

การพัฒนาเกมส์บนมือถือ
GAME DEVELOPMENT ON MOBILE PHONE



นายปริญญา แสงพันธุ์ตา

รฟว.

ร/458 ก

2547

เลขหมู่.....
เลขทะเบียน 61379
วัน,เดือน,ปี 17 ก.ค. 2549

b. 11๕๑๕๕๒๘
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาเกมส่นมือถือ
GAME DEVELOPMENT ON MOBILE PHONE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2547

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาเกมส์บนมือถือ

GAME DEVELOPMENT ON MOBILE PHONE

คณะผู้จัดทำ นายปริญญา แสงพันธุตา รหัส 43010259



.....อาจารย์ที่ปรึกษา
(ดร. วรวัฒน์ ถิมโกคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาเกมบนมือถือ

นายปริญญา แสงพันธุตา 43010259
 ดร. วรวัฒน์ ลิ้มโกคา อาจารย์ที่ปรึกษา
 ปีการศึกษา 2547

บทคัดย่อ

การพัฒนาเกมบนมือถือ มีลักษณะเป็นเกม 2 มิติ เป็นการพัฒนาโปรแกรมเพื่อความบันเทิง โดยใช้
 ภาษา J2ME (Java 2 Micro Edition)

ลักษณะของเกมเป็นเกมแนวขานชิง โดยที่ตัวละครของผู้เล่นจะต้องยิงศัตรูเพื่อเก็บคะแนน และต้อง
 คอยระวังไม่ให้ศัตรูมาโดนตัวละคร

โดยเกมนี้นั้นนอกจากจะให้ความบันเทิงและผ่อนคลายกับผู้เล่นแล้ว ยังทำให้ผู้เล่นได้ฝึกประสาท
 สัมผัสอีกด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GAME DEVELOPMENT ON MOBILE PHONE

