่ข่าย (Server) เปิดบริการอยู่บนเครื่องคอมพิวเตอร์กับเครื่องลูกข่าย (Client) เป็นโปรแกรมประยุกต์ทำหน้าที่จำลองโทรศัพท์มือถือ

ผลที่ได้จากการทำโครงงานนี้ คือ โปรแกรมประยุกต์ที่สามารถทดสอบการทำงานบนตัวจำลองของโทรศัพท์มือถือซึ่งติดต่อกับเครื่องคอมพิวเตอร์ และรองรับการติดต่อสื่อสารกันแบบซ็อกเก็ตได้

Project Title	GAME ONLINE ON MOBILE BY J2ME	
Name	Mr. Thawatchai Janput	ID. 44362606
	Mr. Suriyapong Amphava	ID. 44362804
Project Advisor	Phongphun Kijsanayothin	
Major	Computer Engineering	
Department	Electrical and Computer Engineering	
Academic Year	2004	

ABSTRACT

This project study and develop an application on mobile using JAVA language with J2ME platform. The program was developed for game online. Two players can play a game through an emulator on computer, which communicate to each other by socket connection (server-client communication).

The result of this project is the application, which can examine how mobile work by an emulator, which connect to a personal computer through socket connection.

กิตติกรรมประกาศ

โครงงานวิศวกรรม เกมออนไลน์บนมือถือด้วย J2ME จะไม่สามารถสำเร็จได้ถ้าไม่ได้รับความช่วยเหลือจากบุคคลจากหลายๆ ฝ่าย โดยเฉพาะอย่างยิ่ง อาจารย์พงศ์พันธ์ กิจสนาโยธิน อาจารย์ที่ปรึกษาโครงงาน ซึ่งคอยให้คำปรึกษาที่ดีเพื่อการพัฒนาโปรแกรมให้มีประสิทธิภาพที่ดีขึ้น นอกจากนี้ยังมีกำลังใจและคำปรึกษาที่ได้รับจาก ครอบครัว และเพื่อนๆ วิศวกรรมคอมพิวเตอร์ ซึ่งสิ่งต่างๆ นี้คอยเป็นแรงผลักดันให้คณะผู้จัดทำมีกำลังใจในการทำให้โครงงานผ่านไปได้ด้วยดี

สุดท้ายนี้ต้องขอขอบคุณต่อผู้ที่มีส่วนช่วยเหลือคณะผู้จัดทำในทุกๆ ด้าน ซึ่งรวมไปถึงบุคคลที่ไม่ได้กล่าวถึงมา ณ ที่นี้ด้วย

นายธวัชชัย จันทร์พุฒ

นายสุริยพงศ์ อัมพวา

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูป ..... ซ

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตของโครงงาน 2

1.4 ขั้นตอนการดำเนินงาน 2

1.6 ผลที่คาดว่าจะได้รับ 3

1.7 งบประมาณที่ใช้ในการดำเนินงาน 3

บทที่ 2 หลักการและทฤษฎี

2.1 ภาษาจาวา (Java) คืออะไร 4

2.2 J2ME คืออะไร 5

2.3 J2ME สำหรับอุปกรณ์ไร้สายต่างๆ 8

2.4 หลักการเขียนโปรแกรม 9

2.4.1 การสร้างแพ็กเกจต่างๆ ของ CLDC (The CLDC Packages) 9

2.4.2 ประเภทข้อมูลต่างๆ ของ CLDC (The CLDC Data Types) 10

2.4.3 ข้อจำกัดต่างๆ ของ CLDC (CLDC Limitations) 10

2.4.4 ชุดคำสั่งจัดการโปรแกรมประยุกต์ 11

2.4.5 แพ็กเกจต่างๆ ของ MIDP (The MIDP Packages) 12

2.5 MIDlet คืออะไร 14

2.5.1 ความสำคัญของไฟล์ jar และ jad 14

2.5.2 แหล่งดาวน์โหลด MIDlet 15

สารบัญ(ต่อ)

หน้า

2.5.3 การพัฒนา MIDlet ทีละขั้น 16
2.5.4 การสร้าง MIDlet (Creating a MIDlet) 16
2.5.5 วงจรการทำงานของ MIDlet 17
2.6 เครื่องมือพื้นฐานสำหรับการพัฒนา J2ME 18
2.7 การพัฒนาโปแกรมบนโทรศัพท์มือถือ 19
2.7.1 ขั้นตอนการสร้างส่วนติดต่อกับผู้ใช้ 20
2.7.2 ส่วนติดต่อกับผู้ใช้ (Graphical User Interface : GUI) ของ MIDP 22
2.7.3 การวาดรูปโดยใช้ MIDlet แบบกราฟิก 23
2.7.4 การตรวจสอบเหตุการณ์ 26
2.7.5 รูปกราฟิกใน MIDlet แบบกราฟิก 28
2.8 การเขียนโปรแกรมเครือข่ายบนโทรศัพท์มือถือ โดยการเชื่อมต่อแบบซ็อกเก็ต 29

บทที่ 3 วิธีการดำเนินงาน
3.1 การออกแบบเกมในส่วนติดต่อผู้ใช้ 31
3.2 การทำงานของโปรแกรมประยุกต์ฝั่งเครื่องแม่ข่าย (Server) 32
3.3 การทำงานของโปรแกรมประยุกต์ฝั่งเครื่องลูกข่าย (Client) 35

บทที่ 4 ผลการดำเนินงาน
4.1 การทดสอบการทำงาน 37
4.2 วิธีการทดสอบการทำงาน 37
4.3 ผลการดำเนินงาน 41

บทที่ 5 บทสรุป
5.1 บทสรุป 51
5.2 ข้อเสนอแนะ 51

สารบัญ(ต่อ)

หน้า

เอกสารอ้างอิง 52

ภาคผนวก 53

ประวัติผู้เขียนโครงงาน 61

สารบัญตาราง

ตารางที่		หน้า
2.1	ค่าสีที่มีโอกาสใช้งาน	24
2.2	ค่าของปุ่มกดต่างๆ ในตัวจำลองของ WTK	27

สารบัญรูป

รูปที่		หน้า
2.1	แสดงการเปรียบเทียบระหว่าง J2SE และ J2ME	6
2.2	แสดงรูปแบบของ J2ME สำหรับโปรแกรมประยุกต์ต่างๆ ของอุปกรณ์ไร้สาย	8
2.3	แสดงคลาสไฟล์ที่อยู่ในไฟล์ Bomb.jar	14
2.4	แสดงข้อมูลที่อยู่ในไฟล์ Bomb.jar	15
2.5	แสดงวงจรการทำงานของ MIDlet	17
2.6	ระบบพิกัดร่วมของกราฟิก	23
3.1	แผนภาพแสดงขั้นตอนการทำงานของเกมออนไลน์บนมือถือ	30
3.2	แผนภาพแสดงขั้นตอนการทำงานของเครื่องแม่ข่าย	32
3.3	แสดงการเริ่มต้นของเกม	34
3.4	แผนภาพแสดงขั้นตอนการทำงานของเครื่องลูกข่าย	35
4.1	แสดงการเปิดโปรแกรมประยุกต์ฝั่งแม่ข่าย	38
4.2	แสดงโปรแกรมประยุกต์ตัวจำลองของโทรมือถือ	38
4.3	แสดงการสร้างโปรเจ็กต์ใหม่	39
4.4	แสดงการเปิดโปรเจ็กต์	39
4.5	แสดงการคอมไพล์ของโปรเจ็กต์	40
4.6	แสดงการคอมไพล์ที่มีข้อผิดพลาดเกิดขึ้น	40
4.7	แสดงตัวจำลองของโทรศัพท์มือถือ	41
4.8	แสดงหน้า LOGIN ของเกมออนไลน์	42
4.9	แสดงการใส่ชื่อเพื่อลงทะเบียน	42
4.10	แสดงการเริ่มต้นของเกม	43
4.11	แสดงการเดินของผู้เล่น	44
4.12	แสดงการวางระเบิด	45
4.13	แสดงการระเบิด	46
4.14	แสดงการวางระเบิด 2 ลูก	47
4.15	แสดงการถูกไฟของตัวละคร	48
4.16	แสดงการตายของตัวละคร	49
4.17	แสดงการจบเกม	50

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันเทคโนโลยีต่างๆ มีความเจริญก้าวหน้าไปมาก ยกตัวอย่างเช่น คอมพิวเตอร์ซึ่งมีประสิทธิภาพที่สูงขึ้นและได้รับการพัฒนาโปรแกรมให้สามารถนำไปประยุกต์ใช้ประโยชน์ให้เหมาะสมได้ตามความต้องการของลักษณะการใช้งาน อุปกรณ์ขนาดเล็กก็เช่นเดียวกัน ถึงแม้ว่าอุปกรณ์ขนาดเล็กมีข้อจำกัดในการพัฒนามากกว่าคอมพิวเตอร์ แต่อย่างไรก็ตามในปัจจุบันอุปกรณ์ขนาดเล็กก็ได้รับการพัฒนาขึ้นมาอย่างมาก อีกทั้งยังเพิ่มขีดความสามารถทางด้านต่างๆ ไม่ว่าจะเป็นส่วนที่ติดต่อผู้ใช้ (GUI : Graphic User Interface) เสียงที่มีเครื่องดนตรีต่างๆ (Polyphonic) กราฟิก (Graphic) ที่สวยงามทำให้ดูน่าใช้ขึ้น และที่สำคัญคือสามารถให้นักพัฒนาหรือผู้ใช้ได้พัฒนาโปรแกรมได้

การรับรู้ข่าวสารและข้อมูล การติดต่อสื่อสารเป็นเรื่องสำคัญมาก อุปกรณ์ไร้สายได้สนองความต้องการของผู้ใช้ ตัวอย่างเช่น เครื่องช่วยงานส่วนบุคคลแบบดิจิตอล (PDA: Personal Digital Assistant) โทรศัพท์มือถือ (Mobile Phone) ในโลกแห่งเทคโนโลยี (Technology) การใช้โทรศัพท์มือถือในการสนทนาแลกเปลี่ยนข่าวสารอย่างเดียวคงจะไม่เพียงพอต่อความต้องการอีกต่อไป ผู้ใช้ต้องการโปรแกรมประยุกต์ (Application) บนโทรศัพท์มือถือที่สามารถทำงานโต้ตอบ และกำหนดค่าความต้องการได้มากขึ้น ซึ่งสิ่งเหล่านี้จะช่วยในการยกระดับการดำเนินชีวิต และการทำธุรกิจให้ง่ายกว่าเดิม

ในการสร้างโปรแกรมประยุกต์บนโทรศัพท์มือถือ ผู้ผลิตได้ติดตั้งคลังโปรแกรม (Library) เฉพาะระบบไว้ให้ภาษาที่ใช้ในการเขียนโปรแกรมประยุกต์ก็มีหลายภาษา ตั้งแต่ภาษาซีพลัสพลัส (C++) ภาษาวิชวลเบสิก (Visual Basic) ไปจนถึงภาษาที่มีสคริปต์เฉพาะตัว โปรแกรมประยุกต์ที่เขียนขึ้นจึงเขียนเพื่อใช้กับอุปกรณ์หนึ่งไม่สามารถนำไปใช้กับอุปกรณ์อื่นๆได้ ภาษาจาวา (Java) นั้นได้สามารถที่จะนำไปใช้กับแพลตฟอร์ม (Platform) ต่างๆ ได้

บริษัทซันไมโครซิสเต็มส์ (Sun Microsystems) ออกแพลตฟอร์มใหม่เพิ่มเติมที่นำไปใช้กับอุปกรณ์ขนาดเล็กคือ J2ME (Java 2 Micro Edition) ซึ่งได้รับการออกแบบโดยเน้นกลุ่มผู้ใช้อุปกรณ์อิเล็กทรอนิกส์ ซึ่งรวมถึงอุปกรณ์ไร้สาย เช่น โทรศัพท์มือถือและเครื่องช่วยงานส่วนบุคคลแบบดิจิตอล

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 สามารถเข้าใจ J2ME โดยแพลตฟอร์มของ J2ME จะลดแพ็กแกจ (Package) ของภาษาจาวาที่ใช้บนเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer) ลงเพื่อกำหนดขอบข่ายของงาน

1.2.2 สามารถศึกษาและพัฒนาโปรแกรมประยุกต์ทางด้านเครือข่ายในโทรศัพท์มือถือและสนองความต้องการผู้ใช้ในปัจจุบัน

1.2.3 สามารถนำเสนอแนวคิดถึงโปรแกรมประยุกต์แบบเครือข่าย และหลักการติดต่อ สื่อสารแบบซ็อกเก็ต (Socket) เพื่อเป็นแนวทางในการนำไปใช้ในการติดต่อสื่อสารแบบอื่นๆ

1.3 ขอบเขตของโครงงาน

ขอบเขตของโครงงานนี้จะใช้ J2ME ซึ่ง J2ME จะมีความแตกต่างกับภาษาจาวาทั่วไปตรงที่จะลดความสามารถต่างๆ ให้น้อยลงและในโครงงานนี้จะใช้ตัวบรรณาธิการ (Editor) ในการนำเสนอการทำงานและแสดงถึงโปรแกรมประยุกต์ที่พัฒนาขึ้น และต้องการนำเสนอในด้านเครือข่ายที่มีเครื่องแม่ข่าย (Server) และเครื่องลูกข่าย (Client) ให้เครื่องแม่ข่าย รับการติดต่อการทำงานจากเครื่องลูกข่าย และจะกระจาย(Broadcast)ไปที่เครื่องลูกข่าย ทุกเครื่องที่เข้ามาอยู่ในเครือข่าย ซึ่งโปรแกรมประยุกต์ที่พัฒนานั้นจะเป็นเกมที่ใช้เครือข่ายซ็อกเก็ตในการติดต่อซึ่งแต่ละการกระทำของผู้เล่นจะส่งไปประมวลผลที่เครื่องแม่ข่าย และกระจายไปให้ผู้เล่นทุกคนเห็น

1.4 ขั้นตอนการดำเนินงาน

กิจกรรม	12	1	2	3	4	5	6	7	8	9	10	11
1. ศึกษาและรวบรวมข้อมูล	←							→				
2. ออกแบบโปรแกรมประยุกต์						←						→
3. เขียนโปรแกรมและทดสอบการทำงาน	←			→								
4. แก้ไขข้อบกพร่อง			←			→						
5. จัดทำคู่มือโครงงาน				←		→						
6. ตรวจสอบปรับปรุงแก้ไขคู่มือ					←	→						

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 ได้รับความรู้เกี่ยวกับการเขียนโปรแกรมภาษาจาวาและระบบเครือข่ายทางโทรศัพท์มือถือ (Mobile Network System)

1.5.2 สามารถทดสอบและทดลองเพื่อนำไปใช้ได้จริง

1.5.3 เกิดแนวคิดในการพัฒนาประสิทธิภาพและความเสถียรภาพของระบบมากขึ้น

1.6 งบประมาณที่ใช้

- ค่าหนังสือ	800 บาท
- ค่าถ่ายเอกสารและจัดทำรูปเล่ม	600 บาท
- ค่าวัสดุคอมพิวเตอร์	500 บาท
- ค่าวัสดุสำนักงาน	100 บาท
รวม	2,000 บาท

บทที่ 2
หลักการและทฤษฎี

2.1 ภาษาจาวา (Java) คืออะไร

ภาษาจาวา (Java) ได้กลายเป็นภาษาในการพัฒนาโปรแกรมประยุกต์ (Application) แบบเชิงวัตถุ (Object-oriented) ที่มีจุดเด่นตรงที่สามารถทำงานได้ทุกระบบปฏิบัติการ (Write Once, Run Anywhere) โดยที่ไม่ต้องแปลภาษา (Compile) โปรแกรมใหม่ หลักการทำงานของภาษา คือ โปรแกรมที่เขียนด้วยภาษาจาวาจะถูกนำไปแปลภาษาโดยตัวแปลภาษา (Compiler) ของภาษาจาวา ให้เป็นกลุ่มคำสั่ง (Bytecode) ที่เรียกว่าคลาส (Class) โดยกลุ่มคำสั่งเหล่านี้จะถูกนำไปใช้โดยเครื่องจักรเสมือนของจาวา (JVM : Java Virtual Machine) เพื่อแปลความหมาย (Interpreted) เป็นภาษาเครื่องของแต่ละระบบปฏิบัติการอีกที ด้วยการทำงานในลักษณะนี้ การพัฒนาโปรแกรมโดยจาวาจึงไม่ต้องคำนึงถึงระบบปฏิบัติการที่จะมารองรับการทำงานของโปรแกรม (Machine Independent) ในปัจจุบันรุ่น (Version) ของภาษาจาวา คือ รุ่นที่ 2 (Java 2) โดยแบ่งออกได้เป็น 3 ระดับ ดังนี้

1. J2EE (Java 2 Enterprise Edition) ออกแบบสำหรับใช้พัฒนาโปรแกรมประยุกต์ขนาดใหญ่ทั่วไปบนเครื่องแม่ข่าย (server) ส่วนใหญ่โปรแกรมประยุกต์จะเกี่ยวข้องกับงานธุรกิจขนาดใหญ่ต่างๆ

2. J2SE (Java 2 Standard Edition) ออกแบบสำหรับใช้พัฒนาโปรแกรมประยุกต์ต่างๆ บนเครื่องลูกข่าย (client) หรือเครื่องคอมพิวเตอร์ส่วนบุคคลที่ใช้ตามบ้านเรือนหรือสำนักงานทั่วไป

3. J2ME (Java 2 Micro Edition) ออกแบบมาสำหรับโปรแกรมประยุกต์ของอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็กต่างๆ เช่น โทรศัพท์มือถือ, วิทยุติดตามตัว (Pager), เครื่องช่วยงานส่วนบุคคลแบบดิจิตอล (PDA: Personal Digital Assistant) และเตาไมโครเวฟ (Microwave)

2.2 J2ME คืออะไร

อุปกรณ์ประเภทมือถือส่วนใหญ่จะมีขนาดเล็กและมีทรัพยากรจำกัด เช่น ขนาดหน้าจอ, หน่วยความจำ และหน่วยประมวลผล (Processor) เป็นต้น โดยหน้าจอของโทรศัพท์มือถือจะมีความละเอียดของหน้าจอประมาณ 12,288 (96×128) จุด (pixel) เครื่องช่วยงานส่วนบุคคลแบบดิจิตอลจะอยู่ที่ 20,000 ขึ้นไป และหน่วยความจำของอุปกรณ์ในปัจจุบันก็จะมีขนาดไม่เกิน 4 เมกะไบต์ ทางบริษัทซันไมโครซิสเต็มส์ (Sun Microsystems) จึงได้พัฒนา J2ME ขึ้นมาเพื่อให้เป็นภาษาจาวาที่มีขนาดเล็กที่สุดที่สามารถติดตั้งบนอุปกรณ์เหล่านี้ได้

กลุ่มคลาสของ J2ME แบ่งออกได้เป็นสองกลุ่ม กลุ่มแรกเป็นกลุ่มที่พัฒนาขึ้นมาใหม่และกลุ่มที่สองเป็นคลาสบางส่วนของ J2SE สาเหตุที่ไม่สามารถใช้งานคลาสของ J2SE ได้ทั้งหมดก็เพราะคลาสบางตัวใช้ทรัพยากรของระบบมากและมีรูปแบบการทำงานที่ไม่เหมาะสมกับอุปกรณ์ เช่น คลาส AWT (Abstract Window Toolkit) ซึ่งออกแบบมาเพื่อใช้งานกับระบบปฏิบัติการวินโดว์ (Windows) เป็นต้น

นอกจากนี้โปรแกรมประยุกต์ที่พัฒนาด้วย J2ME ยังสามารถติดตั้งเพิ่มบนอุปกรณ์ประเภทมือถือได้อีกด้วย ซึ่งแต่เดิมโปรแกรมประยุกต์ของอุปกรณ์เหล่านี้จะต้องติดตั้งมาจากโรงงานผู้ผลิตเท่านั้น

ในปัจจุบันความหลากหลายของเทคโนโลยี (Technology) ทำให้อุปกรณ์ประเภทมือถือแต่ละชนิดมีลักษณะแตกต่างกันออกไป ไม่เว้นแม้แต่อุปกรณ์ในกลุ่มเดียวกัน ที่เห็นชัดเจนที่สุดก็คือ ขนาดหน้าจอและหน่วยความจำที่มีขนาดเล็ก เพื่อให้ความหลากหลายของเทคโนโลยีเหล่านี้อยู่ในขอบเขตของ J2ME และสามารถให้การสนับสนุนโปรแกรมประยุกต์ต่างๆ ให้ทำงานได้บนอุปกรณ์ขนาดเล็ก J2ME จึงกำหนดชุดโปรแกรมสามชั้นสำหรับสร้างระบบปฏิบัติการของอุปกรณ์เหล่านี้ดังนี้

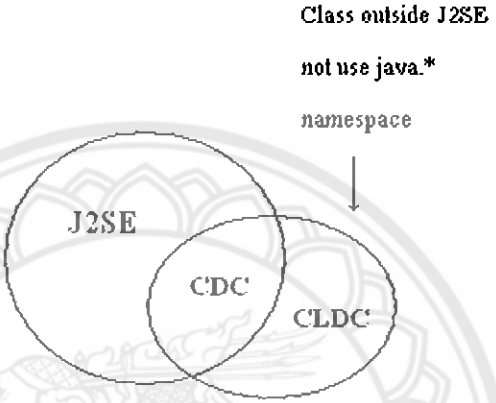
1. ชั้นโครงแบบ (Configuration Layer)

ชั้นโครงแบบ คือ คลาสคลังโปรแกรม (Library) พื้นฐานสำหรับอุปกรณ์มือถือทุกชนิด การที่จะให้กลุ่มคลาสเหล่านี้ครอบคลุมทุกอุปกรณ์นั้น จำเป็นจะต้องนำอุปกรณ์มาจัดกลุ่มโดยใช้คุณสมบัติพื้นฐานของอุปกรณ์ คือ หน่วยความจำ, การแสดงผล, การเชื่อมต่อสัญญาณ และการใช้พลังงาน ปัจจุบันสามารถแบ่งออกได้เป็น 2 กลุ่มดังนี้

1. CDC (Connect Device Configuration)
   - ใช้หน่วยความจำอย่างต่ำ 512 กิโลไบต์ สำหรับจาวา
   - ใช้หน่วยความจำอย่างต่ำ 256 กิโลไบต์ ขณะเวลาดำเนินงาน (Run time)
   - การเชื่อมต่อสัญญาณมีขนาดของช่องส่งสัญญาณ (Bandwidth) สูง

2. CLDC (Connected, Limited Device Configuration)

- ใช้หน่วยความจำ 128 กิโลไบต์ สำหรับจาวา
- ใช้หน่วยความจำ 32 กิโลไบต์ ขณะเวลาดำเนินงาน
- มีส่วนที่ติดต่อกับผู้ใช้ (User Interface) ที่จำกัด
- ใช้พลังงานน้อย
- เชื่อมต่อสัญญาณด้วยขนาดของช่องส่งสัญญาณต่ำ



รูปที่ 2.1 แสดงการเปรียบเทียบระหว่าง J2SE และ J2ME

จากรูปที่ 2.1 แสดงการเปรียบเทียบภาษาจาวาเวอร์ชันต่าง ๆ และตัวอย่างของอุปกรณ์ในแต่ละกลุ่ม จะเห็นว่าอุปกรณ์ในกลุ่ม CDC นั้นมีทรัพยากรของระบบมาก ทำให้สามารถใช้กลุ่มคลาสคลังโปรแกรมเหมือนกับ J2SE ได้ แต่สำหรับอุปกรณ์ในกลุ่ม CLDC เช่น โทรศัพท์และเครื่องช่วยงานส่วนบุคคลแบบดิจิตอล จะไม่สามารถใช้คลาสจาก J2SE ได้ทั้งหมด ดังนั้นจึงต้องพัฒนาคลาสคลังโปรแกรมขึ้นใหม่ โดยมีข้อจำกัดคือ

- ไม่รองรับการทำงานกับข้อมูลชนิดเลขทศนิยม (Floating Point) ทำให้ไม่มีคลาส java.lang.Float และ java.lang.Double รวมถึงวิธี (Method) ส่งผลลัพธ์กลับมาเป็นข้อมูลชนิดนี้ด้วย
- ไม่มีวิธีการทำให้สิ้นสุด (Finalization) ในคลาส java.lang.Object เพื่อให้ขั้นตอนการคืนทรัพยากรให้กับระบบ (Garbage Collector) ทำได้ง่ายขึ้น
- การจับผิดข้อผิดพลาดในขณะทำงานมีเพียงคลาส java.lang.OutOfMrmoryError, java.lang.Error และ java.lang.VirtualMachineError

เมื่อกำหนดกลุ่มคลาสคลังข้อมูลพื้นฐานสำหรับอุปกรณ์ทั้งสองกลุ่มแล้ว จะต้องกำหนดกลุ่มคลาสเพื่อรองรับลักษณะที่แตกต่างกันของแต่ละอุปกรณ์ซึ่ง เรียกว่า โพรไฟล์ (Profiles)