Mr. Parinya Sangpantuta 43010259

Dr. Voravat Limpoka Advisor

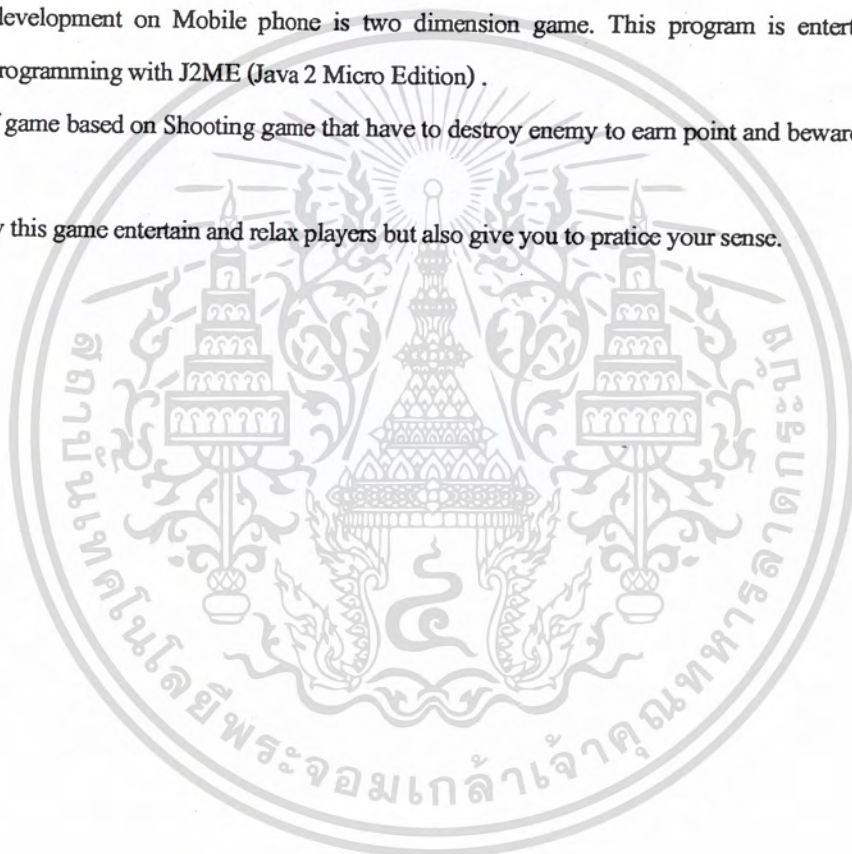
Academic Year 2004

ABSTRACT

Game development on Mobile phone is two dimension game. This program is entertainment program which programming with J2ME (Java 2 Micro Edition) .

Story of game based on Shooting game that have to destroy enemy to earn point and beware enemy attack you too.

Not only this game entertain and relax players but also give you to practice your sense.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สามารถทำได้สำเร็จสมบูรณ์อันเนื่องมาจาก คำแนะนำและคำปรึกษาจากหลายๆฝ่ายด้วยกัน โดยเฉพาะอาจารย์ที่ปรึกษาที่ให้โอกาสให้ข้าพเจ้าได้ทำปริญญานิพนธ์ฉบับนี้ ขอให้ความเอาใจใส่ตลอดจนดูแลสอบถามให้ความช่วยเหลือเสมอมา นั่นก็คือ คร. วรวัฒน์ ถิมโกคา ซึ่งต้องขอบพระคุณเป็นอย่างสูง

รวมทั้งขอขอบคุณทางภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้จัดเตรียมถึงอำนวยความสะดวก เพื่อให้การวิจัยและการพัฒนาโปรแกรมเป็นไปด้วยความสะดวกสบาย อีกทั้งยังมีอินเทอร์เน็ตให้ค้นคว้าหาความรู้ซึ่งจากความสะดวกทั้งหลายเหล่านี้จึงเป็นที่มาของความสำเร็จของโปรเจกต์ที่ปรากฏอยู่

ขอขอบคุณทางห้องโถงล่าที่ให้ที่ไว้ใช้ค้นคว้าและพ่อนคลาย รวมถึงเพื่อนๆที่ๆน้องๆที่ให้ความช่วยเหลือและคำแนะนำต่างๆที่ดีๆขอบคุณครับ

จนมาถึงสุดท้ายที่จะขาดไม่ได้ก็คือ บิดา มารดา ของข้าพเจ้าที่ให้กำเนิดข้าพเจ้ามาและการให้ความเลี้ยงดูที่อบอุ่น ความดูแลเอาใจใส่ รวมถึงโอกาสทางการศึกษาต่างๆที่มอบให้แก่ข้าพเจ้า ข้าพเจ้าจึงอยากขอขอบพระคุณไว้ ณ ที่นี้ด้วย

ปริญญญา แสงพันธุตา

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติประกาศ	III
สารบัญ	IV
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	2
1.4 ขอบเขตของการพัฒนา	2
บทที่ 2 จาวาสำหรับอุปกรณ์ขนาดเล็กและอุปกรณ์ไร้สาย	3
2.1 บทนำ	3
2.2 ภาพรวมของจาวา	3
2.3 J2ME คืออะไร	3
2.3.1 เลขอร์จาวาเวอร์ซอลแมชชีน	4
2.3.2 เลขอร์คอนฟิเจอร์ชัน	4
2.3.3 เลขอร์โพรไฟล์	4
2.4 คอนฟิเจอร์ชันใน J2ME	4
2.4.1 CDC	4
2.4.2 CLDC	5
2.5 โพรไฟล์ของ J2ME	6
2.6 J2ME สำหรับอุปกรณ์ไร้สาย	6
2.7 ความต้องการของระบบ	8
บทที่ 3 หลักการเขียนโปรแกรมบนอุปกรณ์ไร้สายด้วยจาวา	9
3.1 บทนำ	9
3.2 ไลบารีของ CLDC	9
3.2.1 วัชคลาสของ J2SE	9
3.2.2 คลาสที่มีเฉพาะใน CLDC	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ประเภทของข้อมูลพื้นฐานของ CLDC	10
3.3 โลกบาริของ MIDP	10
3.3.1 คลาสโปรแกรมจัดการแอปพลิเคชัน	10
3.3.2 คลาสของส่วนติดต่อกับกราฟิกกับผู้ใช้ (GUI Class)	10
3.3.3 คลาสของพื้นที่เก็บข้อมูลแบบคงตัว (Persistent Storage Class)	11
3.3.4 คลาสของเครือข่าย	11
3.4 MIDlet	11
3.4.1 วงจรการทำงานของ MIDlet	12
3.5 ซอฟต์แวร์จัดการแอปพลิเคชัน (Application Management Software)	13
3.6 ข้อกำหนดต่าง	14
3.6.1 ข้อกำหนดในการจัดการข้อผิดพลาด CLDC	14
3.6.2 ข้อกำหนดของ KVM	14
3.7 ระบบความปลอดภัยใน J2ME	14
3.8 การเวอริฟายและฟรีเวอริฟายไฟล์คลาส	15
บทที่ 4 คอมโพเนนต์หลักของส่วนติดต่อกับผู้ใช้ใน J2ME	16
4.1 บทนำ	16
4.2 Displayable และ Display	16
4.2.1 Displayable	17
4.2.1.1 Screen	17
4.2.1.2 Canvas	18
4.2.2 Display	18
4.3 รูปภาพ (Image)	19
4.3.1 รูปที่แก้ไขไม่ได้	19
4.3.2 รูปที่แก้ไขได้	19
4.4 อีเวนต์และการจัดการอีเวนต์	20
บทที่ 5 การใช้ API ระดับสูงของ J2ME	21
5.1 บทนำ	21
5.2 List และ Choice	21
5.2.1 List	21
5.2.2 Choice	21
5.2.2.1 Exclusive-choice	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2.2 Implicit-choice	21
5.2.2.3 Multiple-choice	21
5.3 Textbox	22
5.4 Alert	23
5.5 Form และ Item	23
5.5.1 Item	23
5.5.1.1 ChoiceGroup	24
5.5.1.2 ImageItem	24
5.5.1.3 StringItem	24
5.5.1.4 Gauge	24
5.5.1.5 DateField	24
5.5.1.6 TextField	24
5.5.2 Form	24
5.6 อีเวนตร์ระดับสูงและการจัดการอีเวนต์	25
5.6.1 Command	25
5.6.2 ItemStateChange	25
บทที่ 6 การออกแบบเกม	26
6.1 การออกแบบลักษณะเกม	26
6.1.1 รูปแบบของเกม	26
6.2 การทำงานของคลาสต่างๆ	26
6.2.1 Galaxy	26
6.2.2 GameThread	26
6.2.3 GalaxyCanvas	26
6.2.4 GalaxyManager	27
6.2.5 Ship,Enemy,Bullet	27
6.2.6 Background	27
6.3 เทคนิคต่างๆที่นำมาใช้ในเกมน	27
6.3.1 เทคนิคการวาดแผนที่	27
6.3.2 เทคนิคการทำภาพแอนิเมชันของตัวละคร	28
6.4 โปรแกรมที่ใช้ในการพัฒนา	28
บทที่ 7 การทดสอบการทำงาน	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.1 ผลการทดสอบ	31
บทที่ 8 บทวิจารณ์และสรุป	35
12.1 สรุปผลการทดลอง	35
12.2 แนวทางในการพัฒนาต่อ	35
บรรณานุกรม	36
ภาคผนวก	37