2. ชั้นโพรไฟล์ (Profiles Layer)

ชั้นโพรไฟล์ คือ ส่วนเพิ่มเติมจากชั้นโครงแบบเป็นกลุ่มคลาสคลังข้อมูลที่รองรับข้อแตกต่างของอุปกรณ์เช่น ชั้นโพรไฟล์ของอุปกรณ์ประเภทมือถือคือ MIDP (Mobile Information Device Profile) จะเป็นกลุ่มคลาสคลังข้อมูลต่างๆ ที่รองรับการพัฒนาโปรแกรมประยุกต์บนอุปกรณ์ประเภทมือถือ เช่น โทรศัพท์มือถือหรือเครื่องช่วยงานส่วนบุคคลแบบดิจิตอล เป็นต้น กลุ่มคลาสคลังข้อมูลที่กล่าวมานี้ประกอบด้วยการเชื่อมต่อกับผู้ใช้ (UI : User Interface), การรับ-ส่งข้อมูล Input-Output), Event Handing, การเก็บข้อมูลที่คงทน (Persistent Storage), เครือข่าย (Networking) และตัวจับเวลา (Timers)

3. ชั้นเครื่องจักรเสมือน (Virtual Machine Layer)

โปรแกรมภาษาจาวาทุกตัวจะต้องทำงานภายใต้เครื่องจักรเสมือนจาวา (JVM : Java Virtual Machine) เสมอ เมื่อแปลภาษาโปรแกรมจาวาเป็นคลาสแล้ว JVM จะทำหน้าที่แปลงคลาสเหล่านั้นเป็นภาษาเครื่องและทำงานตามคำสั่งต่อไป

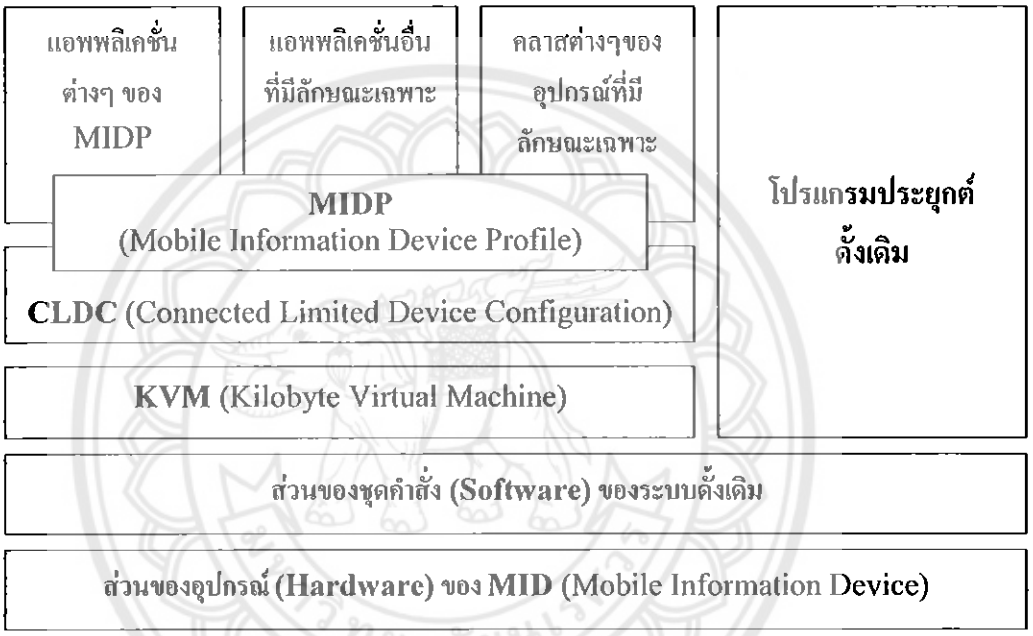
ใน J2ME จะใช้โครงแบบเป็นตัวกำหนด JVM โดยในกลุ่ม CDC ซึ่งใช้กลุ่มคลาสชุดเดียวกับ J2ME ทำให้สามารถใช้ JVM ตัวเดียวกันได้ แต่กลุ่ม CLDC ไม่ได้ใช้กลุ่มคลาสเดียวกันทั้งหมดทำให้ต้องพัฒนา JVM ขึ้นมาใหม่ เรียกว่า K Virtual Machine หรือ KVM
(K ย่อมาจาก Kilo มาจากขนาดของหน่วยความจำ) สำหรับคุณสมบัติของ KVM ก็มีดังนี้

- ทำงานที่หน่วยความจำเพียง 40 และ 80 กิโลไบต์
- ทำงานที่หน่วยประมวลผล 16 บิต และมีความเร็วในการประมวลผล 25 เมกะเฮิรตซ์

2.3 J2ME สำหรับอุปกรณ์ไร้สายต่างๆ

J2ME ช่วยสร้างโปรแกรมประยุกต์ยุคใหม่บนอุปกรณ์ไร้สาย เช่น เกมแบบหลายผู้เล่นที่ทำงานผ่านอินเตอร์เน็ต, การทำธุรกรรมทางโทรศัพท์มือถือ, โปรแกรมประยุกต์สำหรับเครื่องลูกข่ายและเครื่องแม่ข่ายบนโทรศัพท์มือถือ และวิทยุติดตามตัวแบบรับส่งข้อความ

วิธีการที่ MIDP, CLDC และ KVM ทำงานร่วมกันเพื่อให้การสนับสนุนโปรแกรมประยุกต์ต่างๆ ของอุปกรณ์ไร้สาย สถาปัตยกรรมของ J2ME สำหรับโปรแกรมประยุกต์ต่างๆ ของอุปกรณ์ไร้สาย แสดงดังรูปที่ 2.2



รูปที่ 2.2 แสดงรูปแบบของ J2ME สำหรับโปรแกรมประยุกต์ต่างๆ ของอุปกรณ์ไร้สาย

สถาปัตยกรรมของ J2ME สำหรับโปรแกรมประยุกต์ต่างๆ ของอุปกรณ์ไร้สาย สามารถแยกประเภทออกเป็น 5 ชั้น จากล่างสุดไปบนสุด ดังนี้

1. ชั้นส่วนอุปกรณ์ของ MID (MID hardware layer) อ้างถึงโทรศัพท์มือถือ เช่น IDEN3000 ของ Motorola หรือวิทยุติดตามตัวสองทาง เช่น RIM Blackberry 9.50

2. ชั้นชุดคำสั่งของระบบดั้งเดิม (Native system software layer) บรรจุระบบปฏิบัติการดั้งเดิม และคลังโปรแกรมต่างๆ ของระบบที่โรงงานผลิตอุปกรณ์ให้การสนับสนุน

3. ชั้น KVM (KVM layer) ให้การสนับสนุน ส่วนเชื่อมต่อประสานโปรแกรม (API: Application program interface) หลักของจาวาสำหรับโปรแกรมประยุกต์ต่างๆ ของจาวา

4. ชั้น CLDC (CLDC layer) ให้การสนับสนุน ส่วนต่อประสานโปรแกรม ของจาวาหลักสำหรับโปรแกรมประยุกต์ต่างๆ ของอุปกรณ์ไร้สาย

5. ชั้น MIDP (MIDP layer) ให้การสนับสนุนคลังโปรแกรมต่างๆ ของ ส่วนที่ติดต่อผู้ใช้ (GUI: Graphic User Interface) คลังโปรแกรมของที่จัดเก็บข้อมูล คลังโปรแกรมต่างๆ ของเครือข่าย และคลาสต่างๆ ของตัวกำหนดเวลา

นอกจากคลาสคลังโปรแกรมสำหรับ MIDP ผู้ผลิตอาจเตรียมคลาสคลังโปรแกรมเฉพาะอุปกรณ์ไว้ให้นักพัฒนา เพื่อดึงความสามารถของฟังก์ชั่นที่มีอยู่แล้วไปใช้งานให้เกิดประโยชน์สูงสุด เช่น การใช้ข้อมูลร่วมกับโปรแกรมประยุกต์ที่ติดตั้งมาในเครื่อง ตัวอย่างโปรแกรมประยุกต์ ได้แก่ ปฏิทิน, สมุดจดที่อยู่, การตรวจสอบอายุแบตเตอรี่ และความแรงของสัญญาณ แม้ว่าการนำคลาสเฉพาะอุปกรณ์ที่ผู้ผลิตเตรียมไว้มาใช้งาน จะช่วยเพิ่มความสามารถแก่โปรแกรมประยุกต์บนอุปกรณ์ไร้สาย แต่ไม่สามารถเคลื่อนย้ายไปสู่อุปกรณ์อื่นๆ ที่ใช้ MIDP ได้ เนื่องจากคลาสที่นำมาใช้อยู่นอกเหนือขอบเขตของ MIDP

2.4 หลักการเขียนโปรแกรม

การเขียนโปรแกรมของ J2ME คล้ายกับการเขียนโปรแกรมของ J2SE หรือ J2EE ต่างกันตรงที่อุปกรณ์ต่างๆ ใน J2ME จะจำกัดการใช้ทรัพยากรต่างๆ ในการเขียนโปรแกรม (Develop Program of Source code) จะมุ่งไปที่อุปกรณ์ไร้สาย ซึ่งจะเกี่ยวข้องกับ MIDP สภาวะแวดล้อมช่วงเวลาในการทำงานของจาวาบนอุปกรณ์ต่างๆ ของ MIDP ประกอบด้วย

- KVM (K Virtual Machine)
- CLDC (Connected Limited Device Configuration)
- คลังโปรแกรมต่างๆ ของ MIDP
- AMS (Application Management Software)

2.4.1 การสร้างแพ็กเกจต่างๆ ของ CLDC (The CLDC Packages)

แพ็กเกจ(Package) ต่างๆ ของ CLDC ให้การสนับสนุนอุปกรณ์ต่างๆ ที่มีข้อจำกัดในการเชื่อมโยงในระบบระดับสูงและแพ็กเกจต่างๆ ของเครือข่าย คลาสต่างๆ จะมีสองประเภทที่ใช้ใน CLDC ได้แก่

- คลาสต่างๆ ที่เป็นส่วนหนึ่งของคลาสใน J2SE
- คลาสต่างๆ ที่เกี่ยวข้องกับการเชื่อมโยงทั่วไปของ CLDC

คลาสต่างๆ ของ CLDC เป็นคลาสประเภทแรกที่บรรจุในแพ็กเกจส่วนหนึ่งของ J2SE คลาสต่างๆ ได้กำหนดไว้ในแพ็กเกจต่อไปนี้

- java.lang
- java.util
- java.io

คลาสต่างๆ ในแพ็กเกจดังกล่าวจะคล้ายกับที่ใช้ใน J2SE หรือ J2EE คลาสต่างๆ ซึ่งกำหนดไว้ในแพ็กเกจ javax.microedition.io.*

คลาสต่างๆ เหล่านี้เป็นคุณลักษณะเฉพาะของ CLDC แต่จะเข้ากันไม่ได้กับแพ็กเกจต่างๆ ของ J2SE คลาสต่างๆ ที่เกี่ยวข้องกับเครือข่ายจะอยู่ในแพ็กเกจ java.io และ java.net

คลาสต่างๆ ในแพ็กเกจ จะให้การสนับสนุนการสื่อสารทางเครือข่ายสำหรับอุปกรณ์ทั่วไปของ J2ME คลาสต่างๆ ในโครงสร้างการติดต่อทั่วๆ ไป (GCF : Generic Connection Framework) จะให้การสนับสนุนการสื่อสารประเภทต่างๆ เช่น socket, datagram, serial และ http (Hyper Text Transport Protocol)

2.4.2 ประเภทข้อมูลต่างๆ ของ CLDC (The CLDC Data Types)

CLDC ให้การสนับสนุนประเภทข้อมูลดั้งเดิมที่เป็นส่วนหนึ่งของ J2SE ได้แก่ byte, short int, long, char และ boolean ส่วนข้อมูลประเภท float และ double จะไม่ให้การสนับสนุน เนื่องจากเหตุผลต่อไปนี้

- อุปกรณ์หลักของ CLDC ไม่มีสำหรับใช้เลขทศนิยม
- ค่าใช้จ่ายในการใช้เลขทศนิยมในชุดโปรแกรมมีราคาสูงเกินไปสำหรับอุปกรณ์ของ CLDC

CLDC ยังกำหนดคลาสสำหรับประเภทข้อมูลดั้งเดิม ดังนี้

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

2.4.3 ข้อจำกัดต่างๆ ของ CLDC (CLDC Limitations)

CLDC ได้ออกแบบมาสำหรับอุปกรณ์ต่างๆ พร้อมกับจำกัดการใช้ทรัพยากรต่างๆ ในบางกรณีอาจเป็นไปได้ที่ CLDC ให้ความช่วยเหลือคุณลักษณะทั้งหมด และภารกิจที่ J2SE ให้การสนับสนุนบางคุณลักษณะใน J2SE จะไม่มีใช้ใน CLDC เพื่อลดพื้นที่และปรับปรุงสมรรถนะในการทำงานให้สูงขึ้น ดังนั้น จึงต้องทำความเข้าใจรายละเอียดของ J2SE ก่อนเริ่มต้นออกแบบและพัฒนาโปรแกรมประยุกต์ต่าง ๆ ของ J2ME ซึ่งมีข้อจำกัดต่าง ๆ ดังนี้

1. ไม่ให้การสนับสนุนเลขทศนิยม (No Support Floating-Point) : CLDC จะไม่ให้การสนับสนุนเลขทศนิยม (Floating-point) ทำให้ไม่สามารถใช้ประเภทข้อมูล float และ double ในโปรแกรมต่าง ๆ ของ J2ME นั่นคือไม่สามารถใช้คลาส java.lang.Float และ java.lang.Double

2. ไม่ให้การสนับสนุน Finalization (No Support Finalization) : Finalization เป็นกระบวนการการลบค่าของวัตถุ (Object) ต่างๆ ในหน่วยความจำ เมื่อโปรแกรมไม่ต้องใช้งานอีกต่อไป วิธี finalize() จะช่วยจัดการงานดังกล่าวให้ คลังโปรแกรมของ CLDC ไม่ได้รวมวิธี finalize() จากคลาส Object เนื่องจาก finalization จะไม่ให้การสนับสนุนใน CLDC ก็เพื่อให้การทำความสะอาดทำได้ง่ายขึ้น วัตถุต่างๆ ในหน่วยความจำของ KVM หากไม่สร้าง finalize() ไว้ใช้งาน คลาส Object จะสร้างวิธี finalize() ของการทำให้สิ้นสุด เพื่อคืนหน่วยความ จำของวัตถุให้โดยอัตโนมัติ

3. ข้อจำกัดต่างๆ ในภาษานานาชาติ (Internationalization Limitations) : CLDC จะให้การสนับสนุนภาษานานาชาติด้วยคลาส InputStreamReader และ OutputStreamReader

4. ข้อจำกัดต่างๆ ในการเชื่อมโยงกับข้อผิดพลาด (Error-Handling Limitations) : จาวาจะมีสองคลาสหลักที่ใช้ในการตรวจสอบข้อผิดพลาดที่เกิดจากการทำงานของโปรแกรมประยุกต์ต่างๆ ได้แก่ คลาส java.lang.Exception และ java.lang.Error

2.4.4 ชุดคำสั่งจัดการโปรแกรมประยุกต์ (AMS : Application management software)

AMS เป็นโปรแกรมประยุกต์ของชุดคำสั่งพื้นฐานที่มาพร้อมกับเครื่องมือต่างๆ ของ MIDP การดำเนินการของ AMS จะขึ้นอยู่กับอุปกรณ์นั้นๆ แต่การให้บริการพื้นฐานต่างๆ จะมีลักษณะอย่างเดียวกัน ดังนี้

1. AMS จะให้การสนับสนุนผู้ใช้ในการติดตั้งหรือไม่ติดตั้ง MIDlet ในอุปกรณ์ไร้สายทั่วไป รวมทั้งการส่งสารสนเทศแบบอนุกรมไปตามสายเคเบิล (Cable) ที่เชื่อมโยงกับคอมพิวเตอร์หรือส่งผ่านทางอินเตอร์เน็ต

2. AMS ให้การสนับสนุนสภาวะแวดล้อมของการประมวลผลสำหรับ MIDlet ภายหลังจากที่ MIDlet เริ่มต้นทำงาน AMS จะจัดการทรัพยากรของระบบ เช่น คลาสต่างๆ ของ CLDC, MIDP และ KVM นอกจากนี้ยังช่วยจัดการทรัพยากรต่างๆ รวมทั้งไฟล์ต่างๆ ในโปรแกรมประยุกต์ให้ MIDlet ในช่วงเวลาที่ทำงาน

3. AMS จะเชื่อมโยงกับข้อผิดพลาดทั้งหมดที่เกิดขึ้นระหว่างการติดตั้งและการประมวลผลของโปรแกรมประยุกต์ต่างๆ โดยไม่ทำให้ระบบเกิดความเสียหาย

2.4.5 แพ็กเกจต่างๆ ของ MIDP (The MIDP Packages)

แพ็กเกจต่างๆ ของ CLDC ให้การสนับสนุนภารกิจที่เป็นอิสระต่ออุปกรณ์เชื่อมต่อประสานโปรแกรม แพ็กเกจของ MIDP ให้การสนับสนุนภารกิจที่มีคุณลักษณะเฉพาะ ภารกิจนี้ได้รวมการจัดการโปรแกรมประยุกต์ต่างๆ บนอุปกรณ์, การเชื่อมโยงกับผู้ใช้ทางกราฟิก (Graphic), อุปกรณ์จัดเก็บสารสนเทศ และขยายความสามรถของเครือข่าย

1. แพ็กเกจสำหรับจัดการโปรแกรมประยุกต์ (Package for Application Management)

แพ็กเกจ javax.microedition.midlet บรรจุลงคลาส MIDletStateChangeException และ MIDlet เพื่อใช้จัดการโปรแกรมประยุกต์ของอุปกรณ์ไร้สาย คลาส MIDlet เป็นคลาสพื้นฐานสำหรับ MIDlet ทั้งหมด ซึ่งคล้ายกับแอปเพล็ต (Applet) ของภาษาจาวาตรงที่มีวิธีการต่างๆ ให้ใช้งานบ้างเล็กน้อยเท่านั้น คลาสทั้งสองจะช่วยตอบสนองการเปลี่ยนแปลงสถานะต่างๆ การเปลี่ยนแปลงสถานะจะเกิดจากการที่ผู้ใช้โต้ตอบกับโปรแกรมประยุกต์ของ MIDlet ภายใต้สภาวะของการพัฒนา MIDlet สามารถรายงานปัญหาต่างๆ ระหว่างการเปลี่ยนแปลงสถานะ โดยการโยน MIDletStateChangeException ทิ้งไป เมื่อมีข้อผิดพลาดเกิดขึ้น

โปรแกรมประยุกต์ต่างๆ ของ MIDP ทั้งหมด จำเป็นต้องสร้างคลาสที่มีคุณลักษณะเหมือนคลาส MIDlet ที่บรรจุในแพ็กเกจนี้ และต้องรองรับการสนับสนุนจากวิธีต่อไปนี้

- startApp()
- pauseApp()
- destroyApp()

2. แพ็กเกจสำหรับติดต่อกับผู้ใช้ (Package for User Interfaces)

คลาส AWT (Abstract Windowing Toolkit) ใน J2SE ใช้ออกแบบโปรแกรมประยุกต์ต่างๆ บนหน้าจอ AWT จะเข้มงวดกับการใช้หน่วยความจำมากสำหรับอุปกรณ์ไร้สายที่จำกัดการใช้ทรัพยากร MIDP จึงได้กำหนดแพ็กเกจสำหรับสนับสนุนการติดต่อกับผู้ใช้และการเชื่อมโยงกับเหตุการณ์ต่างๆ กลุ่มของ API ได้แก่ คลาสต่างๆ ที่เกี่ยวข้องกับการติดต่อกับผู้ใช้ทางกราฟิก รวมทั้งการเชื่อมโยงเหตุการณ์ต่างๆ ได้กำหนดในแพ็กเกจ javax.microeddition.lcdui3

คลาส Screen เป็นคลาสแม่สำหรับส่วนประกอบ (Component) ต่างๆ ของ API ที่เชื่อมต่อกับผู้ใช้ทั้งหมด ส่วนประกอบต่างๆ ของการเชื่อมต่อกับผู้ใช้เหล่านี้ ได้แก่ Alert, From, List, TextBox และอื่นๆ

คลาส Canvas และคลาส Graphics เป็นคลาสหลักใน API ให้การสนับสนุนการเขียนภาพกราฟิกบนจอภาพของอุปกรณ์ไร้สายต่างๆ ของ GUI ใน API

นอกจากคลาสต่างๆ แล้ว เพื่อให้การสนับสนุนงานเกี่ยวข้องกับการจัดการจอภาพ การแสดงภาพ และการสร้างภาพกราฟิกต่างๆ บนจอภาพของอุปกรณ์ไร้สาย จึงได้มีการบรรจุการเชื่อมต่อต่างๆ ต่อไปนี้ลงในแพ็กเกจ ได้แก่

- javax.microedition.lcdui.**Choice**
- javax.microedition.lcdui.**CommandListener**
- javax.microedition.lcdui.**ItemStateListener**

3. แพ็กเกจสำหรับจัดการแหล่งจัดเก็บข้อมูลที่คงทน (Package for Persistent Storage Management)

โปรแกรมประยุกต์ต่างๆ ของ MIDP บางครั้งต้องการบรรจุข้อมูลอย่างถาวรในแหล่งจัดเก็บข้อมูลของ J2ME สำหรับใช้งานบนอุปกรณ์ไร้สายต่างๆ คลาสต่างๆ ที่กำหนดในแพ็กเกจ javax.microedition.rms

จะให้การสนับสนุนกลไกของแหล่งจัดเก็บข้อมูลของ J2ME ที่เรียกว่า RecordStore ซึ่งจะอนุญาตให้โปรแกรมประยุกต์สามารถเพิ่ม, ลบ และปรับปรุงข้อมูลในแหล่งจัดเก็บข้อมูลของ J2ME บนอุปกรณ์ไร้สายต่างๆ

นอกจากคลาสต่างๆ แล้ว เพื่อให้การสนับสนุนงานที่เกี่ยวกับการจัดเก็บข้อมูลในแหล่งจัดเก็บข้อมูล จึงได้มีการบรรจุการเชื่อมต่อต่างๆ ต่อไปนี้ลงในแพ็กเกจ ได้แก่

- javax.microedition.rms.**RecordComparator**
- javax.microedition.rms.**RecordEnumeration**
- javax.microedition.rms.**RecordFilter**
- javax.microedition.rms.**RecordListener**

4. แพ็กเกจสำหรับเครือข่าย (Package for Network)

โครงสร้างของการเชื่อมต่อทั่วไปใน CLDC ได้บรรจุกลุ่มของการเชื่อมต่อแบบต่างๆ แต่ CLDC จะไม่สนับสนุนโปรโตคอล (Protocol) ที่อยู่เบื้องหลังการเชื่อมต่อ

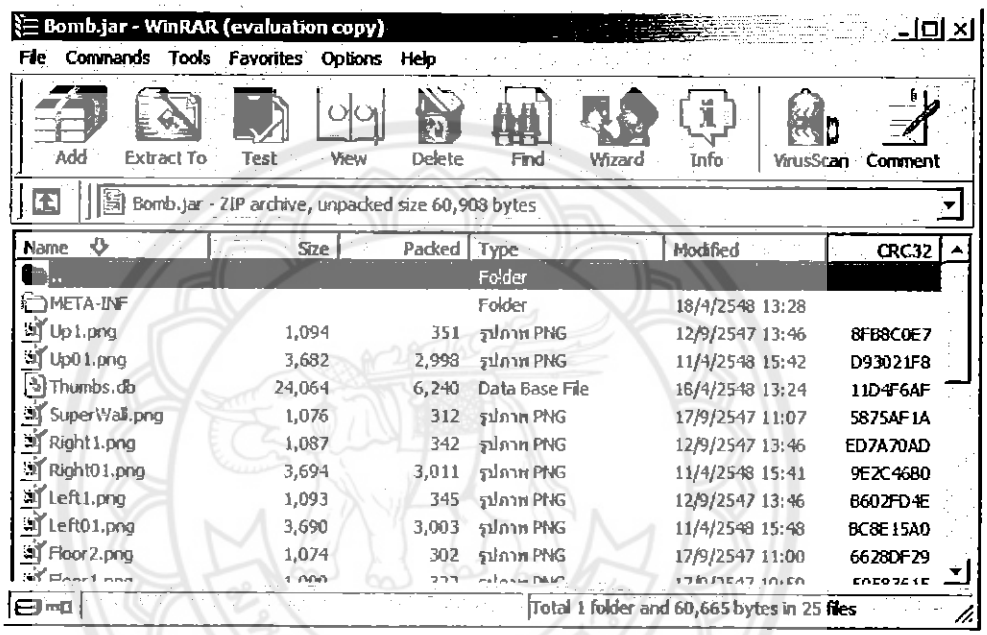
HTTPConnection มีลักษณะเป็นเครื่องมือของ MIDP ทั้งหมด เนื่องจากการสื่อสารของ HTTP สามารถดำเนินการบนอุปกรณ์ของ MIDP คลาสต่างๆ สำหรับการเชื่อมต่อแบบนี้จะพบในแพ็กเกจ javax.microedition.io