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่	หน้าที่
รูปที่ 2.1 ความสัมพันธ์ระหว่าง J2SE และ คลาสไลบรารีใน CDC และ CLDC	5
รูปที่ 2.2 โครงสร้างของ MIDP	7
รูปที่ 3.1 วงจรการทำงานของ MIDlet	12
รูปที่ 4.1 โครงสร้างลำดับชั้นของคลาส Displayable	17
รูปที่ 5.1 List แบบ Implicit- choice	22
รูปที่ 5.2 List แบบ exclusive- choice	22
รูปที่ 5.3 List แบบ multiple – choice	22
รูปที่ 5.4 โครงสร้างลำดับชั้นของคลาส Item	24
รูปที่ 6.3 ภาพในแต่ละเฟรม	28
รูปที่ 6.7 โปรแกรม Adobe PhotoShop	29
รูปที่ 6.8 โปรแกรม J2ME Wireless Toolkit	29
รูปที่ 6.9 อุปกรณ์จำลอง	30
รูปที่ 7.1 หน้าจอเมื่อเริ่มต้นเลือกเข้าสู่เกม	31
รูปที่ 7.2 หน้าจอเมื่อเริ่มต้นเข้าสู่เกม	32
รูปที่ 7.3 หน้าจอเมื่อเลือก Go เพื่อเล่น	32
รูปที่ 7.4 หน้าจอเมื่อกดยิง	33
รูปที่ 7.5 หน้าจอเมื่อมีศัตรู	33
รูปที่ 7.6 หน้าจอเมื่อจบเกม	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1. หลักการและเหตุผล

เนื่องจากในปัจจุบันนี้การพัฒนาซอฟต์แวร์นั้นได้มีการพัฒนาในรูปแบบด้านต่างๆมากมายเช่นด้านโปรแกรมเพื่อการใช้งาน ด้านการติดต่อสื่อสาร ด้านโปรแกรมประเภทกราฟิกและแอนิเมชัน และในด้านความบันเทิง ซึ่งการพัฒนานั้นจะเกิดขึ้นบนเครื่องคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยความจำที่มากและส่วนประมวลผลที่มีความเร็วสูง

ดังนั้นจึงต้องการจะนำความสามารถต่างๆเหล่านั้นมาทำงานอยู่บนอุปกรณ์ที่มีขนาดเล็กซึ่งจำกัดไปด้วยหน่วยความจำที่น้อยและความเร็วของส่วนประมวลผลที่ยังขึ้นอยู่กับข้อดีที่ว่าสามารถเคลื่อนที่ได้สะดวกพกพาได้ง่ายเนื่องจากมีขนาดเล็ก ซึ่งอุปกรณ์ที่จะเลือกพัฒนาในที่นี้คือ มือถือนั่นเอง

ซึ่งซอฟต์แวร์ที่ต้องการจะพัฒนาก็คือ ด้านความบันเทิง เหตุผลที่ต้องเป็นเพื่อความบันเทิงนั่นคือ มือถือนั้นเป็นอุปกรณ์ที่ต้องพกพาไปไหนมาไหนด้วยตลอดดังนั้น เมื่อมีเวลาว่างผู้เป็นเจ้าของมือถือก็สามารถที่จะผ่อนคลายได้ รวมไปถึงการพัฒนาซอฟต์แวร์ประเภทนี้นั้นสามารถใช้เป็นพื้นฐานขั้นต้นในการที่จะพัฒนาซอฟต์แวร์ขั้นสูงต่อไป

2. วัตถุประสงค์

1. เพื่อศึกษาการทำงาน โปรแกรมและการพัฒนาโปรแกรมด้วย J2ME บน โทรศัพท์มือถือ
2. เพื่อฝึกทักษะในการเขียน โปรแกรมในด้านอัลกอริทึม
3. เพื่อพัฒนาบุคลากรทางด้านอุตสาหกรรมซอฟต์แวร์
4. เพื่อศึกษาการเขียนโปรแกรมด้วย J2ME เพื่อนำไปใช้ในการพัฒนาเกมและ โปรแกรมต่างๆ บน อุปกรณ์โทรศัพท์
5. เพื่อรู้จักใช้เวลาว่างให้เกิดประโยชน์ ทั้งทางด้านความรู้ และ การทำงานเป็นกลุ่มมีความรับผิดชอบ และรู้จักการทำงานอย่างเป็นระบบ
6. เพื่อนำความรู้ที่ได้จากการเขียนเกมมาประยุกต์ให้เข้ากับการเขียน โปรแกรมประเภทอื่นๆ ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ประโยชน์ที่คาดว่าจะได้รับ

ส่วนของโปรแกรมในส่วนของ Low-Level API อินเทอร์เฟซระดับต่ำเป็นซึ่งเป็นส่วนหลักของเกม ในการทำ animation ต่างๆของเกมซึ่งในส่วนนี้จะสามารถนำไปประยุกต์ต่อไปในการที่จะทำด้านกราฟิกซึ่งในส่วนนี้ผู้เขียน application สามารถออกแบบทุกอย่างเองได้ตามใจชอบ

4. ขอบเขตของการพัฒนา

จะพัฒนาในรูปแบบของแอปพลิเคชันในมือถือซึ่งการพัฒนานั้นใช้ภาษา J2ME (Java to Micro Edition) ซึ่งในส่วนนี้จะเป็นส่วนเกิดการเล่นเกมจริงๆและเป็นส่วนที่ติดต่อกับผู้ใช้ซึ่งส่วนแสดงผลคือหน้าจอ มือถือนั่นเอง และส่วนที่ทำการติดต่อเพื่อรับคำสั่งจากผู้ใช้คือ ปุ่มกดต่างๆบนเครื่องโทรศัพท์นั่นเอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

จาวาสำหรับอุปกรณ์ขนาดเล็กและอุปกรณ์ไร้สาย

2.1 บทนำ

หลังจากที่บริษัทซัน ไมโครซิสเต็มส์ได้ออก จาวา 2 ไมโครอี디션 (Java 2 Micro Edition : J2ME) ซึ่งมีความยืดหยุ่นสูง เพื่อใช้พัฒนาแอปพลิเคชันรุ่นใหม่สำหรับอุปกรณ์อิเล็กทรอนิกส์และอุปกรณ์เคลื่อนที่ขนาดเล็กซึ่งส่วนมากได้แก่ โทรศัพท์เคลื่อนที่และเพจเจอร์รับส่งข้อมูล

2.2 ภาพรวมของจาวา

ภาษาจาวาได้กลายเป็นแพลตฟอร์มการพัฒนาแอปพลิเคชันแบบเชิงวัตถุ (Object-oriented) ในอุปกรณ์และงานต่างๆ อย่างเต็มตัว นับตั้งแต่แอปพลิเคชันเซิร์ฟเวอร์ระดับองค์กร เครื่องคอมพิวเตอร์แบบตั้งโต๊ะทั่วไป จนถึงแอปพลิเคชันฝังตัวสำหรับอุปกรณ์ขนาดเล็กอีกด้วย

แพลตฟอร์มจาวา 2 ที่ใช้งานอยู่ในขณะนี้ มี 3 รุ่นด้วยกัน โดยที่แต่ละรุ่นนั้นก็จะเป็นนำไปใช้กับเฉพาะกลุ่มแอปพลิเคชันดังนี้

- จาวา 2 เอ็นเทอร์ไพรส์อี디션 (Java 2 Enterprise Edition : J2EE) ใช้งานกับแอปพลิเคชันบนเครื่องเซิร์ฟเวอร์สำหรับองค์กรที่รองรับระบบงานใหญ่ๆ และไคลเอนต์จำนวนมาก
- จาวา 2 สแตนดาร์ดอี디션 (Java 2 Standard Edition : J2SE) ใช้งานกับแอปพลิเคชันบนเครื่องคอมพิวเตอร์แบบตั้งโต๊ะทั่วไป
- จาวา 2 ไมโครอี디션 (Java 2 Micro Edition : J2ME) ใช้งานกับแอปพลิเคชันรุ่นใหม่ซึ่งเน้นกลุ่มผู้ใช้อุปกรณ์อิเล็กทรอนิกส์และอุปกรณ์ฝังตัว (Embedded)