2.5 MIDlet คืออะไร

MIDlet แต่ละตัวที่พัฒนาขึ้นหรือดาวน์โหลดมาใช้งาน จะประกอบด้วยไฟล์ 2 ไฟล์ คือ ไฟล์นามสกุล jar กับไฟล์นาสกุล jad

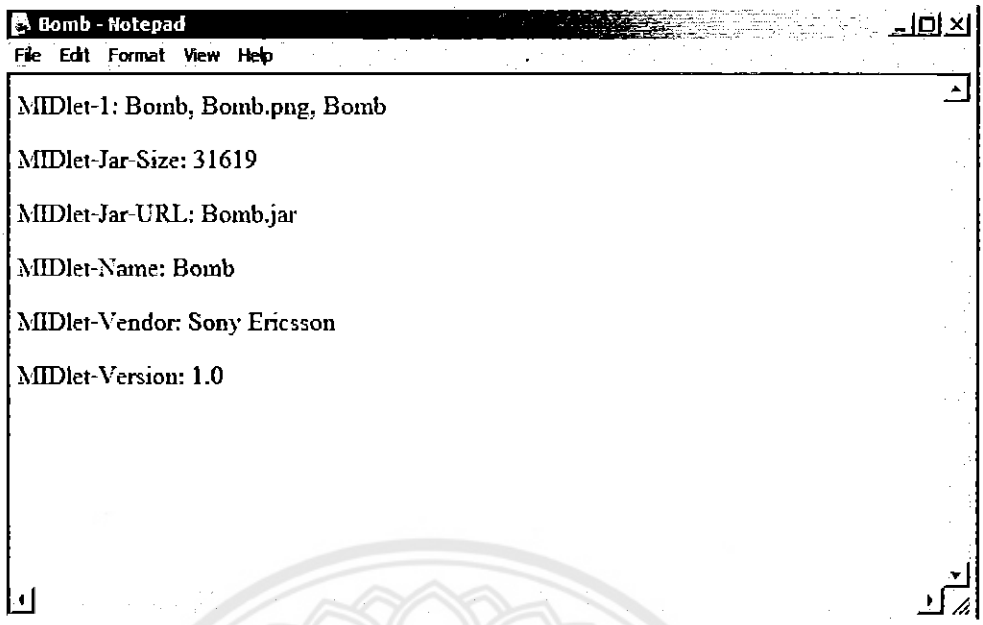
2.5.1 ความสำคัญของไฟล์ jar และ jad

ไฟล์นามสกุล jar (Java ARchive) เป็นที่เก็บคลาสไฟล์ต่างๆ ของ MIDlet นั้น โดยสามารถใช้โปรแกรมอย่าง WinRAR เพื่อเปิดดูคลาสไฟล์ที่อยู่ข้างในได้



รูปที่ 2.3 แสดงคลาสไฟล์ที่อยู่ในไฟล์ Bomb.jar

ส่วนไฟล์ jad เป็นไฟล์ข้อความธรรมดาที่ใช้บรรยายหรืออธิบายไฟล์ jar ว่ามีชื่ออะไร เวอร์ชันอะไร หรือมีขนาดไฟล์เท่าไหร่ สามารถใช้โปรแกรมอย่างเช่น Notepad เพื่อเปิดดูข้อมูลในไฟล์นี้ได้ ในการพัฒนา MIDlet จะต้องสร้างไฟล์ jar ขึ้นมาก่อน จากนั้นจึงเอาข้อมูลไฟล์ jar ไปสร้างไฟล์ jad และเมื่อจะเผยแพร่หรือแจกจ่าย MIDlet ก็จะต้องให้ไฟล์ทั้งสองควบคู่กันไป



รูปที่ 2.4 แสดงข้อมูลที่อยู่ในไฟล์ Bomb.jar

2.5.2 แหล่งดาวน์โหลด MIDlet

โทรศัพท์มือถือที่สนับสนุน จาวา มักมีตัวอย่าง MIDlet ติดตั้งมาให้สำหรับทดลองใช้งานอยู่แล้ว แต่ก็สามรถดาวน์โหลด MIDlet อื่นๆ มาติดตั้งเพิ่มเติมได้ สำหรับการนำ MIDlet ติดตั้งบนโทรศัพท์มือถือนั้น จะขึ้นอยู่กันว่าโทรศัพท์มือถือรุ่นดังกล่าวสนับสนุนวิธีการใดบ้างในการถ่ายเอาไฟล์ MIDlet โทรศัพท์ เช่น มีสายเคเบิลสำหรับเชื่อมต่อกับคอมพิวเตอร์ หรือจะใช้วิธีการดาวน์โหลด MIDlet ผ่าน WAP ที่เรียกว่า OTA (Over The Air) แหล่งดาวน์โหลด MIDlet ในอินเตอร์เน็ตก็มีหลายแห่งเช่น midlet.org สามารถดาวน์โหลดไฟล์ jar และ jad มาให้ทดลองใช้ได้

คุณลักษณะอย่างหนึ่งของ MIDlet ที่คิดว่าอาจเป็นตัวสร้างรายได้ให้แก่โอเปอเรเตอร์หรือผู้ให้บริการโทรศัพท์มือถือ และทำให้โทรศัพท์ที่สนับสนุน จาวา ได้รับความนิยมเพิ่มมากขึ้น คือ การดาวน์โหลด MIDlet เข้าสู่โทรศัพท์มือถือแบบ OTA ที่คล้ายคลึงกับการดาวน์โหลดโลโก้หรือเสียงเรียกเข้าที่ได้รับความนิยมอย่างแพร่หลาย

การหารายได้จาก MIDlet ของผู้ให้บริการโทรศัพท์มือถือก็คือ นำ MIDlet ไปวางไว้ในเครื่องแม่ข่ายเพื่อให้ผู้ใช้ดาวน์โหลด ซึ่งอาจมีการคิดค่าใช้จ่ายในการดาวน์โหลดแต่ละครั้งแล้วนำไปรวมอยู่ในบิลค่าโทรศัพท์ประจำเดือน โดยอาจถือว่าเป็นการใช้บริการเสริมอย่างหนึ่ง ส่วนรายได้จากการดาวน์โหลด MIDlet ก็แบ่งกันระหว่างผู้ให้บริการโทรศัพท์มือถือกับผู้พัฒนา MIDlet การดาวน์โหลด MIDlet แบบ OTA ในปัจจุบัน จะกระทำการผ่านโพรโตคอล WAP ดังนั้นโทรศัพท์มือถือที่ใช้จะต้องสนับสนุน WAP ด้วย

2.5.3 การพัฒนา MIDlet ทีละขั้น

จากที่ได้ทราบว่า MIDlet และ J2ME รวมถึงเครื่องมือที่ใช้พัฒนา MIDlet เพื่อให้เกิดความเข้าใจและเห็นความสัมพันธ์ในแต่ละขั้นตอนของการพัฒนา MIDlet มีดังนี้

1. Edit เขียนโปรแกรมจาวาด้วยเครื่องมือที่ใช้พัฒนาโปรแกรม (Text Editor)
2. Compile คอมไพล์หากไม่มีอะไรผิดพลาดก็จะได้คลาสไฟล์นามสกุล class ออกมา
3. Preverify ตรวจสอบคลาสไฟล์ที่ได้จากข้อ 2 ว่าถูกต้องตามข้อกำหนดหรือไม่ หากไม่มีอะไรผิดพลาดก็จะได้คลาสไฟล์นามสกุล class ออกมา
4. Run/Debug on emulated device นำคลาสไฟล์ที่ได้จากข้อ 3 ไปทดลองรันผ่าน Emulator หากมีข้อผิดพลาดหรือต้องการแก้ไขใดๆ ก็จะต้องย้อนกลับไปยังข้อ 1

เมื่อได้คลาสไฟล์ที่ทำงานถูกต้องตามความต้องการแล้วก่อนที่จะนำไปติดตั้งลงโทรศัพท์มือถือหรืออุปกรณ์ที่สนับสนุน J2ME เราต้องทำสิ่งที่เรียกว่า Packaging เสียก่อน ซึ่งเป็นการรวมเอาคลาสไฟล์และไฟล์อื่นๆที่ต้องใช้ในระหว่างการทำงานของ MIDlet ไปรวมอยู่ในไฟล์เดียวกัน ผลจากการทำ Packaging นี้จะได้ไฟล์นามสกุล jad และ jar สิ่งนี้เรียกว่า MIDlet Suite

2.5.4 การสร้าง MIDlet (Creating a MIDlet)

โปรแกรมประยุกต์บน MIDP เรียกว่า MIDlet ซึ่งมีส่วนคล้ายกับจาวา Applet แม้จะไม่มีวิธี main() ให้ใช้งาน แต่ MIDlet ก็ได้ใช้วิธี startApp(), pauseApp() และ destroyApp() จากคลาส java.microedtion.midlet.MIDlet

คลาส MIDlet ระบุวิธีที่สามารถเรียกใช้โดยชุดคำสั่งจัดการโปรแกรมประยุกต์ (AMS : Application Management Software) เพื่อสั่งให้โปรแกรม MIDlet เริ่มต้นและหยุดทำงาน

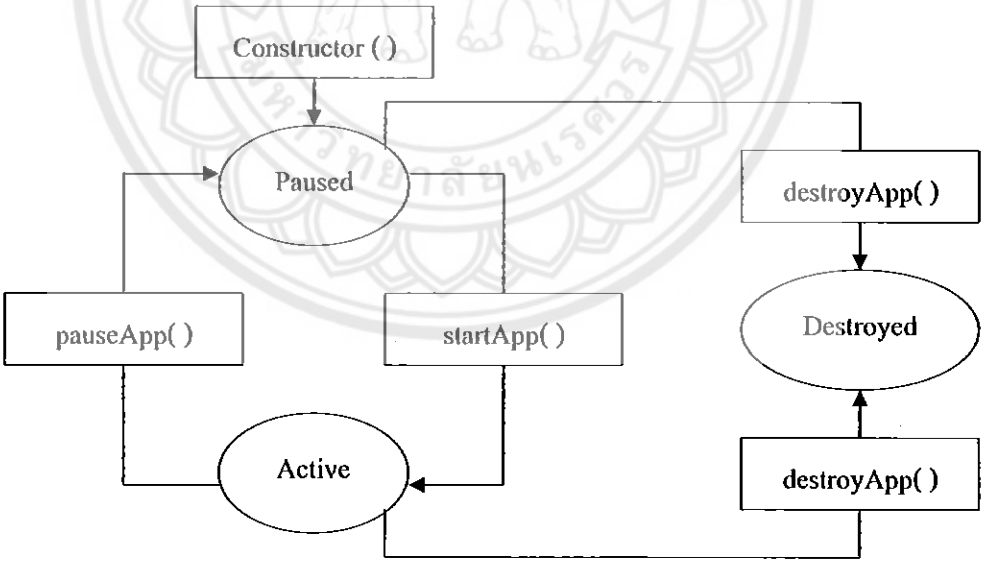
MIDlet จำเป็นต้องกำหนดตัวสร้าง (constructor) แบบสาธารณะ (public) ที่ไม่มีอาร์กิวเมนต์ (Argument) ใดๆ คำนิยามของคลาส javax.microedtion.midlet.MIDlet ประเภท abstract

2.5.5 วงจรการทำงานของ MIDlet

วงจรการทำงานของ MIDlet (MIDlet Life Cycle) การกระทำของ MIDlet ประกอบไปด้วย 3 สถานะ คือ กำลังทำงาน (Active), หยุดชั่วคราว (Pause) และถูกทำลาย (Destroy) โดยอาศัย AMS เป็นตัวควบคุมการเปลี่ยนสถานะหนึ่งไปยังอีกสถานะหนึ่งควบคุมด้วยวิธี startApp() pauseApp() และ destroyApp() ที่มาพร้อมกับ MIDlet รูปที่ 2.3 แสดงการส่งผ่านระหว่างสถานะทั้งสามผ่านการเรียกใช้วิธีเหล่านี้

จากรูปที่ 2.3 แสดงให้เห็นจุดเปลี่ยนระหว่างสถานะทั้ง 3 โดยการเรียกใช้วิธีข้างต้น เมื่อ MIDlet พร้อมที่จะประมวลผล เริ่มแรก AMS จะสร้างตัวอย่าง MIDlet ขึ้นมาก่อนโดยใช้ตัวสร้าง แบบสาธารณะที่ไม่มีอาร์กิวเมนต์ใดๆ เมื่อ MIDlet จะอยู่ในสถานะหยุดชั่วคราว จากนั้น AMS จะเรียกวิธี startApp() ขึ้นมาและ MIdlet จะเข้าสู่สถานะกำลังทำงานเปิดรับทรัพยากรต่างๆ ตามที่ต้องการและเริ่มต้นการทำงาน ในสถานะนี้ MIDlet จะทำงานและดึงทรัพยากรที่ต้องการไว้ใช้งาน

เมื่อ AMS ไม่ต้องการให้ MIDlet ทำงานต่อไปก็จะเรียกวิธี pauseApp() ทำให้ MIDlet หยุดทำงานและเข้าสู่สถานะหยุดชั่วคราว พร้อมทั้งคืนทรัพยากรที่ดึงมาใช้งานและเข้าสู่ภาวะไม่ทำงาน MIDlet สามารถกลับไปอยู่ที่สถานะกำลังทำงานได้อีกครั้งเมื่อชุดคำสั่งจัดการโปรแกรมประยุกต์เรียกวิธี startApp() ขึ้นมา



รูปที่ 2.5 แสดงวงจรการทำงานของ MIDlet

ท้ายที่สุดเมื่อ AMS ไม่ต้องการเรียกใช้งาน MIDlet หรือต้องการเตรียมพื้นที่ของหน่วยความจำสำหรับให้โปรแกรมอื่นได้ใช้งาน AMS จะส่งสัญญาณเตรียมทำลาย MIDlet ด้วยการเรียกวิธี destroyApp() และเข้าสู่สถานะถูกทำลาย ซึ่ง MIDlet จะปล่อยทรัพยากรทั้งหมดและบันทึกข้อมูลต่างๆ และหยุดการทำงานทั้งหมด หาก MIDlet อยู่ระหว่างการประมวลผลสำคัญอยู่ AMS สามารถร้องขอที่จะไม่เข้าสู่สถานะถูกทำลายได้โดยเรียกใช้ MIDletStateChangeException

อย่างไรก็ตาม AMS อาจปฏิเสธหรือยินยอมตามคำร้องขอนี้ก็ได้ ตัวแปร unconditional ประเภทบูลีนที่เป็นอาร์กิวเมนต์ของวิธี destroyApp() จะดำเนินการไม่ว่าการร้องขอนั้นจะได้รับการตอบสนองหรือไม่ ถ้าตัวบ่งชี้ unconditional มีค่าเท่ากับจริง (true) การร้องขอเป็นอันตกไป ตรงกันข้ามหากมีค่าเท่ากับเท็จ (false) ก็จะได้รับการตอบรับและจะเรียกวิธี destroyApp() ขึ้นมา

หาก MIDlet ต้องการเข้าสู่สถานะหยุดชั่วคราวหรือถูกทำลาย ก็สามารถทำได้โดยเรียกวิธี notifyPause() หรือ notifyDestroy() โดยวิธีทั้ง 2 จะแจ้งให้ AMS ทราบว่า MIDlet ได้เข้าสู่ สถานะหยุดชั่วคราวหรือถูกทำลายแล้ว ในกรณีนี้ AMS จะไม่เรียกใช้วิธี pauseApp() หรือ destroyApp() MIDlet จะเลิกดำเนินการต่างๆ แล้วแจ้งให้ AMS ทราบ

MIDlet จะเข้าสู่สถานะหยุดทำงานได้ก็ต่อเมื่อผ่านสถานะทำงานมาก่อนแล้ว แต่สามารถเข้าสู่สถานะถูกทำลายได้โดยตรงจากสถานะหยุดทำงานชั่วคราวและสถานะกำลังทำงาน นอกจากนี้ MIDlet ยังสามารถเข้าสู่สถานะหยุดการทำงานได้ในขณะที่กำลังทำงานหรือเมื่อได้รับคำสั่งจากAMS

2.6 เครื่องมือพื้นฐานสำหรับการพัฒนา J2ME

J2ME Wireless Toolkit หรือ WTK เป็นเครื่องมือพื้นฐานสำหรับพัฒนาโปรแกรม J2ME อย่างเป็นทางการของ บริษัท ซัน ไมโครซิสเต็ม สร้างขึ้นเพื่อให้นักพัฒนาสามารถเขียนโปรแกรมและทดสอบ MIDlet สะดวกมากขึ้น โดยจะทำการคอมไพล์โปรแกรมที่เขียนด้วยภาษาจาวา และสร้างออกมาเป็น MIDlet และยังมีตัวจำลองของโทรศัพท์มือถือ (Emulator) สำหรับทดสอบการทำงานของ MIDlet อีกด้วย

WTK มีทั้งเวอร์ชัน Windows และ UNIX ซึ่งสามารถดาวน์โหลดได้จากเว็บไซต์ของบริษัท ซัน ไมโครซิสเต็มส์ แต่ก่อนที่จะติดตั้ง WTK ในเครื่องจะต้องมี J2SDK ติดตั้งเอาไว้ด้วย

2.7 การพัฒนาโปรแกรมบนโทรศัพท์มือถือ

ส่วนติดต่อกับผู้ใช้ (User Interface : UI) คือ ส่วนที่ผู้ใช้โทรศัพท์มือถือจะสามารถมองเห็นได้จากหน้าจอ มีหน้าที่ติดต่อรับคำสั่งจากผู้ใช้ รวมทั้งแสดงผลการทำงานต่างๆ ออกมาให้ผู้ใช้เห็น การสร้างส่วนติดต่อกับผู้ใช้สามารถสร้างขึ้นมาได้ 2 วิธีใหญ่ๆ ทั้งการเรียกใช้สิ่งที่มีมาให้สำเร็จรูปและการสร้างขึ้นมาเองทั้งหมด

J2ME สามารถทำงานได้บนหลากหลายอุปกรณ์ โดยที่อุปกรณ์เหล่านี้มีทรัพยากรและความสามารถในการแสดงผลที่แตกต่างกันมาก ดังนั้นทางบริษัท ซัน ไมโครซิสเต็มส์ จึงไม่ได้กำหนดคลาสสมบูรณ์แบบให้ J2ME สำหรับติดต่อผู้ใช้ แต่กำหนด APIs ของส่วนติดต่อผู้ใช้ในลักษณะชั้นโครงแบบ (Profile) ของกลุ่มอุปกรณ์ประเภทต่างๆ แทน

APIs ที่ใช้สร้างส่วนติดต่อผู้ใช้ของ MIDP จะถูกบรรจุอยู่ในชุดคำสั่ง (Package) javax.microedition.lcdui โดย APIs นี้ถูกสร้างขึ้นมาใหม่ทั้งหมดเพื่อให้เหมาะสมกับการนำมาใช้ในอุปกรณ์ขนาดเล็กที่มีทรัพยากรจำกัด (ไม่ใช้หน่วยความจำและหน่วยประมวลผลมากเกินไป) เหตุที่ไม่สามารถนำ Abstract Windows Toolkit (AWT) และ Swing Component ใน J2SE มาใช้ได้ เนื่องจากมีขนาดใหญ่และใช้ทรัพยากรในการทำงานมากและมีความซับซ้อนเกินกว่าจะใช้ในอุปกรณ์ขนาดเล็กได้

คลาสใน javax.microedition.lcdui มีคลาสหลักๆ อยู่ทั้งหมด 9 คลาส ซึ่งแต่ละคลาสมีหน้าที่และความสัมพันธ์กัน ดังนี้

AlertType	แสดงประเภทของคลาส Alert ที่ใช้แจ้งเตือนหรือข้อผิดพลาด
Command	สร้างเมนูให้แก่ผู้ใช้ตามคำสั่งที่กำหนดไว้
Display	จัดการแสดงผลหน้าจอของอุปกรณ์
Displayable	เก็บข้อมูลที่ต้องการแสดงผล
Font	คลาสตัวอักษร
Graphics	ใช้งานเกี่ยวกับกราฟิก
Image	ใช้เก็บข้อมูลที่เป็นรูปภาพ
Item	ซูเปอร์คลาสของส่วนประกอบที่สามารถใช้ร่วมกับ Form ได้
Ticker	ใช้ในการแสดงข้อความที่สามารถเคลื่อนไหวในแนวนอนได้

2.7.1 ขั้นตอนการสร้างส่วนติดต่อกับผู้ใช้

ขั้นแรก : สร้างวัตถุ (Object) Display

วัตถุของคลาส Display จะเป็นตัวนำวัตถุ Displayable ออกไปแสดงผลทางหน้าจอของอุปกรณ์ โดยใน MIDlet หนึ่งๆ จะมีวัตถุของคลาส Display ได้เพียงวัตถุเดียวเท่านั้น การสร้างวัตถุ Display ใน MIDlet สามารถทำได้ดังนี้

```
Display myDisplay;
startApp(){
        .....
        myDisplay = Display.getDisplay(this);
}
```

1. ประกาศตัวแปรให้มีชนิดเป็นคลาส Display โดยในที่นี้ตั้งชื่อว่า myDisplay
2. สร้างวัตถุของคลาส Display ด้วยวิธี (Method) getDisplay(MIDlet c) ของคลาส Display โดยส่งค่า (Parameter) c เป็นคำสำคัญ (Keyword) this เพื่อแทน MIDlet ที่กำลังทำงานอยู่

ขั้นที่ 2 : เตรียมข้อมูลที่จะแสดงด้วยคลาส Displayable

วัตถุของคลาส Displayable จะเป็นตัวเก็บข้อมูลที่ต้องการแสดงเพื่อให้วัตถุ Display นำข้อมูลไปแสดงผลต่อไป ซึ่งสามารถสร้างวัตถุขึ้นมาแทนสิ่งที่ต้องการแสดงไปที่หน้าจอมือถือ ซึ่งวัตถุเหล่านี้จะเป็นวัตถุของคลาสย่อยที่สืบทอดคุณสมบัติมาจากคลาส Displayable อีกทีหนึ่ง

คลาส Displayable ประกอบไปด้วยคลาสย่อย 3 กลุ่ม ดังนี้

1. Screen ที่กำหนดไว้ก่อนแล้ว (Alert, List และ TextBox) : โปรแกรมประยุกต์ (Application) ไม่สามารถเพิ่มส่วนประกอบอื่นๆ ในวัตถุของคลาสนี้ได้
2. Screen ทั่วไป (Form) : โปรแกรมประยุกต์สามารถเพิ่มส่วนประกอบอื่นๆ เช่น ข้อความ รูปภาพ ลงในวัตถุของคลาสนี้ได้
3. Canvas : โปรแกรมประยุกต์สามารถควบคุมการแสดงผลและเข้าจัดการเหตุการณ์ (Event) ของส่วนประกอบได้อย่างเต็มที่

คลาส Screen นั้น ใช้ในการสร้าง High-Level APIs ส่วนคลาส Canvas นั้น ใช้ในการสร้าง Low-Level APIs

ในโปรแกรมประยุกต์หนึ่งๆ สามารถมีได้ Screen แต่จะมีเพียง Screen เดียวที่ถูกแสดงบนหน้าจอและผู้ใช้ก็สามารถติดต่อกับส่วนต่างๆ ของ Screen นี้ได้เพียง Screen เดียวเท่านั้น โดยจะเรียก Screen ที่กล่าวถึงนี้ว่า Screen ปัจจุบัน (Current Screen)

ขั้นที่ 3 : สั่งให้แสดงผล

หลังจากเตรียมข้อมูลที่จะใช้แสดงผลโดยการสร้างวัตถุของคลาส Displayable เรียบร้อยร้อย ซึ่งอาจจะเป็น รูปแบบ (Form), กล่องข้อความ (TextBox) หรือคลาสย่อยอื่นๆ ที่สืบทอดคุณสมบัติมาจากคลาส Displayable ถ้าจะสั่งให้แสดงข้อมูลเหล่านั้นด้วยวิธี setCurrent ของคลาส Display โดยอยากแสดง Screen ใด ก็ให้ใส่วัตถุของ Screen นั้นเป็นค่าตัวเลข ดังตัวอย่าง

```
Display myDisplay;
TextBox textbox;
textbox = new TextBox("Example", "Hello, My name is Thawatchai.",20,
TextField.ANY);
.....
myDisplay = Display.getDisplay(this);
myDisplay.setCurrent(textbox);
```

ส่วนของโปรแกรมข้างต้นเป็นตัวอย่างการนำกล่องข้อความไปแสดงบนหน้าจอ โดยมีขั้นตอนการทำงาน ดังนี้

1. ประกาศตัวแปรชนิด Display โดยในที่นี้มีชื่อว่า myDisplay
2. ประกาศตัวแปรชนิด TextBox โดยในที่นี้มีชื่อว่า textbox
3. สร้างวัตถุ TextBox ให้แสดงข้อความว่า "Hello, My name is Thawatchai."
4. สร้างวัตถุของคลาส Display
5. ส่งวัตถุ TextBox ไปให้ Display เพื่อแสดงผลออกทางหน้าจอโดยใช้วิธี setCurrent()

ขั้นที่ 4 : เตรียมตอบสนองต่อเหตุการณ์ที่จะเกิดขึ้น

เมื่อสร้างส่วนติดต่อผู้ใช้ซึ่งอาจจะประกอบด้วยช่องรับข้อความหรือปุ่มต่างๆ ต้องเตรียมรับมือกับการทำงานของผู้ใช้อาจจะมีการตอบโต้กับ MIDlet เช่น กดปุ่มพิมพ์ข้อมูล หรืออะไรก็ตามที่จะต้องตอบสนองต่อการกระทำเหล่านั้นให้เหมาะสม

การตอบสนองต่อเหตุการณ์ใน J2ME แบ่งออกเป็น 2 ประเภทคือ ระดับสูง (High-Level) และระดับล่าง (Low-Level) โดยระดับสูงจะเป็นการตอบสนองต่อเหตุการณ์ที่มีการเลือกคำสั่งจากเมนูหรือปุ่มที่สร้างขึ้นมาในโปรแกรมรวมทั้งเหตุการณ์ที่ Item ต่างๆ ในฟอร์มมีการเปลี่ยนแปลง ส่วนระดับล่างจะเป็นการตอบสนองกับเหตุการณ์ที่ผู้ใช้กดปุ่มบนโทรศัพท์มือถือโดยตรงซึ่งนิยมใช้ในการควบคุมเกม

ถ้ามีเหตุการณ์เกิดขึ้น จะรู้ได้ก็ต่อเมื่อได้สร้างสิ่งที่เรียกว่า ตัวดักจับเหตุการณ์ (Event Listener) ขึ้นมา ซึ่งจะทำให้มีการเรียกวิธีบางวิธีขึ้นมาทำงานเพื่อตอบสนองกับเหตุการณ์นั้นๆ โดยจะต้องเขียนวิธีดังกล่าวขึ้นมาและสั่งให้โปรแกรมทำงานที่ต้องการตามเหตุการณ์ที่เกิดขึ้นเอง

การจัดการกับเหตุการณ์ที่เกิดขึ้นประกอบด้วยขั้นตอนต่างๆ ดังนี้

1. สร้างส่วนติดต่อผู้ใช้ Event Listener ในที่นี้เป็นการสร้าง CommandListener
2. สร้างคำสั่งให้ผู้ใช้ผ่านทางวัตถุ Command โดยในที่นี้ใช้ชื่อ exitCommand ค่าตัวแรกแทนชื่อปุ่มที่แสดงออกมาที่หน้าจอ ค่าตัวที่ 2 ใช้กำหนดประเภทของปุ่ม ส่วนค่าตัวที่ 3 ใช้กำหนดความสำคัญของปุ่ม เมื่อนำไปเทียบกับปุ่มอื่นๆ (ถ้ามี) โดยค่ายิ่งน้อยจะยิ่งสำคัญ
3. ผูกวัตถุ Command กับส่วนที่ติดต่อผู้ใช้ เช่น รูปแบบ โดยใช้วิธี addCommandListener()
4. แนะนำวัตถุ CommandListener ให้แก่ส่วนที่ติดต่อผู้ใช้ เช่น Screen โดยใช้วิธี setCommandListener()
5. วัตถุ CommandListener ดักจับเหตุการณ์ exitCommand และตอบสนองต่อเหตุการณ์ซึ่งในที่นี้ก็คือออกจาก MIDlet

2.7.2 ส่วนติดต่อกับผู้ใช้ (Graphical User Interface : GUI) ของ MIDP

ส่วนติดต่อกับผู้ใช้หรือ GUI ของ MIDP คือ ส่วนที่ติดต่อกับผู้ใช้ ไม่ว่าจะเป็นการแสดงผล การรับข้อมูลหรือหารตรวจสอบเหตุการณ์ต่างๆ ของอุปกรณ์ที่กำลังรัน MIDlet ซึ่งมีอยู่ 2 แบบ คือ High-Level APIs และ Low-Level APIs

High-Level APIs : ดูแลส่วนติดต่อผู้ใช้ระดับสูง

High-Level APIs เป็นส่วนติดต่อผู้ใช้สำหรับการรับและแสดงผลข้อมูล เช่น กรอบรับข้อความ (TextBox), รายการตัวเลือก (List หรือ ChoiceGroup) เป็นต้น ซึ่ง High-Level APIs มีลักษณะคล้ายกับ Control ของ Visual Basic สามารถเลือกมาสร้างวัตถุใช้งานใน MIDlet ได้ทันที โดยคลาสของ High-Level APIs นี้ เป็นคลาสที่สืบทอดมาจากคลาส Screen

ข้อเสียของการใช้ High-Level APIs คือ ไม่สามารถควบคุมหรือเปลี่ยนแปลงลักษณะบางอย่างของวัตถุได้ เช่น ขนาด, ตำแหน่งการแสดงผล, สีหรือรูปแบบตัวอักษร เป็นต้น แต่มีข้อดีตรงที่ จะไม่ต้องกังวลในเรื่องการแสดงผลออกมาของอุปกรณ์ที่ต่างกัน เพราะ High-Level APIs มีตัวจัดการเรื่องการแสดงผลเอาไว้โดยเฉพาะ แนวคิดของ High-Level APIs คือ สามารถสร้างและใช้ส่วนติดต่อผู้ใช้ (ที่เตรียมไว้ให้) อย่างอิสระโดยไม่ขึ้นกับอุปกรณ์หรือคุณสมบัติของอุปกรณ์ที่จะรัน MIDlet

Low-Level APIs : จัดการกราฟิกระดับต่ำ

Low-Level APIs ถูกออกแบบมาสำหรับ MIDlet ที่ต้องการแสดงหรือจัดการเกี่ยวกับกราฟิกโดยตรง เช่น แสดงข้อความ, วาดเส้นตรง, วงกลม หรือแสดงรูปกราฟิก รวมถึงการตรวจสอบเหตุการณ์ต่างๆ ในระดับต่ำ เช่น สถานะการกดปุ่ม เป็นต้น ซึ่งจะเห็นว่า Low-Level APIs นั้นเหมาะสำหรับใช้สร้าง MIDlet ที่เป็นเกมกราฟิก โดยคลาสของ Low-Level APIs สืบทอดมาจากคลาส Canvas

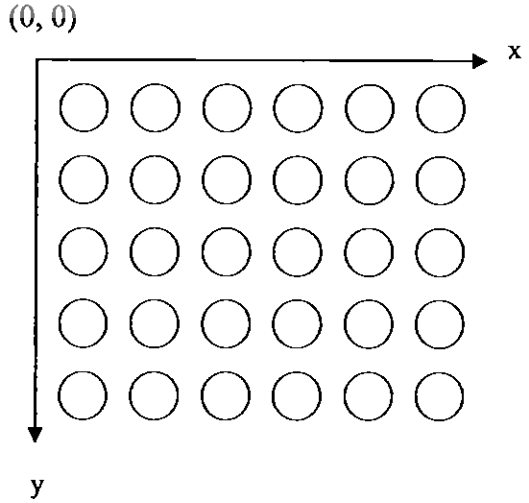
สำหรับการใช้ Low-Level APIs หากนำ MIDlet ที่ได้ ไปรันในอุปกรณ์ต่างกัน ก็อาจจะได้ผลการทำงานที่ต่างกันหรืออาจใช้งานไม่ได้เลย ตัวอย่างที่เห็นได้ชัดคือ ขนาดหน้าจอแสดงผลของอุปกรณ์ที่สนับสนุน J2ME แต่ละรุ่น มักจะมีขนาดไม่เท่ากัน หากเขียน MIDlet โดยกำหนดพื้นที่การแสดงผลกราฟิกตายตัวไว้ที่ขนาด 100x90 พิกเซล แล้วนำ MIDlet ดังกล่าวไปรันบนอุปกรณ์ที่มีขนาดจอภาพเล็กกว่า ก็จะต้องเกิดปัญหากับการแสดงผลอย่างแน่นอน หรือ MIDlet บางตัวที่กำหนดให้ใช้ปุ่มหมายเลข 1-9 ในการทำงาน หากนำ MIDlet นั้นไปรันในเครื่องปาล์ม ก็จะใช้งานไม่ได้ เพราะไม่มีปุ่มให้กดนั่นเอง

2.7.3 การวาดรูปโดยใช้ MIDlet แบบกราฟิก

การเขียน MIDlet แบบกราฟิกนั้นเป็นแบบ Low-Level APIs คลาสที่จะใช้คำสั่งเกี่ยวกับการวาดหรือสร้างรูปกราฟิกได้ จะต้องเป็นคลาสที่ขยายมาจากคลาส Canvas นอกจากจะมีคลาส Graphics MIDlet ซึ่งเป็นคลาสหลักแล้ว ยังมีคลาสชื่อ DrawCanvas() ด้วย ซึ่งคลาสนี้ขยายมาจาก Canvas เพื่อใช้สำหรับวาดและแสดงรูปกราฟิกของ MIDlet ออกมา โดยคลาสนี้จะถูกเรียกหรือสร้างเป็นวัตถุผ่านคลาสหลักอีกทีหนึ่ง และคำสั่งสร้างหรือวาดรูปกราฟิกจะต้องเขียนไว้ในวิธี paint() เท่านั้น

ระบบพิกัดร่วม (Co-ordinate) ของกราฟิก

ระบบพิกัดร่วมหรือพิกัด (x, y) ของวัตถุกราฟิกใน J2ME จะเริ่มจากมุมบนซ้ายนับเป็นตำแหน่งที่ (0, 0) ค่า x จะเพิ่มขึ้นจากทางด้านซ้ายไปขวา และค่า y จะเพิ่มขึ้นจากบนลงล่าง



รูปที่ 2.6 ระบบพิกัดร่วมของกราฟิก

การกำหนดสี

กราฟิก จะสนับสนุนการใช้สีในระดับ 24 บิต คือ สีแดง (R), สีเขียว (G) และสีน้ำเงิน (B) อย่างละ 8 บิต แต่ในการใช้งานจริงจะต้องคำนึงถึงอุปกรณ์ที่รัน MIDlet ด้วย ว่าสามารถแสดงสีได้ในระดับไหน สำหรับคำสั่งที่ใช้กำหนดสีก่อนการสร้างหรือวาดรูปกราฟิกก็คือ setColor()

คำสั่งนี้สามารถระบุค่าสีได้ 2 แบบคือ ระบุแม่สี (RGB) เป็นเลขฐาน 16 เพียงค่าเดียว หรือจะระบุแยกค่าแม่สีแต่ละสี (R, G, B) ก็ได้ ตัวอย่างเช่น

```
g.setColor(255, 255, 255);
```

เป็นการกำหนดสีให้เป็นสีขาว ซึ่งอาจเขียนได้อีกแบบคือ

```
g.setColor(0xFFFFFF);
```

สำหรับค่าสีที่ต้องใช้งานบ่อยๆ ได้แสดงไว้ในตารางที่ 2.1

คำสั่งที่ใช้วาดรูปและคำสั่งที่เกี่ยวข้องของกราฟิก มีหลายคำสั่ง แต่ที่นำมาใช้บ่อยๆ ใน MIDlet คือ การวาดเส้นตรง, รูปสี่เหลี่ยม, ส่วนของวงกลมหรือวงรี และการแสดงข้อความ

ตารางที่ 2.1 ค่าสีที่มีโอกาสใช้งาน

สี	แดง (R)	เขียว (G)	น้ำเงิน (B)	RGB (ฐาน 16)
ขาว	255	255	255	0xFFFFFF
ดำ	0	0	0	0x000000
แดง	255	0	0	0xFF0000
เขียว	0	255	0	0x00FF00
น้ำเงิน	0	0	255	0x0000FF
เทาสว่าง	192	192	191	0xC0C0C0
เทาเข้ม	128	128	128	0x8088
ม่วง	255	0	255	0xFF00FF
เหลือง	255	255	0	0xFFFF00

การวาดเส้นตรง

คำสั่งหรือวิธีที่ใช้วาดเส้นตรง คือ drawLine() มีโครงสร้างดังนี้

```
drawLine(int x1, int y1, int x2, int y2)
```

x1, x2 คือ พิกัดเริ่มต้นของเส้นตรง

y1, y2 คือ พิกัดสุดท้ายของเส้นตรง

ปร.
ช395ก
2547
i5000683

การวาดรูปสี่เหลี่ยมโปร่ง

คำสั่งหรือวิธีที่ใช้วาดเส้นตรง คือ drawRect() มีโครงสร้างดังนี้

drawRect(int x1, int y1, int width, int height)

x1, y1 คือ พิกัดมุมบนซ้ายของรูปสี่เหลี่ยม

width คือ ความกว้างของรูปสี่เหลี่ยม (พิกเซล)

height คือ ความสูงของสี่เหลี่ยม (พิกเซล)

การวาดรูดสี่เหลี่ยมทึบ

คำสั่งหรือวิธีที่ใช้วาดเส้นตรง คือ fillRect() มีโครงสร้างดังนี้

fillRect(int x1, int y1, int width, int height)

x1, y1 คือ พิกัดมุมบนซ้ายของรูปสี่เหลี่ยม

width คือ ความกว้างของรูปสี่เหลี่ยม (พิกเซล)

height คือ ความสูงของสี่เหลี่ยม (พิกเซล)

ส่วนคำสั่งที่ใช้วาดรูปสี่เหลี่ยมทึบขอบมน คือ fillRoundRect() มีโครงสร้างดังนี้

fillRoundRect(int x1, int y1, int width, int height, int arcWidth, int atcHeight)

x1, y1 คือ พิกัดมุมบนซ้ายของรูปสี่เหลี่ยม

width คือ ความกว้างของรูปสี่เหลี่ยม (พิกเซล)

height คือ ความสูงของสี่เหลี่ยม (พิกเซล)

arcWidth คือ ความกว้างของส่วนโค้งบริเวณมุมของสี่เหลี่ยม

arcHeight คือ ความสูงของส่วนโค้งบริเวณมุมของสี่เหลี่ยม

การวาดส่วนของวงกลมหรือวงรี

คำสั่งที่ใช้วาดบางส่วนของวงกลมหรือวงรีคือ drawArc() มีโครงสร้างดังนี้

fillRect(int x1, int y1, int width, int height, int startAngle, int arcAngle)

x1, y1 คือ พิกัดที่มุมบนซ้ายของวงรี

width คือ ความกว้างในแนวแกน x

height คือ ความกว้างในแนวแกน y

startAngle คือ มุมเริ่มต้น

arcAngle คือ จำนวนองศา (นับจากมุมเริ่มต้น)

การกำหนดรูปแบบตัวอักษรและแสดงข้อความ

การกำหนดรูปแบบอักษรทำได้โดยใช้วิธี getFont() เพื่อเก็บรูปแบบตัวอักษรที่ต้องการไว้ในตัวแปร ขั้นต่อไปคือการนำค่าตัวอักษรจากตัวแปรมากำหนดการใช้งานด้วยวิธี setFont() ข้อความที่แสดงออกมาจึงจะมีรูปแบบตัวอักษรตามที่ต้องการ หากไม่มีการกำหนดหรือเลือกใช้รูปแบบตัวอักษรแล้ว กราฟิก ก็จะใช้รูปแบบตัวอักษรมาตรฐานของระบบในการแสดงข้อความแทน

โครงสร้างคำสั่ง getFont() เป็นดังนี้

getFont(int face, int style, int size)

face คือ ลักษณะของตัวอักษร ซึ่งเลือกได้ 3 แบบ ได้แก่ Font.FACE_SYSTEM, Font.FACE_MONOSPACE หรือ Font.FACE_PROPORTIONAL

style คือ รูปแบบการแสดงตัวอักษร ซึ่งเลือกได้ 4 แบบ ได้แก่ Font.STYLE_PLAIN, Font.STYLE_BOLD, Font.STYLE_ITALIC หรือ Font.STYLE_LARGE

size คือ ขนาดของตัวอักษร ซึ่งเลือกได้ 3 แบบ ได้แก่ Font.SIZE_SMALL, Font.SIZE_MEDIUM, Font.SIZE_LARGE

2.7.4 การตรวจสอบเหตุการณ์

สำหรับการสร้าง MIDlet ที่ติดต่อกับ Canvas (ใช้หรือขยายจากคลาส Canvas) หรือ Low-Level APIs อย่างเกมกราฟิกนั้น สามารถตรวจสอบเหตุการณ์การทำงานในระดับต่ำ เช่น การกดปุ่ม, ปล่อยปุ่ม , การแตะหรือสัมผัสของตัวชี้ (Pointer) วิธีที่สามารถใช้งานกับการตรวจสอบเหตุการในระดับต่ำ มีดังนี้

- protected void keyPressed(int keyCode)
- protected void keyReleased(int keyCode)
- protected void keyRepeated(int keyCode)
- protected void pointerPressed(int x, int y)
- protected void pointerDragged(int x, int y)
- protected void pointerReleased(int x, int y)
- protected void showNotify()
- protected void hideNotify()
- protected abstract void paint(Graphics g)
- commandAction() method of the CommandListener interface

หากต้องการตรวจสอบเหตุการณ์ใด ก็เขียนดักจับเหตุการณ์นั้นเอาไว้ในคลาส แล้วเขียนคำสั่งที่ต้องการให้ทำงานเมื่อเกิดเหตุการณ์ดังกล่าว

ตารางที่ 2.2 ค่าของปุ่มกดต่างๆ ในตัวจำลองของ WTK

ปุ่มกด	keyCode		getGameAction	
	ค่าตัวเลข	ชื่อของปุ่ม	ค่าตัวเลข	ชื่อของปุ่ม
บน	-1	-	1	UP
ล่าง	-2	-	6	DOWN
ซ้าย	-3	-	2	LEFT
ขวา	-4	-	5	RIGHT
กลาง	-5	-	8	FIRE
0	48	KEY_NUM0	0	-
1	49	KEY_NUM1	0	-
2	50	KEY_NUM2	0	-
3	51	KEY_NUM3	0	-
4	52	KEY_NUM4	0	-
5	53	KEY_NUM5	0	-
6	54	KEY_NUM6	0	-
7	55	KEY_NUM7	9	GAME_A
8	56	KEY_NUM8	0	-
9	57	KEY_NUM9	10	GAME_B
*	42	KEY_STAR	11	GAME_C
#	35	KEY_POUND	12	GAME_D

ค่าตัวเลข keyCode และ getGameAction() ของปุ่มต่างๆ ในตัวจำลองของ WTK แสดงดังตารางที่ 2.2

การตรวจสอบเหตุการณ์ไม่ว่าจะเป็นการกดปุ่ม (keyPressed) หรือปล่อยจากการกดปุ่ม (keyReleased) ก็ตาม จะตรวจสอบได้จากสิ่งที่เรียกว่า keyCode เมื่อผู้ใช้กดหรือปล่อยและได้เขียนดักจับเหตุการณ์เอาไว้ ก็จะได้ค่าตัวเลข keyCode ซึ่งบอกให้ทราบว่าผู้ใช้กดปุ่มหรือปล่อยปุ่มใดอยู่ เพราะแต่ละปุ่มหรือแต่ละปุ่มจะมีค่า keyCode ไม่ซ้ำกัน และ MIDP ยังได้แบ่งปุ่มออกมาอีกกลุ่มหนึ่งเรียกว่า Game Action (ได้แก่ ปุ่ม Arrow Key ทั้งสี่และปุ่มอื่นๆ อีกบางส่วน) ถ้าจะตรวจสอบว่าปุ่มในกลุ่ม Game Action นี้ถูกกดหรือไม่ ก็เพียงแต่นำค่า keyCode ไปหาด้วยคำสั่ง getGameAction()

การตรวจสอบปุ่มที่กดนั้น อาจเขียนโค้ดตรวจสอบจากค่าตัวเลขประจำปุ่มกด หรือใช้ชื่อของปุ่ม เช่น KEY_NUM0, KEY_STAR, UP หรือ DOWN ก็ได้ (ชื่อของปุ่มกดต่างๆ นี้ กำหนดเอาไว้ในคลาส Canvas) ยกตัวอย่างเช่น การตรวจสอบการกดปุ่มหมายเลข 0 จะเขียนโค้ดได้ดังนี้

```
if (keyCode == 48)
```

หรือ `if (keyCode == KEY_NUM0)`

ในที่นี้ควรจะเขียนโค้ดเป็นแบบตัวหลัง เพราะจะทำให้การตรวจสอบและแก้ไขโค้ดทำได้ง่ายขึ้น เนื่องจากชื่อของปุ่มกดช่วยสื่อความหมายอยู่แล้ว และในทางเทคนิคของการเขียนโปรแกรมยิ่งจำเป็นต้องใช้ชื่อปุ่มกดแทนการใช้ตัวเลข เพราะในอุปกรณ์บางอย่างนั้น ค่าตัวเลขของบางปุ่มกดจะไม่ตรงกับปุ่มกดในตัวจำลองของ WTK

2.7.5 รูปกราฟิกใน MIDlet แบบกราฟิก

การนำรูปกราฟิกที่มีนามสกุล png มาใช้กับ MIDlet ที่เขียนหรือจัดการกับรูปกราฟิกผ่าน Canvas มีอยู่ 2 แบบ คือ แบบแรกใช้วิธีโหลดรูปกราฟิกจากไฟล์ png เข้ามาแสดงโดยตรง (ไฟล์ png จะถูกดึงเข้าไปอยู่มนไฟล์ jar ของ MIDlet Suite) และอีกแบบคือ ใช้วิธีฝังข้อมูลของรูปกราฟิก png ไว้ในไฟล์ .class

การแสดงรูปกราฟิกแบบแรกเป็นวิธีที่นิยมใช้กัน เพราะมีขั้นตอนที่ไม่ซับซ้อน ในเกมบอมเบอร์แมนก็เป็นการสร้าง MIDlet ที่แสดงรูปออกมาทางจอภาพ ผู้ใช้สามารถเลื่อนตำแหน่งรูปไปทางด้านบน-ล่าง, ซ้าย-ขวา ปุ่มกดที่ใช้สำหรับเลื่อนรูปนั้นมี 2 แบบ คือ ใช้ Arrow Key หรือใช้ปุ่มหมายเลข 2, 4, 6, 8 และ 5 ในที่นี้จะใช้ปุ่มที่เป็นตัวเลข เนื่องจากโทรศัพท์มือถือรุ่นใหม่ที่สนับสนุน J2ME บางรุ่นนั้น ได้วางตำแหน่ง Arrow Key ไม่เอื้ออำนวยต่อการใช้งาน ทำให้การเลื่อนรูปทำได้ลำบากและบางรุ่นไม่มีปุ่ม Select (ปุ่มกลางที่อยู่ระหว่าง Arrow Key) โดยการเลื่อนรูปไปยังตำแหน่งบน-ล่าง, ซ้าย-ขวา ทำได้โดยการกดปุ่มหมายเลข 2, 8, 4, 6 ได้ และ 5 ก็กำหนดให้ทำงานเหมือนกับปุ่ม Select

ตัวอย่างการวาดรูปออกทางจอภาพบนโทรศัพท์มือถือ

```
g.drawImage(imageData[1], x, y, 0)
```

ในการแสดงรูปกราฟิกออกทางจอภาพด้วยวิธี drawImage() โดยจะต้องระบุค่าพารามิเตอร์ 4 ตัว คือ

1. ตัวแปรที่เก็บรูปกราฟิก
2. พิกัดในแกน x
3. พิกัดในแกน y
4. ตำแหน่งการแสดงผลของรูปกราฟิก

2.8 การเขียนโปรแกรมเครือข่ายบนโทรศัพท์มือถือ โดยการเชื่อมต่อแบบซ็อกเก็ต

ซ็อกเก็ตเป็นการติดต่อสื่อสารระหว่างโปรแกรมที่ทำงานบนเครือข่าย เช่น อุปกรณ์ไร้สายหรือโทรศัพท์มือถือกับเครื่องแม่ข่ายระยะไกล หรือระหว่างโทรศัพท์มือถือ 2 เครื่อง

ซ็อกเก็ตช่วยนักพัฒนา J2ME สามารถพัฒนาโปรแกรมประยุกต์ทุกชนิดเพื่อใช้งานบนอุปกรณ์ไร้สายได้ อย่างไรก็ดีการเชื่อมต่อแบบซ็อกเก็ตก็ยังมีข้อจำกัดในการบรรจุลงในอุปกรณ์ไร้สาย และไม่มีความสะดวกในการเชื่อมโยงไปยังเครือข่ายไร้สายที่แตกต่างกัน

การนำซ็อกเก็ตมาใช้ ก่อนอื่นผู้รับผู้ส่งจะต้องสร้างการเชื่อมต่อระหว่างซ็อกเก็ตก่อน โดยฝ่ายหนึ่งจะคอยฟังคำร้องขอเชื่อมต่อ ขณะที่อีกฝ่ายหนึ่งขอเชื่อมต่อเครือข่าย เมื่อซ็อกเก็ตทั้ง 2 ด้านเชื่อมต่อเข้าด้วยกัน ก็จะสามารถส่งผ่านข้อมูลไปยังอีกด้านหนึ่งได้

การรับข้อมูลจากเครื่องเครื่องแม่ข่ายที่อยู่ห่างไกลไกลจะต้องสร้าง InputConnection ขึ้นมาเพื่อรับ InputStream จากการเชื่อมต่อ ส่วนการส่งข้อมูลไปยังเครื่องแม่ข่ายระยะไกลจะต้องสร้าง OutputConnection ขึ้นมาเพื่อรับ OutputStream จากการเชื่อมต่อ ใน J2ME มีการเชื่อมต่อทั้ง 3 ประเภทที่กำหนดไว้เพื่อจัดการกับ Stream ข้อมูลเข้าและข้อมูลออก ได้แก่

- InputConnection
- OutputConnection
- StreamConnection

การเชื่อมต่อ InputConnection กำหนดให้ InputStream สามารถรับส่งข้อมูลได้ ขณะที่ OutputConnection ก็กำหนดให้ OutputStream สามารถส่งข้อมูลได้ ส่วน StreanConnection จะกำหนดทั้ง InputStream และ OutputStream ให้มีความเหมาะสมกับความต้องการในการรับข้อมูล ส่งข้อมูล หรือทั้งรับและส่งข้อมูล

การเขียนโปรแกรมแบบเครือข่ายโดยใช้ซ็อกเก็ตใน J2ME มีขั้นตอนการทำงานดังนี้

1. เปิดการเชื่อมต่อแบบซ็อกเก็ตกับเครื่องแม่ข่ายระยะไกลหรืออุปกรณ์ไร้สายอื่นโดยใช้ Connector.open()
2. สร้าง InputStream และ OutputStream จากเปิดการเชื่อมต่อแบบซ็อกเก็ตเพื่อใช้ในการรับ – ส่งกลุ่มข้อมูล
3. รับ - ส่งข้อมูลจากเครื่องแม่ข่ายระยะไกลผ่านการเชื่อมต่อแบบซ็อกเก็ต ด้วยการอ่าน และเขียนใน InputStream หรือ OutputStream
4. ปิดการเชื่อมต่อแบบซ็อกเก็ตและ input / output stream ก่อนออกจากโปรแกรม

บทที่ 3
วิธีการดำเนินงาน

ในการสร้างเกมออนไลน์ที่สามารถมีผู้เล่นได้มากกว่า 1 คน บนโทรศัพท์มือถือนั้น มีส่วนสำคัญที่ต้องทำความเข้าใจ 2 ส่วน คือ การสร้างเกมบนโทรศัพท์มือถือ ซึ่งเกมที่สร้างขึ้นคือเกมบอมเบอร์แมน (Bomberman) ที่สามารถเล่นได้ 2 คนพร้อมกัน อีกส่วนหนึ่งคือการติดต่อสื่อสารกันระหว่างโทรศัพท์มือถือให้สามารถรับส่งข้อมูลกันได้โดยผ่านเครื่องแม่ข่าย (Server) ที่เป็นตัวกลางอยู่บนเครื่องคอมพิวเตอร์ โดยใช้วิธีการติดต่อกันแบบซ็อกเก็ต (Socket)

รูปที่ 3.1 แผนภาพแสดงขั้นตอนการทำงานของเกมออนไลน์บนมือถือ

3.1 การออกแบบเกมในส่วนติดต่อผู้ใช้

เกมบอมเบอร์แมน เป็นเกมที่มีลักษณะเป็นช่อง (Block) ทำให้การเขียนโปรแกรมไม่ยุ่งยากมากนัก จอภาพที่แสดงผลมีขนาด 7x7 แต่ละช่องจะแสดงออกมาเป็นรูปภาพสามารถแยกออกเป็น 3 ส่วน ประกอบด้วย

Floor เป็นส่วนที่ตัวละครสามารถเดินผ่านได้

Block เป็นส่วนที่ตัวละครเดินผ่านไม่ได้ แต่เมื่อส่วนนี้ถูกระเบิดจะเปลี่ยนเป็น Floor

Wall เป็นส่วนที่ตัวละครเดินผ่านไม่ได้ เมื่อถูกระเบิดจะไม่มีการเปลี่ยนแปลง

การเดินของตัวละครบอมเบอร์แมน จะเกิดขึ้นเมื่อลูกข่ายมีการกดปุ่มหมายเลข 2, 4, 6 หรือ 8 จากโทรศัพท์มือถือ การเดินของตัวละครทั้งสองสามารถเดินซ้อนทับตำแหน่งกันได้ ซึ่งแต่ละปุ่มจะทำหน้าที่แตกต่างกันออกไป ดังนี้

เมื่อกดปุ่มหมายเลข 2 เดินขึ้นบน

เมื่อกดปุ่มหมายเลข 4 เดินทางซ้าย

เมื่อกดปุ่มหมายเลข 6 เดินทางขวา

เมื่อกดปุ่มหมายเลข 8 เดินลงล่าง

การวางระเบิดของตัวละคร จะเกิดขึ้นเมื่อเครื่องลูกข่ายมีการกดปุ่มหมายเลข 5 ซึ่งการวางระเบิดนี้ผู้เล่น 1 คนสามารถวางระเบิดได้ 2 ลูก

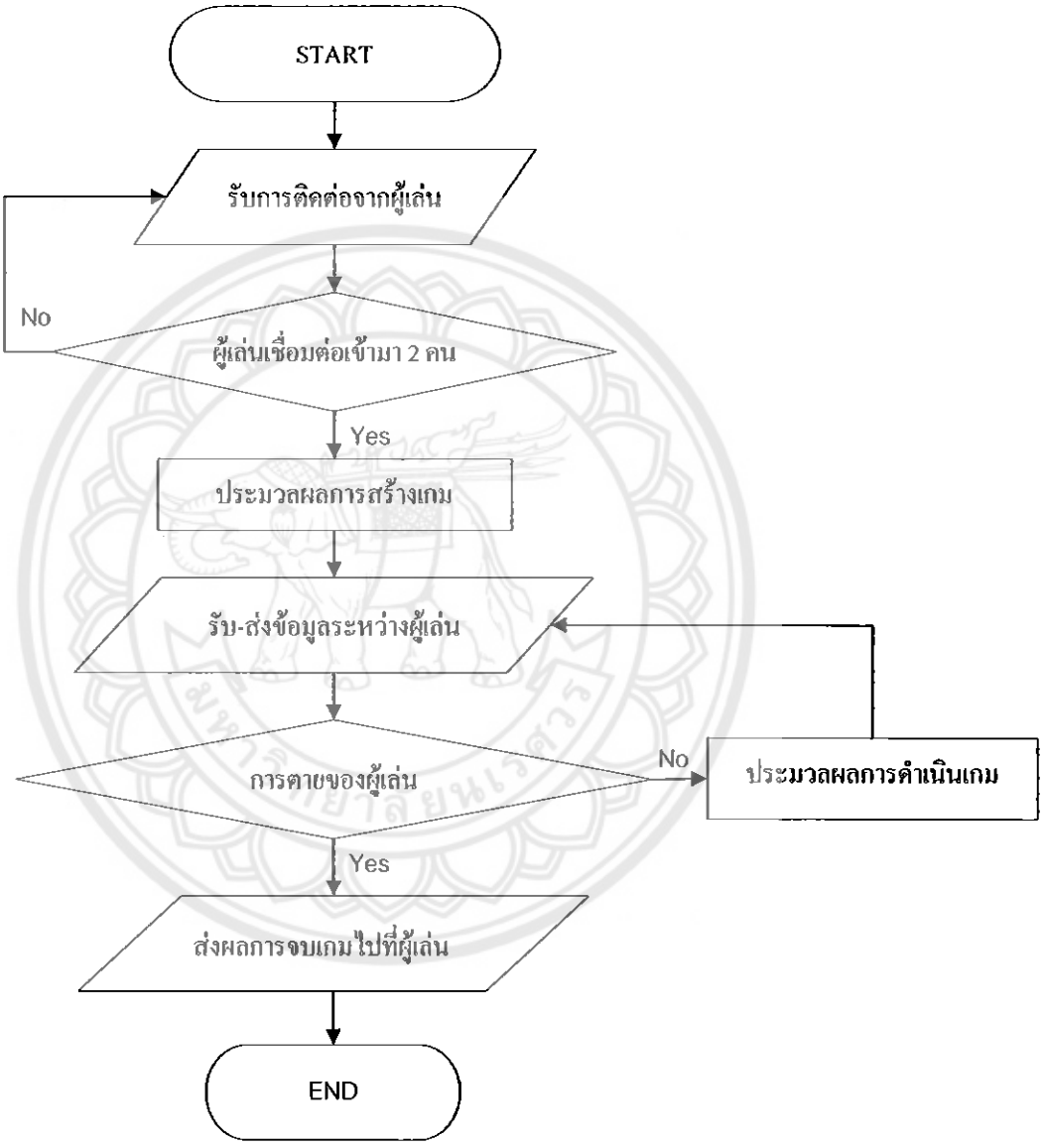
การระเบิดของลูกระเบิด จะเกิดขึ้นเมื่อการวางระเบิดลูกนั้นผ่านไปประมาณ 3 วินาที การระเบิดนี้ทำให้เกิดไฟในตำแหน่งรอบๆ ของตำแหน่งที่วางระเบิด

การจบเกมหรือการตายของตัวละคร จะเกิดขึ้นเมื่อตัวละครฝ่ายใดฝ่ายหนึ่งอยู่ในตำแหน่งเดียวกับไฟที่เกิดจากการระเบิด ก็คือตัวละครเดินไปถูกไฟหรือไฟที่เกิดจากการระเบิดถูกตัวละคร

การแสดงรูปบนจอภาพของโทรศัพท์มือถือซึ่งเป็นเครื่องลูกข่ายในเกมออนไลน์นี้ไม่สามารถวาดรูปเองได้ จะต้องรอรับค่าที่ใช้สำหรับการวาดจากเครื่องแม่ข่ายเท่านั้น เพราะเกมออนไลน์จะมีเครื่องลูกข่ายมากกว่า 1 ตัวที่ทำงานพร้อมกัน การกระทำทุกอย่างของผู้เล่นคนหนึ่งอีกฝ่ายหนึ่งจะต้องรับรู้ด้วย ดังนั้น การประมวลผลทุกๆ อย่างจะต้องทำที่เครื่องแม่ข่าย และส่งค่าที่ได้ไปที่เครื่องลูกข่ายทุกตัวในแบบกระจาย (Board cast) ที่เชื่อมต่อกันอยู่ในเครือข่าย

3.2 การทำงานของโปรแกรมประยุกต์ฝั่งเครื่องแม่ข่าย (Server)

รูปแบบของโปรแกรมประยุกต์ซึ่งเป็นเกมออนไลน์บนโทรศัพท์มือถือที่เขียนด้วยภาษาจาวา (JAVA) ใช้การติดต่อสื่อสารกันแบบซ็อกเก็ต โดยที่โปรแกรมประยุกต์นี้แบ่งออกเป็น 2 ส่วน คือ ส่วนของแม่ข่ายและลูกข่าย



รูปที่ 3.2 แผนภาพแสดงขั้นตอนการทำงานของเครื่องแม่ข่าย

เกมออนไลน์บนมือถือจะเริ่มจากเครื่องแม่ข่ายเปิดบริการไว้ก่อน แล้วมีเครื่องลูกข่ายที่เป็นผู้เล่นเชื่อมต่อเข้ามา โดยเริ่มแรกผู้เล่นจะต้องโปรแกรมประยุกต์ที่เป็นเกมอยู่ในโทรศัพท์มือถือของตัวเองก่อน เมื่อต้องเข้าเล่นเกมก็จะทำการเปิดโปรแกรมประยุกต์แล้วเชื่อมต่อมาที่เครื่องแม่ข่าย ก็จะสามารถเล่นเกมผ่านเครือข่ายได้

เกมออนไลน์ที่สร้างขึ้นจะมีการเริ่มต้นเกมก็ต่อเมื่อมีผู้เล่นเชื่อมต่อเข้ามาในระบบเครือข่ายในขณะนั้น 2 ผู้เล่น เครื่องแม่ข่ายจะมีความสำคัญมาก เพราะจะเก็บทุกสิ่งทุกอย่างไว้ ตั้งแต่ตารางของเกม ตำแหน่งของผู้เล่นทั้งสอง ตำแหน่งของระเบิด ตำแหน่งของไฟ หมายเลขของภาพที่เครื่องลูกข่ายต้องทำการวาด ตรวจสอบตำแหน่งใดที่ผู้เล่นสามารถเดินได้ เดินไม่ได้ รวมทั้งยังตรวจสอบการจบเกมหรือผู้เล่นตาย

เมื่อเครื่องแม่ข่ายได้รับการเชื่อมต่อมาจากทั้ง 2 ผู้เล่นแล้ว ก็จะทำการส่งค่าเริ่มต้นของเกมไปให้ทั้ง 2 ผู้เล่นในแบบการแพร่กระจาย (Board cast) ทำให้ผู้เล่นทั้งสองฝ่ายที่เข้ามาในระบบออนไลน์สามารถมองเห็นกันได้ในทุกๆ การเคลื่อนไหว ซึ่งค่าเริ่มต้นของเกมประกอบด้วย ค่าที่ใช้ในการวาดตารางและค่าที่ใช้ในการวาดตัวละคร

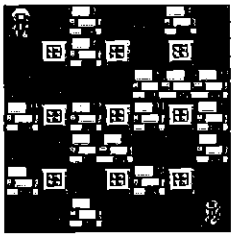
ค่าที่ใช้วาดตารางขนาด 7x7 จะมีค่าที่แตกต่างกัน 3 ค่า คือ 1, 2 และ 3 เพื่อทำการวาดรูปในตารางทางฝั่งผู้เล่น ค่าต่างๆ เหล่านี้จะถูกเก็บไว้ในตัวแปร map[i][j] การกำหนดเริ่มต้นที่เป็นตาราง มีวิธีการกำหนดค่าดังนี้

- ค่า map[i][j] = 3 แสดงรูป Wall เป็นส่วนที่แสดงขึ้นตายตัวทุกๆ เกม เครื่องแม่ข่ายจะกำหนดตำแหน่งที่ (1, 1), (1, 3), (1, 5), (3, 1), (3, 3), (3, 5), (5, 1), (5, 3) และ (5, 5) มีค่าเป็น 3

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

ในตำแหน่งที่ (0, 0), (0, 1), (1, 0), (5, 6), (6, 5) และ (6, 6) จะกำหนดค่าให้เป็น 1 เพื่อทำให้ผู้เล่นสามารถเดินได้ในตอนเริ่มต้น

ค่าเริ่มต้นของตัวละครทั้งสองตัว เครื่องแม่ข่ายจะส่งค่าเริ่มต้นสำหรับการวาดตัวละครให้ทางฝั่งผู้เล่นเพื่อทำการวาดรูปตัวละคร 3 ค่า คือ ตัวแปรที่เก็บรูปกราฟิก, พิกัดในแกน x และพิกัดในแกน y ส่วนตำแหน่งการแสดงผลของรูปกราฟิกไม่ได้ส่งให้เครื่องลูกข่าย เพราะการวาดรูปทุกๆ รูปใช้ค่าเดียวกันทั้งหมด คือ 0 ซึ่งในเครื่องลูกข่ายสามารถกำหนดค่าได้เอง ในการเริ่มต้นเกมผู้เล่นคนที่ 1 จะอยู่ในตำแหน่ง (0, 0) ส่วนคนที่ 2 อยู่ในตำแหน่ง (6, 6)



รูปที่ 3.3 แสดงการเริ่มต้นของเกม

ค่าที่รับ-ส่งกันระหว่างเครื่องแม่ข่ายและผู้เล่นจะเป็นข้อมูลประเภท String โดยจะมีการส่งข้อตกลง (Protocol) เป็นข้อความที่มีคำขึ้นต้นต่างๆ กัน ในตำแหน่งที่ 0 และ 1 ของค่าที่ส่ง ส่วนตำแหน่งต่อๆ ไปจะเป็นค่าที่รับ-ส่งกันจริง ทั้งนี้เพื่อเป็นการแยกความแตกต่างของส่วนต่างๆ ในเกม

ค่าที่เครื่องแม่ข่ายรับมาจากเครื่องลูกข่ายก็คือค่าที่รับจากปุ่มกดนั้น จะมีการส่งข้อตกลง เป็นข้อความที่มีคำขึ้นต้นด้วย K: แล้วตามด้วยค่าของปุ่มที่กด

การเดินของตัวละครซึ่งจะแสดงผลออกทางจอภาพของผู้เล่นทั้งสองนั้น ค่าที่รับมาเพื่อใช้ในการเดินจะรับมาจากปุ่มกดของผู้เล่น คือ ปุ่มหมายเลข 2, 4, 6 และ 8 เมื่อผู้เล่นกดปุ่มตัวละครยังไม่สามารถเดินได้ โปรแกรมประยุกต์ที่อยู่ในเครื่องของผู้เล่นจะทำการตรวจสอบเหตุการณ์การกดปุ่มว่าตรงกับค่าใด และจะส่งค่าสำหรับการกดนั้นๆ ไปที่เครื่องแม่ข่ายเพื่อประมวลผลว่าการกดปุ่มนั้นมาจากผู้เล่นฝ่ายใดและเป็นค่าใด เครื่องแม่ข่ายจะนำค่าที่รับมาไปใช้ในการปรับปรุงค่า (Update) ทั้งตำแหน่งและรูปภาพของผู้เล่นฝ่ายที่มีการเดินหรือกดปุ่ม จากนั้นจะส่งค่าที่ใช้ในการวาดรูปตัวละครที่ปรับปรุงแล้วไปที่ผู้เล่นทั้งสองในแบบการแพร่กระจาย ดังนั้นผู้เล่นฝ่ายที่ไม่มีการกดปุ่มในขณะนั้นก็จะสามารถมองเห็นได้ว่าฝ่ายตรงข้ามมีการเดินเกิดขึ้น

การวางระเบิด การทำงานและการรับ-ส่งค่าต่างๆ จะคล้ายกับการเดินของตัวละคร เมื่อมีผู้เล่นฝ่ายใดฝ่ายหนึ่งกดปุ่มหมายเลข 5 เพื่อวางระเบิด เครื่องแม่ข่ายจะตรวจสอบดูว่าว่าผู้เล่นฝ่ายใดเป็นฝ่ายกดปุ่มวางระเบิดเพื่อหาค่าต่างๆ ที่จะต้องวาดระเบิด รวมทั้งต้องเก็บค่าเหล่านั้นเอาไว้ด้วย โดยที่ค่าตำแหน่งของระเบิดจะเป็นตำแหน่งเดียวกับที่ผู้เล่นฝ่ายที่วาง จากนั้นเครื่องแม่ข่ายส่งค่าที่ใช้ในการวาดรูประเบิดไปให้ผู้เล่นทั้งสองก่อนที่ระเบิดจะระเบิดเป็นไฟจะมีการหน่วงเวลาไว้ประมาณ 3 วินาที

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

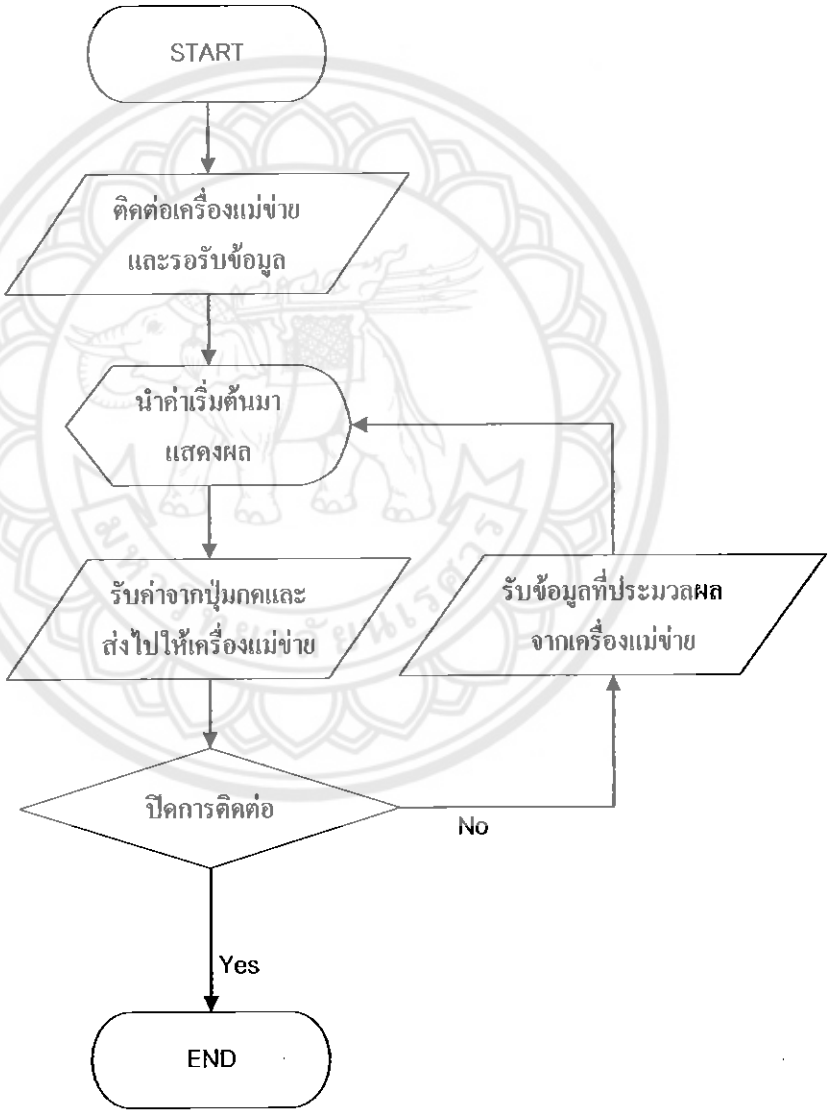
ถ้าเป็นรูป Floor จะแสดงรูปไฟออกมาประมาณ 1 วินาที จากนั้นในตำแหน่งที่แสดงรูปไฟจะเปลี่ยนเป็นรูป Floor เหมือนเดิม

ถ้าเป็นรูป Block จะแสดงรูปไฟออกมาประมาณ 1 วินาทีเช่นกัน จากนั้นในตำแหน่งที่แสดงรูปไฟจะเปลี่ยนเป็นรูป Floor

การจบเกมหรือการตายของผู้เล่น จะเกิดขึ้นเมื่อผู้เล่นเดินเข้าไปในตำแหน่งที่มีไฟอยู่หรือเมื่อเกิดระเบิดแล้วไฟที่เกิดขึ้น เกิดในตำแหน่งเดียวกับที่ผู้เล่นยืนอยู่

3.3 การทำงานของโปรแกรมประยุกต์ฝั่งเครื่องลูกข่าย (Client)

การทำงานของโปรแกรมประยุกต์ทางฝั่งเครื่องลูกข่ายที่เป็นโทรศัพท์มือถือ เมื่อเชื่อมต่อกับเครื่องแม่ข่ายเข้าไปอยู่ในระบบออนไลน์แล้ว จะมีหน้าที่รับค่าตัวแปรต่างๆ จากเครื่องแม่ข่ายแล้วนำค่าเหล่านั้นมาทำการวาดรูปและรับค่าจากการกดปุ่มที่ใช้ในการเดินของตัวละคร



รูปที่ 3.4 แผนภาพแสดงขั้นตอนการทำงานของเครื่องลูกข่าย

โปรแกรมประยุกต์ทางฝั่งเครื่องลูกข่ายจะมีการสร้างช่องรับข้อความที่เป็น TextField ใช้รับชื่อของผู้เล่นก่อนการเกม เพื่อนำชื่อของผู้เล่นฝ่ายชนะไปแสดงเมื่อจบเกม

ในการวาดรูปของเครื่องลูกข่าย ค่าที่รับมาจากเครื่องแม่ข่ายจะเป็นค่าตำแหน่งในอาเรย์ที่มีขนาด 7x7 คือ มีค่าตั้งแต่ (0, 0) ถึง (6, 6) จะไม่สามารถนำมาวาดรูปได้ เพราะการวาดรูปในเครื่องลูกข่าย ค่าที่ใช้ในการวาดจริงๆ จะใช้ค่าตำแหน่งของพิกเซลที่เป็นหน้าจอของโทรศัพท์มือถือ (x, y) หน้าจอของตัวลองโทรศัพท์มือถือที่นำมาใช้มีขนาด 128x128 พิกเซล เพราะฉะนั้นก่อนทำการวาดเครื่องลูกข่ายจะต้องนำค่าที่รับมาจากเครื่องแม่ข่ายมาคำนวณหาตำแหน่งที่ถูกต้องด้วยสมการ x = (i * 16) + 8 และ y = (j * 16) + 8 แต่ละช่องในตารางจะมีขนาด 16x16 พิกเซล และยังมีส่วนของขอบที่ไม่ได้ใช้แสดงผลอีก 8 พิกเซล

ค่าที่เครื่องลูกข่ายได้รับมาจากเครื่องแม่ข่ายที่ใช้นั้น จะมีอยู่ 3 ลักษณะที่แตกต่างกัน คือ ค่าที่ใช้ในการวาดตาราง ค่าที่ใช้ในการวาดตัวละคร และค่าที่ใช้ในการแสดงชื่อของผู้เล่นฝ่ายชนะเมื่อจบเกม

1. ค่าที่ใช้ในการวาดตาราง เครื่องแม่ข่ายจะส่งค่ามาทั้งหมด 51 ค่า โดยมีข้อตกลงในการส่งคือข้อความที่ขึ้นต้นด้วย T: ซึ่งเป็นค่าตัวที่ 1 และ 2 ส่วนค่าที่เหลืออีก 49 ค่า จะเป็นค่าที่ใช้ในการวาดตารางขนาด 7x7

2. ค่าที่ใช้ในการวาดตัวละคร เครื่องแม่ข่ายจะส่งค่ามาทั้งหมด 8 ค่า โดยมีข้อตกลงในการส่งคือข้อความที่ขึ้นต้นด้วย H: ซึ่งเป็นค่าตัวที่ 1 และ 2 ส่วนค่าที่เหลืออีก 6 ค่า จะเป็นค่าที่ใช้ในการวาดตัวละครของผู้เล่นคนที่ 1 และ 2

3. ค่าที่ใช้ในการแสดงชื่อของผู้เล่นฝ่ายชนะเมื่อจบเกม มีข้อตกลงในการส่งคือข้อความที่ขึ้นต้นด้วย W: ซึ่งเป็นค่าตัวที่ 1 และ 2 ส่วนค่าที่เหลือตั้งค่าที่ 3 เป็นต้นไปจะเป็นชื่อของผู้เล่นฝ่ายที่ชนะ

บทที่ 4
ผลการดำเนินงาน

การทำงานของเกมออนไลน์บนโทรศัพท์มือถือนี้ เริ่มต้นด้วยการเปิดโปรแกรมประยุกต์ทางฝั่งแม่ข่าย (Server) ไว้ที่คอมพิวเตอร์เครื่องหนึ่ง จากนั้นก็เปิดโปรแกรมประยุกต์ของฝั่งเครื่องลูกข่าย (Client) ที่เป็นตัวจำลอง (Emulator) ของโทรศัพท์มือถือ ซึ่งมี 2 ผู้เล่นในเครือข่าย โดยทั้ง 3 โปรแกรมประยุกต์อาจจะอยู่ในคอมพิวเตอร์เครื่องเดียวกันหรือคนละเครื่องก็ได้

4.1 การทดสอบการทำงาน

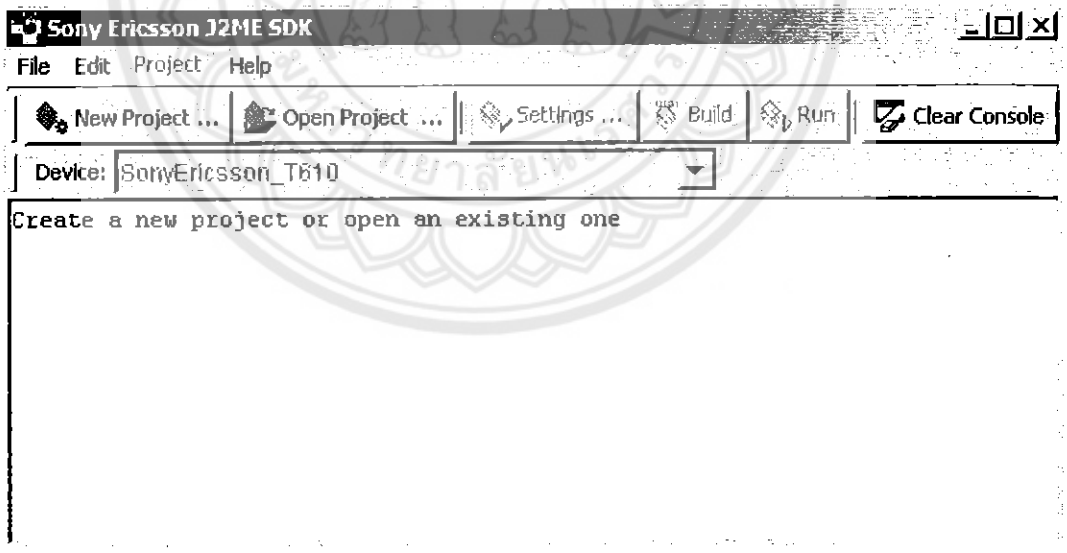
การทดสอบการทำงานของเกมออนไลน์บนโทรศัพท์มือถือที่สร้างนี้ เป็นการตรวจสอบและแก้ไขข้อผิดพลาดต่างๆ ที่เกิดขึ้นจากการเขียนโปรแกรมทั้งทางฝั่งแม่ข่ายและลูกข่าย เพื่อทำให้โปรแกรมประยุกต์มีความถูกต้องตามที่ออกแบบไว้มากที่สุด รวมทั้งยังเป็นการทดสอบอุปกรณ์โทรศัพท์มือถือที่นำมาใช้ว่ามีทรัพยากรเพียงพอต่อการนำโปรแกรมประยุกต์นี้ไปใช้หรือไม่

4.2 วิธีการทดสอบการทำงาน

ในการทดสอบได้เปิดโปรแกรมประยุกต์ฝั่งแม่ข่ายไว้ในเครื่องคอมพิวเตอร์ที่มี IP Address 127.0.0.1 และช่องส่งข้อมูล (Port) 6000 เครื่องแม่ข่ายจะเริ่มให้บริการโดยรอการติดต่อเข้ามาจากผู้เล่นทั้งสอง จากนั้นทำการเปิดโปรแกรมประยุกต์ของผู้เล่นคนที่ 1 เมื่อผู้เล่นติดต่อเข้าไปเพื่อร้องขอการเล่นเกม ที่จอภาพของผู้เล่นคนที่ 1 จะยังไม่แสดงผลอะไร จะต้องรอให้มีผู้เล่นอีกคนเชื่อมต่อเข้ามา เมื่อมีผู้เล่นอยู่ในระบบออนไลน์ 2 คนแล้ว เครื่องแม่ข่ายจะสร้างเกมขึ้นมาแล้วส่งค่าต่างๆ ที่ใช้ในการเล่นไปให้ผู้เล่นทั้งสอง จอภาพของผู้เล่นจะแสดงเป็นเกมบอมเบอร์แมนขึ้นมา โดยที่การทดสอบโปรแกรมประยุกต์ทั้งฝั่งแม่ข่ายและผู้เล่นทั้งสองจะอยู่ในคอมพิวเตอร์เครื่องเดียวกัน ภาพที่ 4.1 แสดงการเปิดโปรแกรมประยุกต์ฝั่งแม่ข่าย ซึ่งเป็นการเปิดบริการรอรับการเชื่อมต่อจากเครื่องลูกข่ายที่จะเข้ามาเล่นเกม

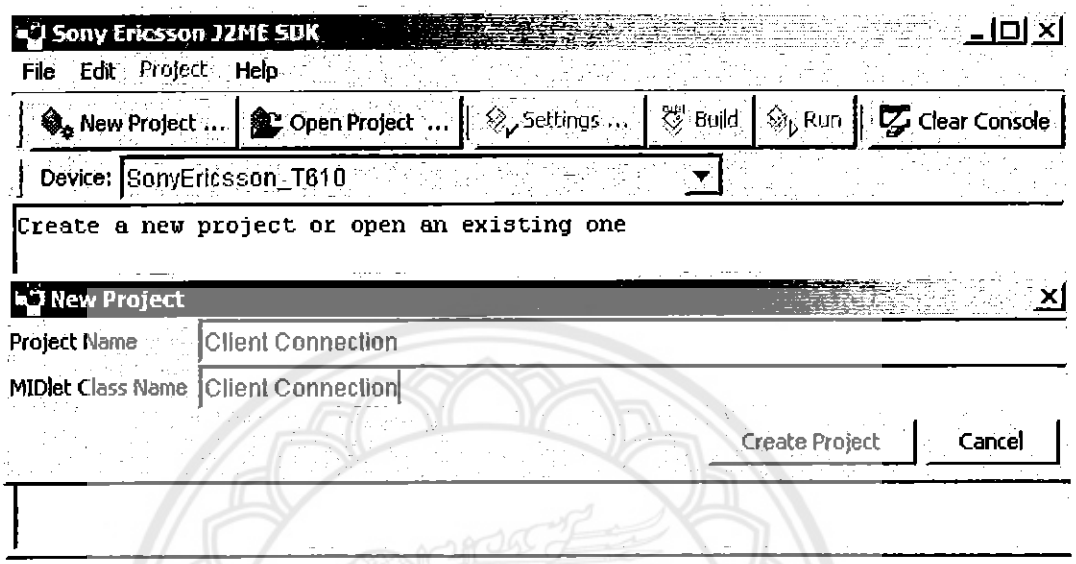
รูปที่ 4.1 แสดงการเปิดโปรแกรมประยุกต์ฝั่งแม่ข่าย

ในการเปิดโปรแกรมโปรแกรมประยุกต์ฝั่งลูกข่ายหรือผู้เล่นนั้น จะต้องเปิดในตัวจำลองของโทรศัพท์มือถือ ซึ่งแสดงดังรูปที่ 4.2



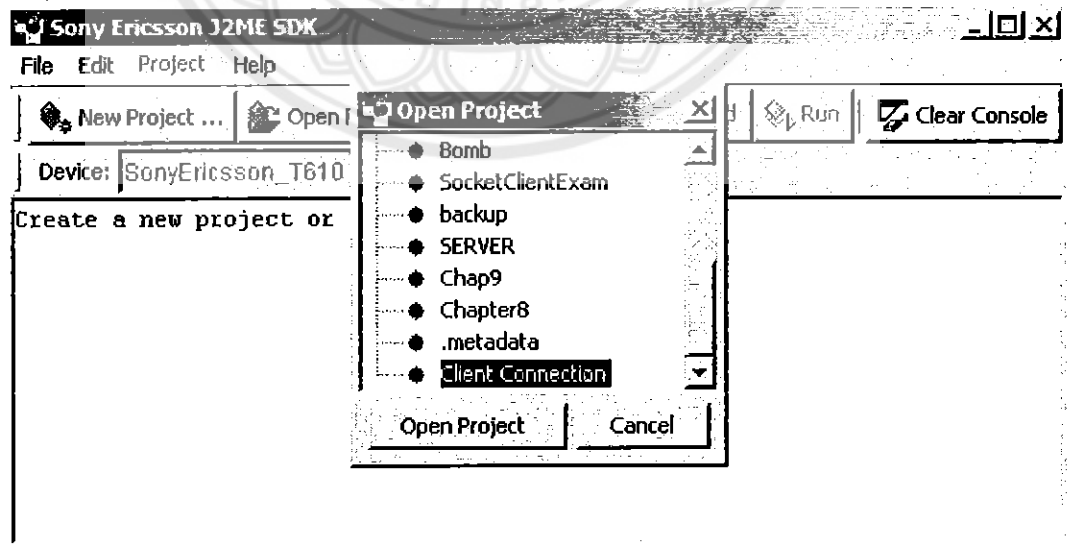
รูปที่ 4.2 แสดงโปรแกรมประยุกต์ตัวจำลองของโทรมือถือ

การเริ่มต้นการใช้งานโปรแกรมประยุกต์ตัวจำลองของโทรมือถือนั้น ในขั้นแรกจะสร้างโปรเจ็กต์ใหม่ขึ้นมาก่อน โดยเลือกที่ New Project ในโปรแกรมประยุกต์ ดังรูปที่ 4.3 โปรเจ็กต์ทั้งหมดที่สร้างขึ้นถูกเก็บไว้ไดเร็กทอรี /apps



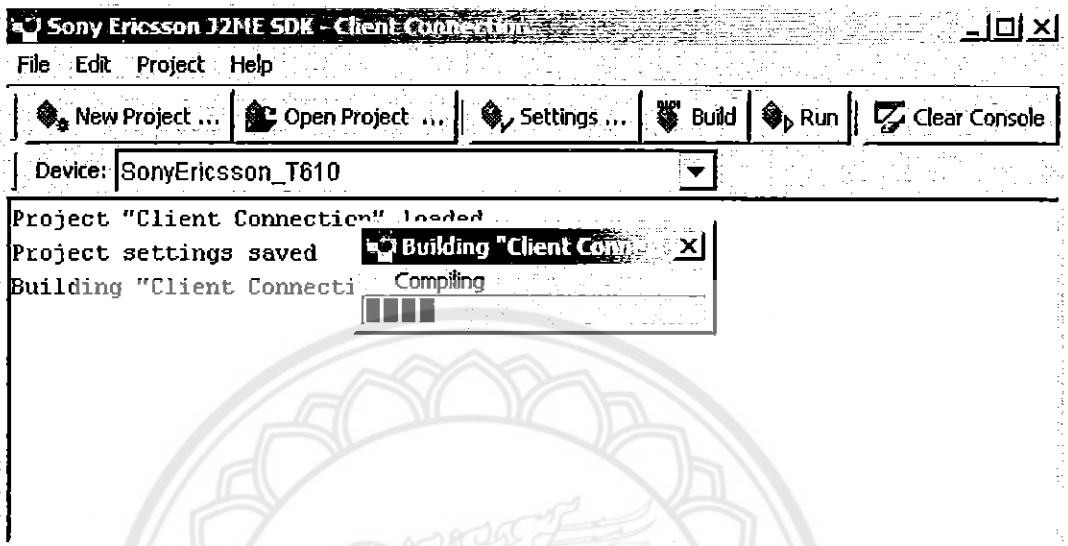
รูปที่ 4.3 แสดงการสร้างโปรเจ็กต์ใหม่

การทดสอบการทำงานของโปรแกรมที่สร้างขึ้น ในขั้นแรกต้องเปิดโปรเจ็กต์ขึ้นมาก่อน โดยเลือกที่ Open Project แล้วเลือกโปรเจ็กต์ที่สร้างไว้ ดังรูปที่ 4.4

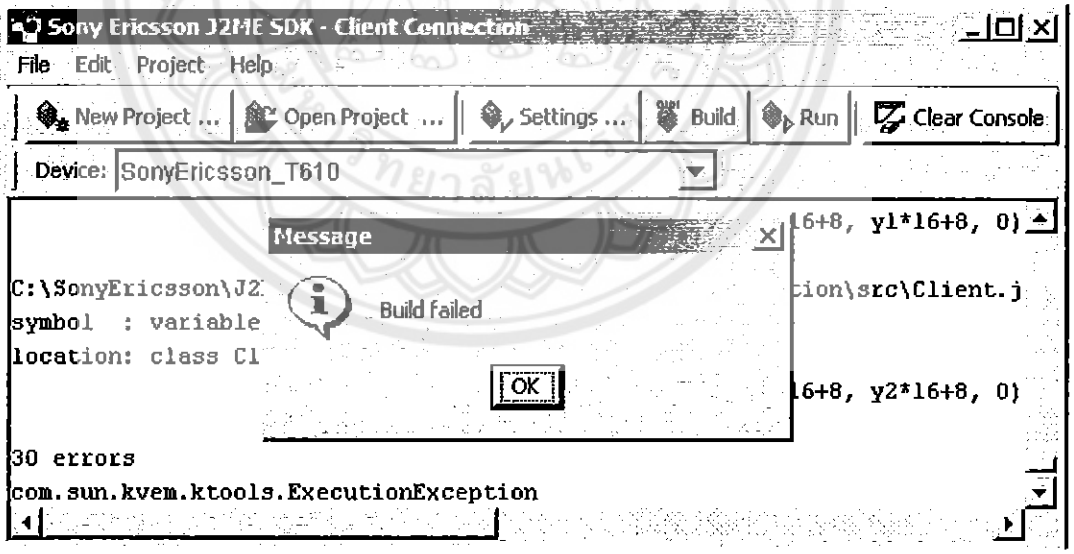


รูปที่ 4.4 แสดงการเปิดโปรเจ็กต์

การคอมไพล์ (Compile) ของโปรเจ็กต์ที่สร้างขึ้นสามารถทำได้โดยเลือกที่ Build ดังรูปที่ 4.5 ถ้าการคอมไพล์ผ่าน โปรแกรมประยุกต์จะแจ้งว่า Build complete แต่ถ้าการคอมไพล์มีข้อผิดพลาดเกิดขึ้น โปรแกรมประยุกต์ก็จะแจ้ง Build failed และยังแจ้งด้วยว่าส่วนใดที่ผิดพลาด ดังรูปที่ 4.6



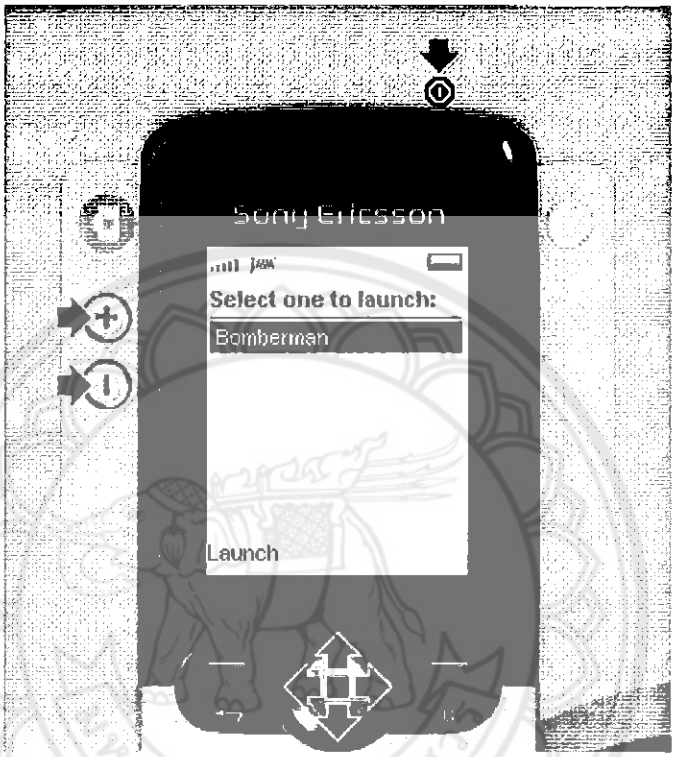
รูปที่ 4.5 แสดงการคอมไพล์ของโปรเจ็กต์



รูปที่ 4.6 แสดงการคอมไพล์ที่มีข้อผิดพลาดเกิดขึ้น

4.3 ผลการดำเนินงาน

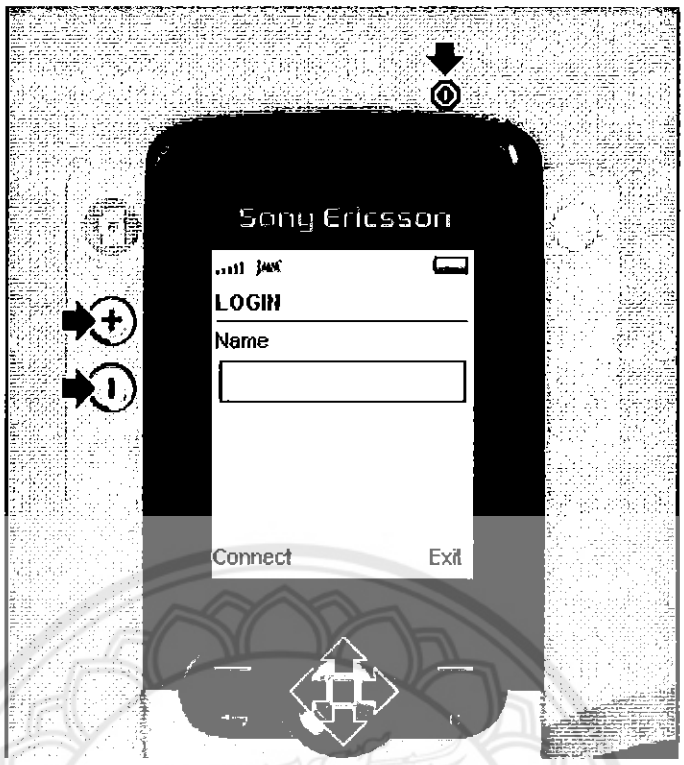
หลังจาก Compile เรียบร้อยแล้ว ขั้นตอนต่อไปของการทดสอบก็คือ การรันตัวจำลองของโทรศัพท์มือถือขึ้นมา โดยเลือกที่ Run ที่หน้าจอจะปรากฏรูปของตัวจำลองโทรศัพท์มือถือขึ้นมา ดังรูปที่ 4.7



รูปที่ 4.7 แสดงตัวจำลองของโทรศัพท์มือถือ

ในบางครั้งการรันของโปรแกรมประยุกต์จะมีปัญหาเกิดขึ้น คือไม่สามารถรันได้เมื่อใช้งานไปในระยะเวลาหนึ่ง โดยโปรแกรมประยุกต์จะแจ้งว่า MemoryOutOfBoundsException เนื่องจากการมีหน่วยความจำที่จำกัดของโทรศัพท์มือถือ

เมื่อผู้เล่นต้องการเล่นเกมออนไลน์ จะต้องทำการ LOGIN เข้าสู่ระบบออนไลน์ก่อน โดยการใส่ชื่อเข้าไปเพื่อเป็นการลงทะเบียน ดังรูปที่ 4.8 และ 4.9

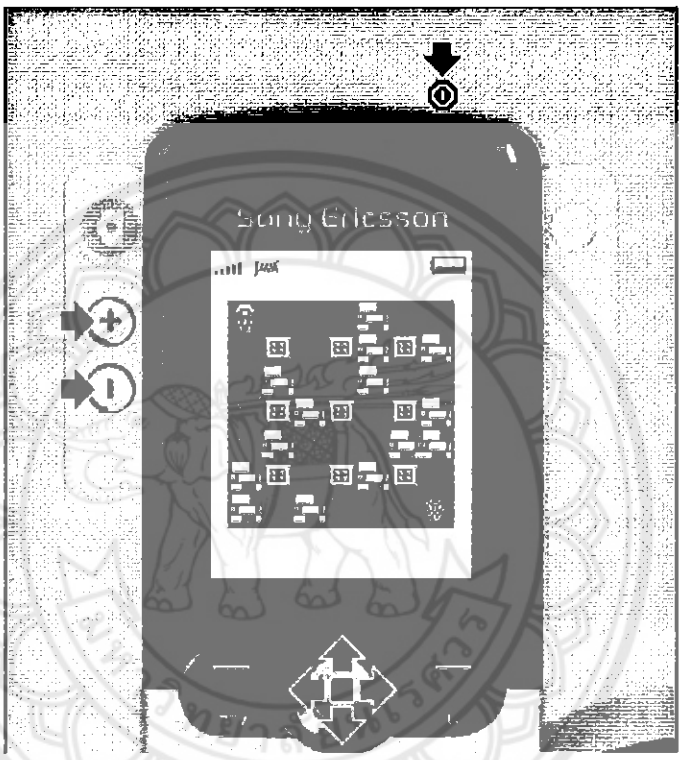


รูปที่ 4.8 แสดงหน้า LOGIN ของเกมออนไลน์

รูปที่ 4.9 แสดงการใส่ชื่อเพื่อลงทะเบียน

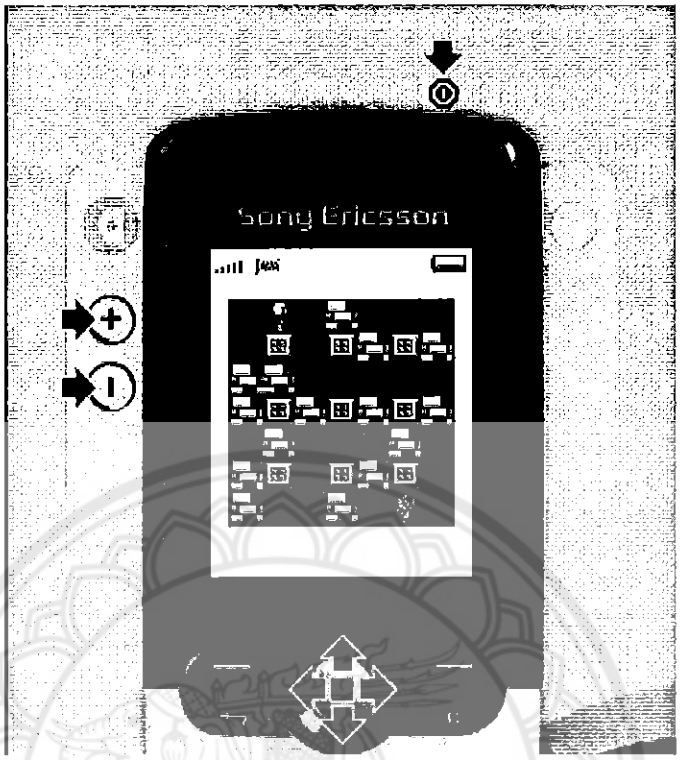
เมื่อใส่ชื่อเรียบร้อยแล้ว ผู้เล่นสามารถเริ่มเล่นเกมได้โดยเลือกที่ปุ่ม Connect ในการเริ่มต้นของเกม เครื่องแม่ข่ายจะรอให้มีผู้เล่นเชื่อมต่อเข้ามา 2 ผู้เล่นก่อนจึงจะส่งค่าเริ่มต้นของเกมไปให้ผู้เล่นทั้งสอง ดังรูปที่ 4.10

ปัญหาที่เกิดขึ้นในการแสดงผล คือการแสดงผลในจอภาพของผู้เล่นทั้งสองจะเกิดขึ้นไม่พร้อมกัน โดยผู้เล่นที่เชื่อมต่อเข้ามาทีหลังจะแสดงผลได้ช้ากว่าเล็กน้อย



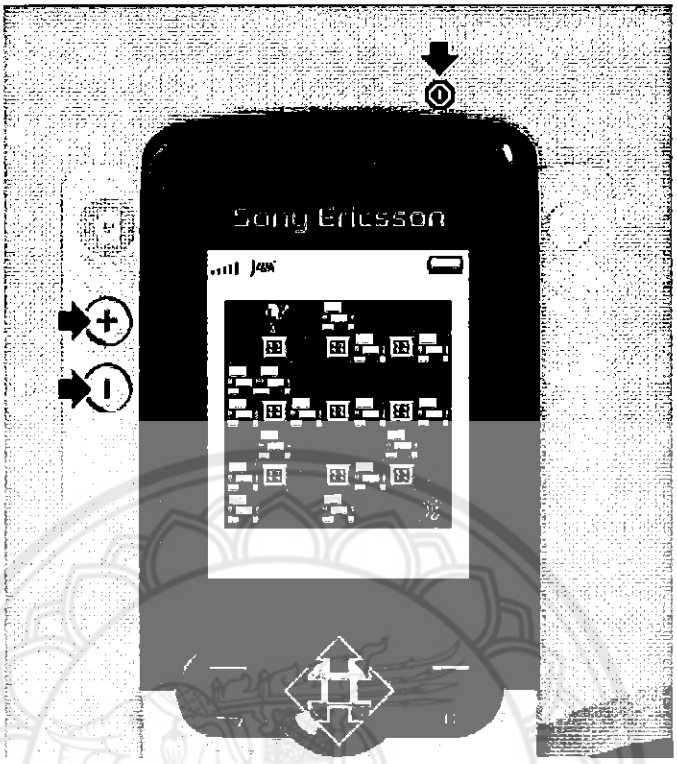
รูปที่ 4.10 แสดงการเริ่มต้นของเกม

เมื่อเครื่องแม่ข่ายประมวลผลเกมแล้ว จะส่งข้อมูลไปให้เครื่องลูกข่ายทำการแสดงผล ซึ่งจะมีการแสดงผลที่เหมือนกันระหว่างเครื่องลูกข่ายทั้งสอง การเริ่มต้นของเกมจะมีตัวผู้เล่นอยู่คนละฝั่งของเกมและจะทำการวางระเบิดเพื่อทำลายฝ่ายตรงข้าม



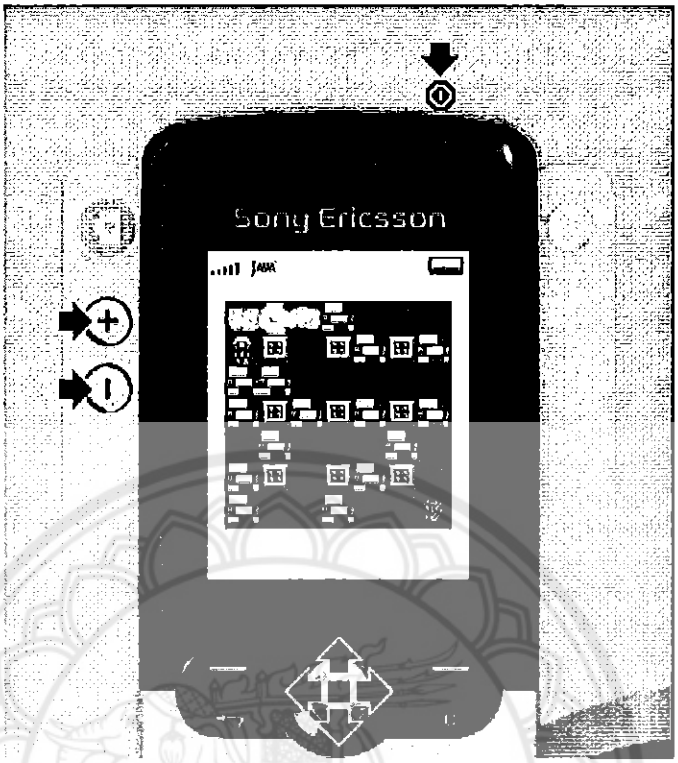
รูปที่ 4.11 แสดงการเดินของผู้เล่น

รูปที่ 4.11 เป็นการแสดงการเดินของผู้เล่นซึ่งจะแสดงผลที่เครื่องลูกข่ายทั้งสองเหมือนกัน แต่อาจจะมีความคลาดเคลื่อนเล็กน้อยสำหรับการแสดงผลในจอภาพของเครื่องลูกข่ายทั้งสอง คือแสดงผลออกมาไม่พร้อมกัน



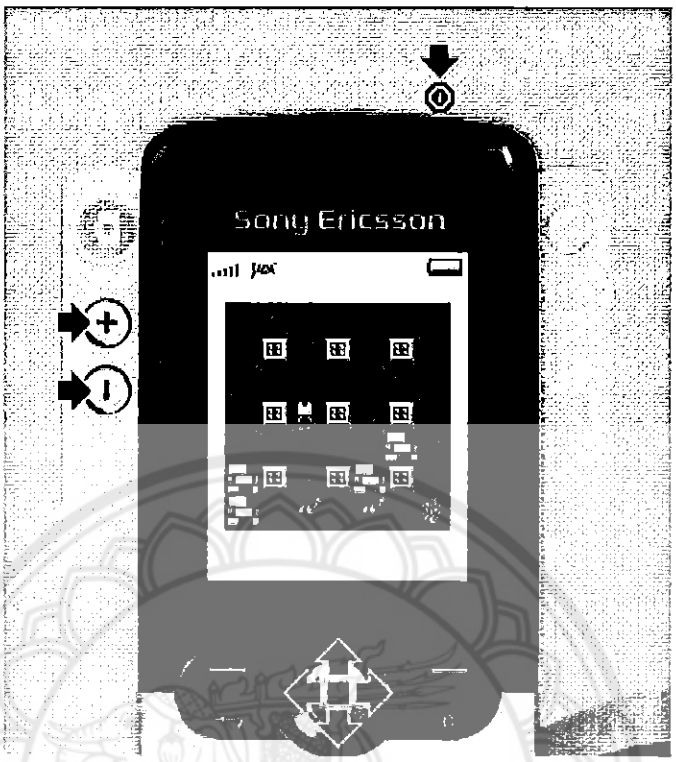
รูปที่ 4.12 แสดงการวางระเบิด

ลักษณะการวางระเบิดของผู้เล่นเมื่อผู้เล่นทำการกดปุ่มหมายเลข 5 ก็คือจะทำการวางลูกระเบิด ซึ่งลูกระเบิดอยู่ในตำแหน่งเดียวกับตัวละคร ลูกระเบิดนี้ผู้เล่นไม่สามารถเดินผ่านได้ และจะระเบิดภายในระยะเวลาประมาณ 3 วินาที ดังรูปที่ 4.12



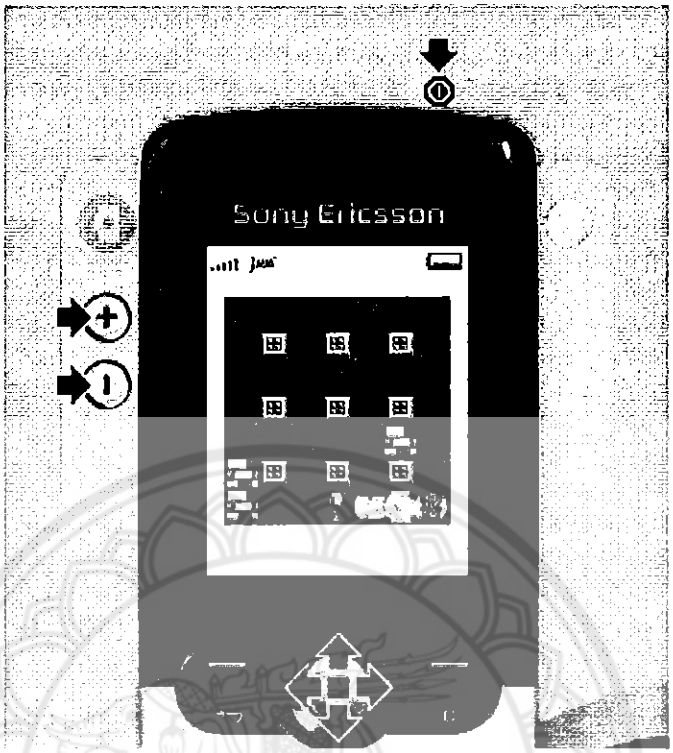
รูปที่ 4.13 แสดงการระเบิด

เมื่อเวลาผ่านไปประมาณ 3 วินาที ระเบิดที่วางไว้จะเกิดการระเบิดและแสดงรูปไฟออกมาในตำแหน่งรอบๆ ระเบิดที่วางถ้าตำแหน่งนั้นไม่ใช่รูป Wall การแสดงรูปไฟจะแสดงออกมาเป็นเวลาประมาณ 1 วินาที จากนั้นรูปไฟจะหายไป ดังรูปที่ 4.13



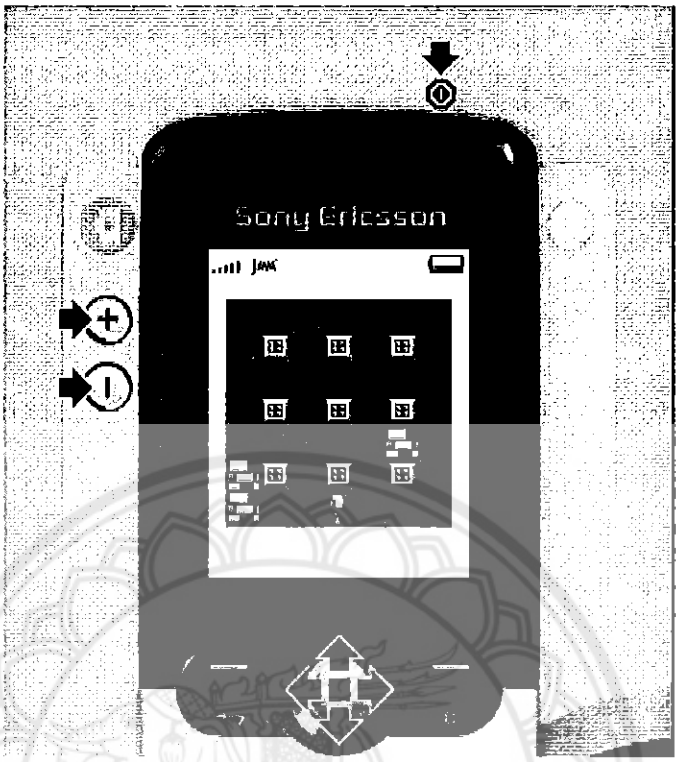
รูปที่ 4.14 แสดงการวางระเบิด 2 ลูก

ผู้เล่นสามารถทำการวางระเบิดได้ถึง 2 ลูกโดยที่ลูกระเบิดจะระเบิดตามเวลาที่กำหนด เพื่อให้ผู้เล่นได้มีโอกาสที่จะทำลายผู้เล่นฝ่ายตรงข้ามได้มากขึ้นได้ ดังรูปที่ 4.14



รูปที่ 4.15 แสดงการถูกไฟของตัวละคร

เมื่อมีผู้เล่นฝ่ายใดที่โดนการระเบิดไม่ว่าลูกระเบิดจะเป็นผู้เล่นของฝ่ายใดก็**ตามก็จะมีรูปไฟโดน**ตัว**ผู้เล่นนั้น**และทำให้ผู้เล่นฝ่ายที่โดนระเบิดตายไป ดังรูปที่ 4.15



รูปที่ 4.16 แสดงการตายของตัวละคร

เมื่อมีผู้เล่นฝ่ายใดฝ่ายระเบิดแล้ว ผู้เล่นฝ่ายนั้นจะไม่สามารถแสดงรูปตัวออกมา ซึ่งถือว่าผู้เล่นนั้นได้แพ้ไปแล้วในเกม ดังรูปที่ 4.16

รูปที่ 4.17 แสดงการจบเกม

เมื่อจบเกมก็จะมีการแสดงภาพของการจบเกมออกมา และจะแสดงชื่อของผู้เล่นที่เล่นชนะด้วย ดังรูปที่ 4.17

บทที่ 5
บทสรุป

5.1 บทสรุป

จากการจัดทำโปรแกรมประยุกต์ (Application) ที่เป็นเกมออนไลน์บนมือถือด้วย J2ME ด้วยการติดต่อสื่อสารกันแบบซ็อกเก็ต (Socket) ได้ข้อสรุปจากการทำโครงงานได้ดังนี้

จากการที่ได้ศึกษาและการพัฒนาเกมออนไลน์บนโทรศัพท์มือถือด้วยภาษาจาวา (JAVA) นี้ ทำให้คณะผู้จัดทำมีความเข้าใจในวิธีการเขียนโปรแกรมในระบบเครือข่ายที่มีอุปกรณ์ต่างชนิดกัน คือ ระหว่างคอมพิวเตอร์กับโทรศัพท์มือถือมากยิ่งขึ้น รวมทั้ง การเขียนโปรแกรมบนโทรศัพท์มือถือ ที่นอกเหนือจากการเรียนในห้องเรียน ซึ่งจากการศึกษาพบว่า การติดต่อสื่อสารของโทรศัพท์มือถือแต่ละเครื่องยังมีข้อจำกัดอยู่ คือ ไม่สามารถรองรับการทำงานของการติดต่อในรูปแบบต่างๆ ที่จะทำงานร่วมกันได้ทั้งหมด โดยการทดสอบในครั้งนี้ ใช้การติดต่อสื่อสารแบบซ็อกเก็ต ทำการติดต่อระหว่างเครื่องแม่ข่ายที่พัฒนาจากภาษาจาวา และเครื่องลูกข่ายที่ใช้ตัวจำลองของโทรศัพท์มือถือ (Emulator) เป็นตัวทดสอบ ผลที่ได้คือ เกมสามารถดำเนินไปได้โดยราบรื่น สำหรับในการทดสอบกับโทรศัพท์มือถือจริงๆ นั้น จะต้องนำไฟล์ของเครื่องแม่ข่ายไปไว้ที่เครื่องแม่ข่ายจริงซึ่งจำเป็นจะต้องมี J2SDK เป็นโปรแกรมรองรับการทำงานที่เครื่องแม่ข่ายด้วย ส่วนไฟล์ของเครื่องลูกข่ายจะต้องนำไปไว้ที่เครื่องโทรศัพท์มือถือที่รองรับระบบ J2ME เมื่อทำการติดต่อเครื่องแม่ข่ายกับลูกข่ายที่เป็นโทรศัพท์มือถือจริงนั้นไม่สามารถทำได้เพราะโทรศัพท์มือถือไม่รองรับการติดต่อของซ็อกเก็ต จึงทำให้ผลการทดลองที่ได้เป็นเพียงตัวทดสอบ และท้ายที่สุดสิ่งที่ได้รับจากการทำโครงงานในครั้งนี้คือ การทำให้รู้จักทำงานเป็นกลุ่ม การรู้จักแก้ไขปัญหาที่เกิดขึ้นจากการทำงานด้วยตัวเอง อันส่งผลให้การจัดทำโครงงานสามารถดำเนินการและผ่านไปได้ด้วยดี โดยประสบการณ์ที่ได้รับในครั้งนี้ ทางคณะผู้จัดทำสามารถนำไปประยุกต์ใช้ได้โดยตรงต่อการทำงานในชีวิตจริง

5.2 ข้อเสนอแนะ

- เครื่องแม่ข่ายที่รองรับส่วนใหญ่จะไม่มีการเปิดให้รันโปรแกรมบางอย่างได้ จะต้องพัฒนาโปรแกรมในลักษณะของ Servlet หรือ JSP เพื่อที่จะสามารถรันได้ทันทีบน Web Server
- ข้อตกลงที่ใช้ในการรับส่งข้อมูลส่วนใหญ่ ที่ใช้สำหรับการติดต่อของ J2ME นี้เป็น HTTP Protocol ซึ่งจะรองรับการติดต่อสื่อสารของโทรศัพท์มือถือได้
- เครื่องโทรศัพท์มือถือแต่ละเครื่องจะมีความสามารถในการรองรับไม่เหมือนกัน โดยบางเครื่องจะไม่สามารถใช้การทำงานของ J2ME ได้ไม่เหมือนกัน

เอกสารอ้างอิง

[1] เจนวิทย์ เหลืองอร่าม. ปิยวิทย์ เหลืองอร่าม. การเขียนโปรแกรมสำหรับ Wireless Application ด้วย J2ME. กรุงเทพมหานคร : โรงพิมพ์ธรรมสาร. 2546.

[2] ศิวณัฏฐ์ มาศสุรางค์. ประสิทธิ์ วรฉัตราวณิช. JAVA 2 MICRO EDITION J2ME คู่มือสำหรับเริ่มต้นพัฒนาจาวาบนมือถือ. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : โรงพิมพ์พิมพ์ดี. 2546.

[3] วีระศักดิ์ ซึงถาวร. Java Programming Volume I. กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน). 2543.

[4] วีระศักดิ์ ซึงถาวร. Java Programming Volume II. กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน). 2546.

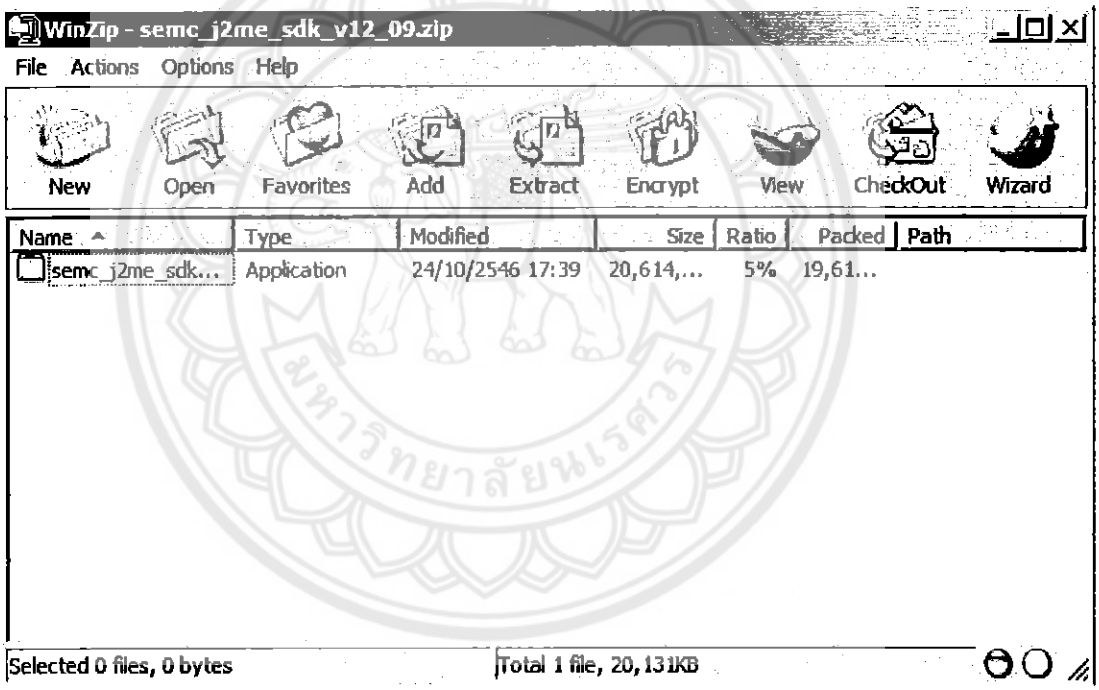
[5] วีระศักดิ์ ซึงถาวร. Java Programming Volume III. กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน). 2547.

ภาคผนวก

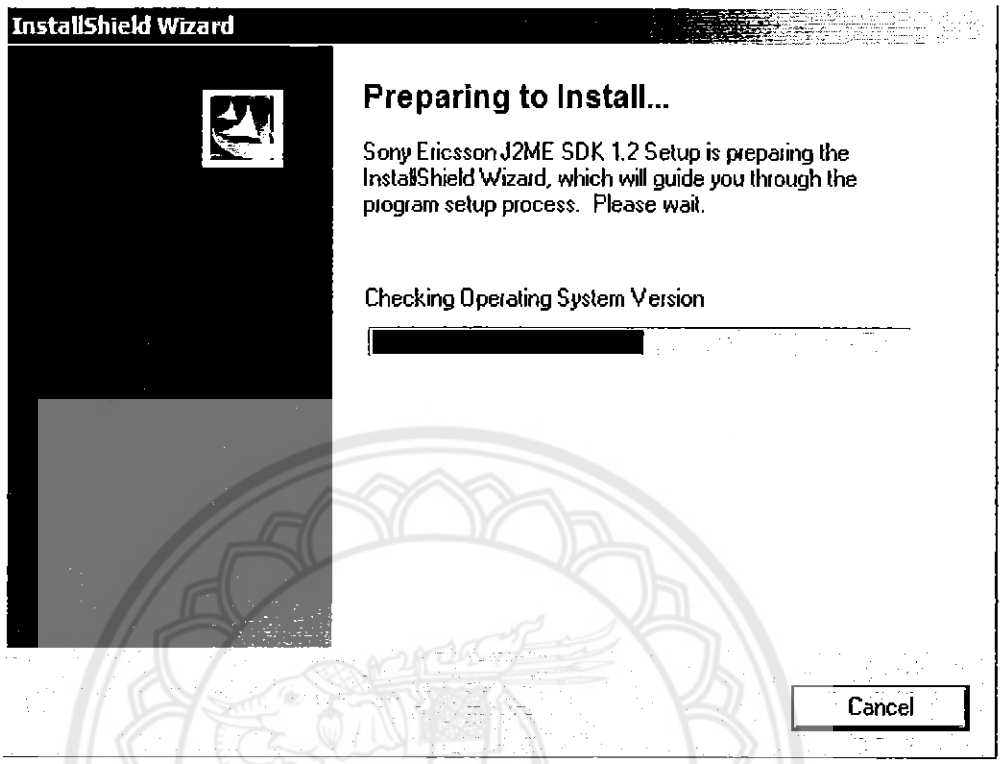
การติดตั้งโปรแกรมของตัวจำลองของโทรศัพท์มือถือมือถือ (Emulator)

ในการพัฒนาโปรแกรมประยุกต์ (Application) บนโทรศัพท์มือถือนั้น จะต้องมีการทดสอบโปรแกรมที่เขียนขึ้นซึ่งการทดสอบไม่สามารถทำบนโทรศัพท์มือถือได้โดยตรง ดังนั้น เพื่อความสะดวกและรวดเร็วในการพัฒนา คณะผู้จัดทำจึงนำตัวจำลองของโทรศัพท์มือถือมาใช้เป็นตัวจำลองของโทรศัพท์มือถือยี่ห้อโซนี่อิริคสัน รุ่น T610 สามารถทำงานได้ในระบบปฏิบัติการวินโดว์ (Windows) ซึ่งมีวิธีการติดตั้งโปรแกรม ดังนี้

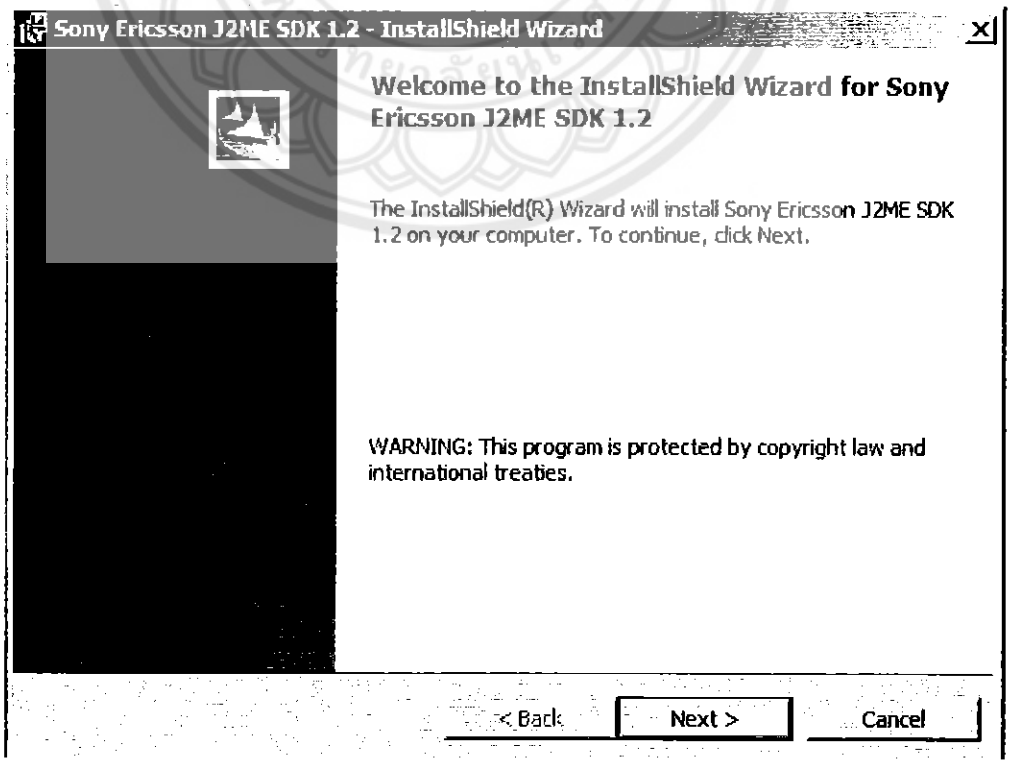
1. ดาวน์โหลด (Download) โปรแกรมที่ใช้ในการติดตั้งตัวจำลองของโทรศัพท์มือถือ จากเว็บไซต์ www.sonyericson.com โปรแกรมที่ใช้ในการติดตั้งที่ได้จะต้องใช้ WinZIP เปิดขึ้นมา



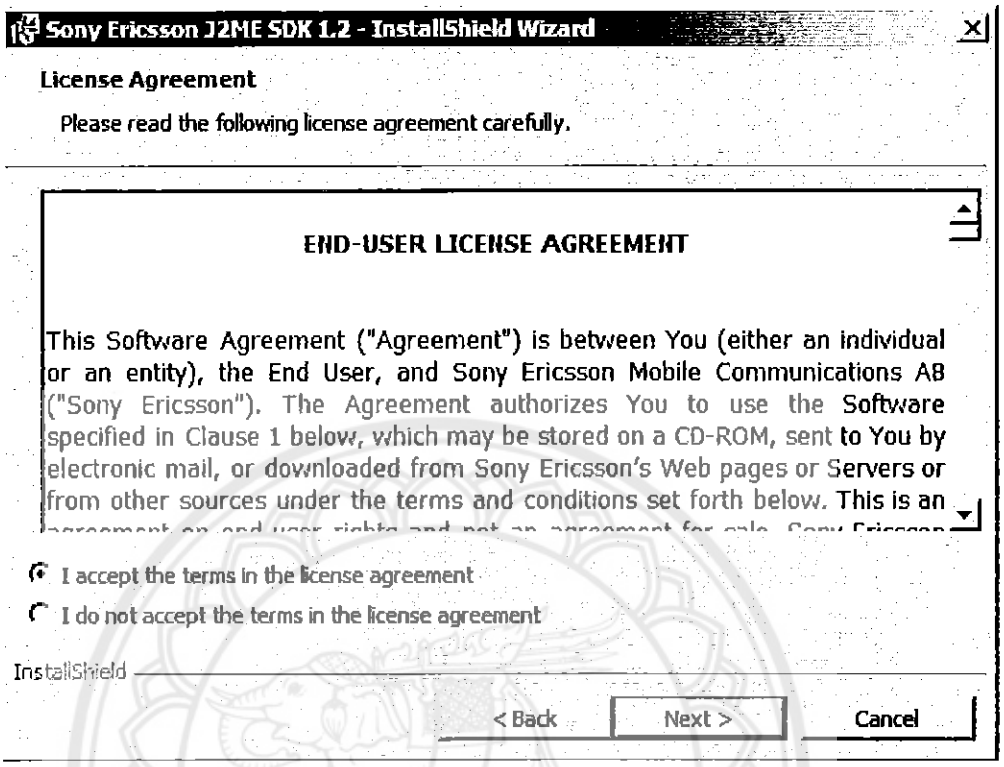
2. ดับเบิลคลิกไฟล์ที่มีชื่อว่า semc_j2me_sdk_v12_09 ที่หน้าจอจะปรากฏ ดังรูป



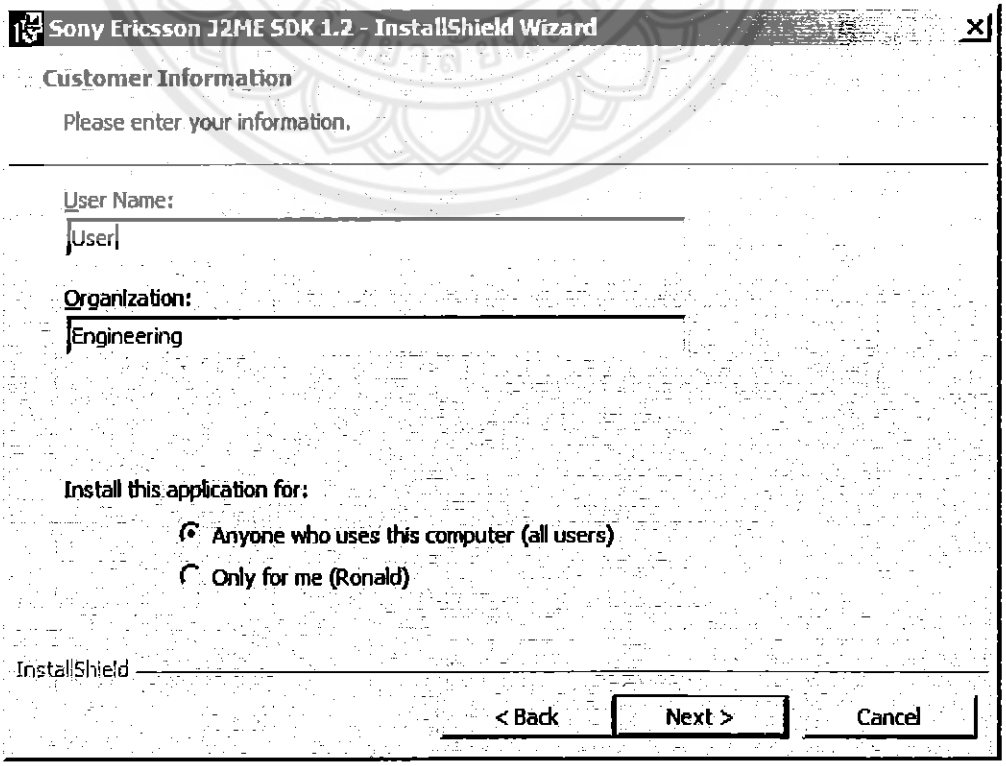
3. คลิก Next > เพื่อดำเนินการต่อไป



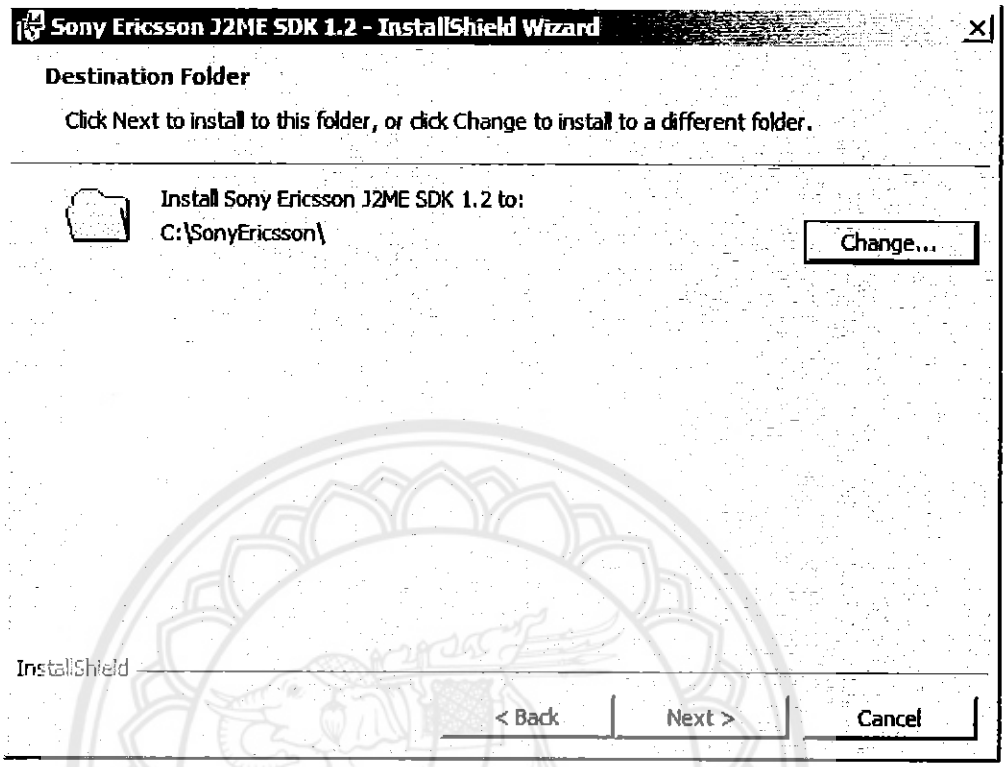
4. เลือกที่ I accept the terms in the license agreement เพื่อยอมรับข้อตกลงแล้วคลิก Next >



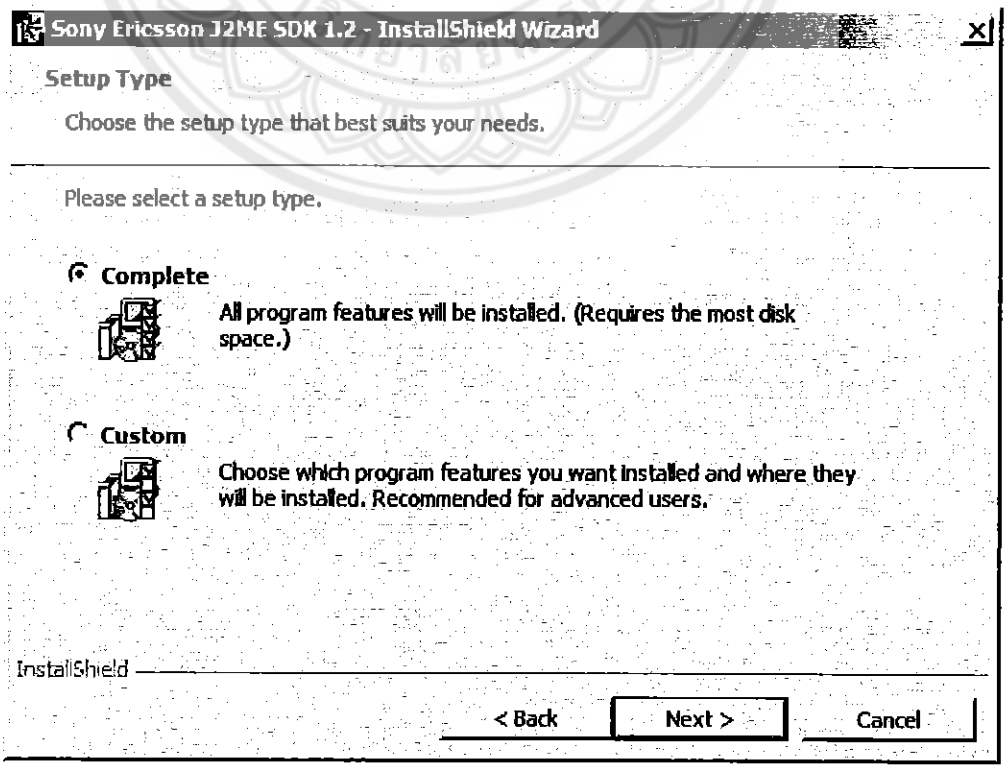
5. ใส่ User Name และ Organization แล้วคลิก Next >



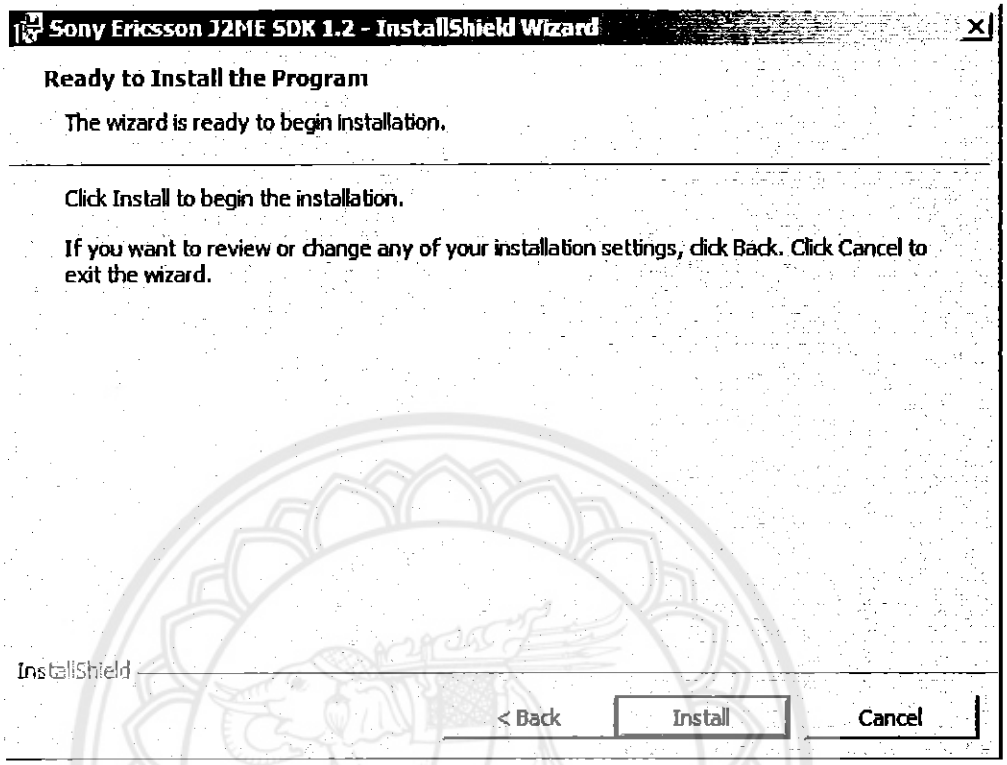
6. เลือกไดเร็กทอรีที่ต้องการเก็บสิ่งที่ติดตั้ง แล้วคลิก Next >



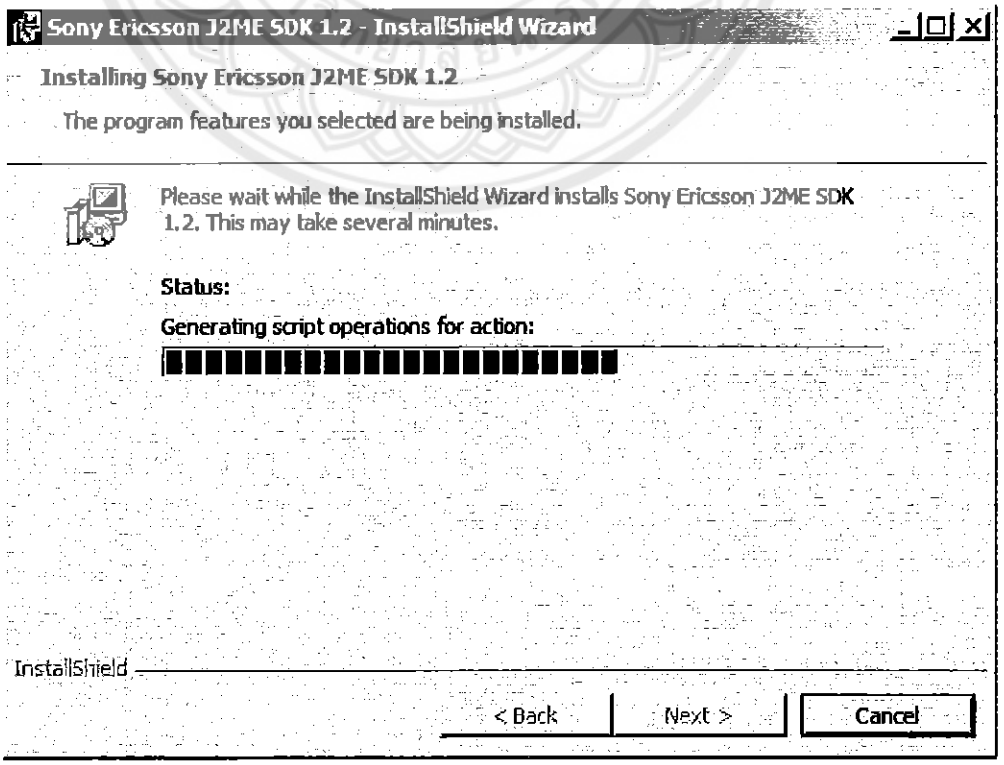
7. เลือกรูปแบบการติดตั้ง แล้วคลิก Next >



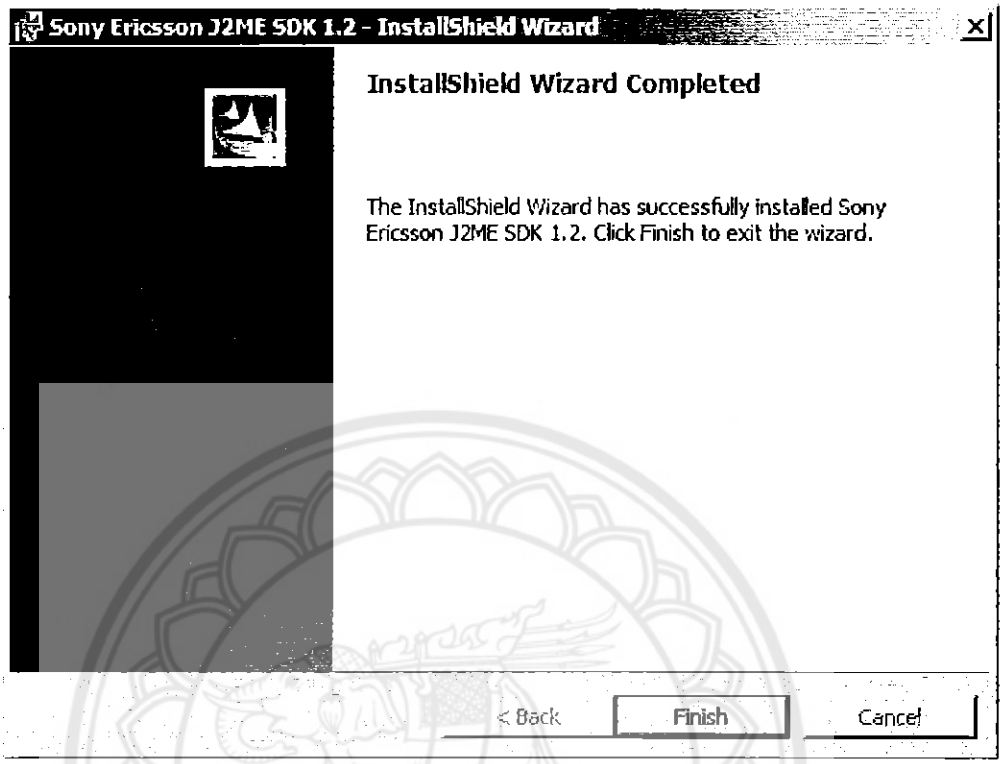
8. คลิกที่ Install เพื่อทำการติดตั้ง



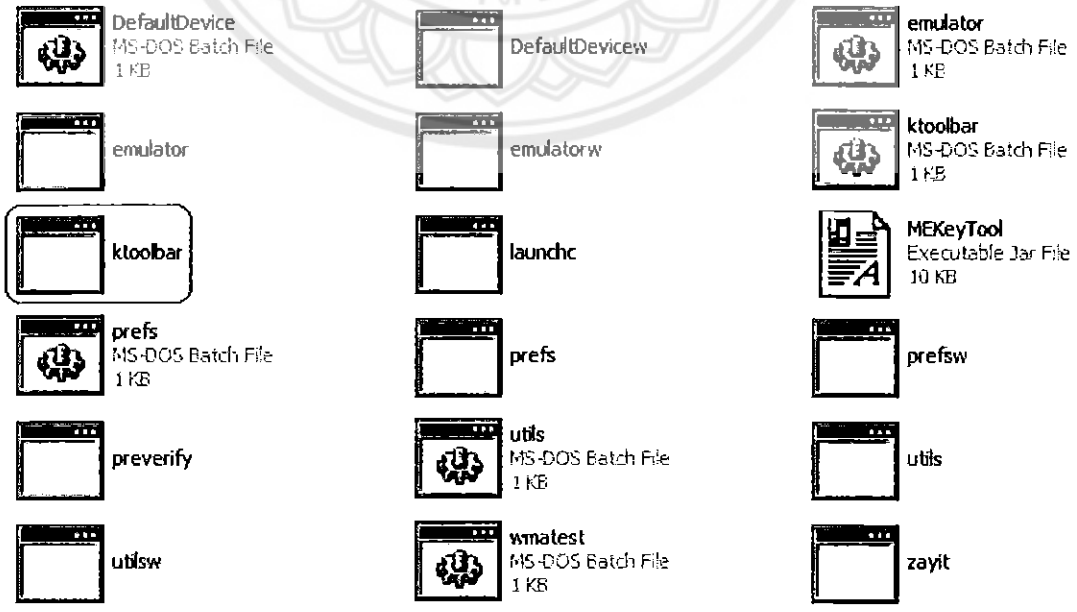
9. กระบวนการติดตั้งซึ่งใช้เวลาประมาณ 1 นาที



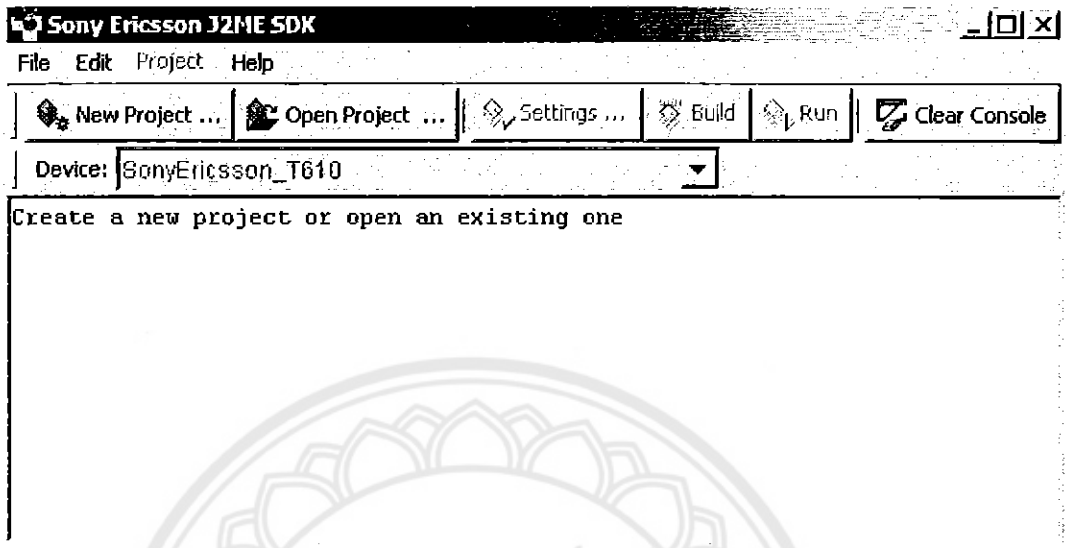
10. คลิกที่ Finish เมื่อเสร็จขั้นตอนการติดตั้ง



เมื่อเสร็จขั้นตอนการติดตั้งสามารถเปิดโปรแกรมตัวจำลองของโทรศัพท์มือถือได้จาก C:\SonyEricsson\J2ME_SDK_11\WTK104_01\bin ที่มีชื่อว่า KToolbar



โปรแกรมจำลองของโทรศัพท์มือถือที่ใช้เป็นเครื่องมือในการทดสอบการทำงานของเกมออนไลน์บนโทรศัพท์มือถือ สามารถทำงานได้บนระบบปฏิบัติการวินโดว์ แสดงดังรูป



ประวัติผู้เขียนโครงงาน

ชื่อ นายธวัชชัย จันทร์พุฒ

ภูมิลำเนา 296 หมู่ 13 ต.บ้านกร่าง อ.เมือง จ.พิษณุโลก 65000

ประวัติการศึกษา

- จบระดับมัธยมศึกษา โรงเรียนพิษณุโลกพิทยาคม
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : twcdo_del@hotmail.com

ชื่อ นายสุริยพงศ์ อัมพวา

ภูมิลำเนา 210/9 ถนนบรมอาสน์ ต.ท่าอิฐ อ.เมือง จ.อุตรดิตถ์ 53000

ประวัติการศึกษา

- จบระดับมัธยมศึกษา โรงเรียนวินิตศึกษา จังหวัดลพบุรี
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : juicyboy@hotmail.com