2.3 J2ME คืออะไร

J2ME เป็นแพลตฟอร์มจาวาที่ออกแบบมาเพื่อใช้กับแอปพลิเคชันที่ทำงานบนอุปกรณ์ขนาดเล็กเช่น โทรศัพท์มือถือ พีดีเอ โทรศัพท์พร้อมจอภาพที่ต่อกับอินเทอร์เน็ต โทรศัพท์ดิจิตอลขนาดเล็ก อุปกรณ์บันเทิง และระบบนำทางในรถยนต์ สวิตช์ในระบบเครือข่าย คอมพิวเตอร์ของเครื่องอำนวยความสะดวกภายในบ้าน ฯลฯ โดย J2ME ได้นำโครงสร้างแบบโมดูลที่มีความยืดหยุ่นสูงเข้ามาใช้ เพื่อให้สามารถสนับสนุนการทำงานอุปกรณ์หลากหลายประเภท J2ME กำหนดชั้นของซอฟต์แวร์ (Software) ไว้ 3 เลเยอร์ (Layer) ด้วยกัน โดยเลเยอร์ทั้งหมดจะอยู่เหนือชั้นระบบปฏิบัติการของอุปกรณ์ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 เลเยอร์จาวาเวอร์ชวลแมชชีน (Java Virtual Machine) เป็นเลเยอร์ของจาวาเวอร์ชวลแมชชีนปรับแต่งให้เข้ากับระบบปฏิบัติการของอุปกรณ์ และรองรับแต่ละ คอนฟิกูเรชัน (Configuration) ของJ2ME และเวอร์ชวลแมชชีน ของ J2MEได้แก่ (Connected Virtual Machine : CVM) และ (KilobyteVirtual Machine : KVM) ทั้ง CVM และ KVM ต่างสนับสนุนยูทิลิตี้ JavaCodeCompact หรือคลาส prelinker preloader และ ROmnizer ยูทิลิตี้นี้จะโยง จาวาคลาสเข้ากับเวอร์ชวลแมชชีน ช่วยลดระยะเวลาเริ่มใช้งาน VM ได้มากขึ้น ตัวอย่างเช่น เรียก KVM และ CLDC ขึ้นมาก่อนโหลดไปยังหน่วยความจำแบบROM ของอุปกรณ์ คลาส CLDC จะถูกโยงเข้าไปใน KVM โดยตรง เพื่อปรับปรุงประสิทธิภาพการทำงานเราเรียกขั้นตอนนี้ว่าการโหลดเข้าหน่วยความจำแบบ ROM (ROMized)

2.3.2 เลเยอร์คอนฟิกูเรชัน เป็นเลเยอร์ของคอนฟิกูเรชันของJ2ME ซึ่งกำหนดคลาสไลบรารีอุปกรณ์ ใช้งานทั่วไปหรือกลุ่มอุปกรณ์ที่มีความต้องการหน่วยความจำและหน่วยประมวลผลใกล้เคียงกันคอนฟิกูเรชันในจาวาสำหรับอุปกรณ์ขนาดเล็ก มี 2 ประเภทด้วยกัน คือ CDC (Connected DeviceConfiguration) และ CLDC (Connected Limited Device Configuration)

2.3.3 เลเยอร์โพรไฟล์ (Profile) เป็นเลเยอร์ที่สร้างเหนือเลเยอร์คอนฟิกูเรชัน โดยกำหนดคลาสไลบรารีเพื่อสนองตอบความต้องการของตลาดเฉพาะกลุ่ม ตัวอย่างของโพรไฟล์ในเลเยอร์นี้ได้แก่ PDA Profile , MID Profile , Foundation Profile และ Personal Profile

2.4 คอนฟิกูเรชันใน J2ME

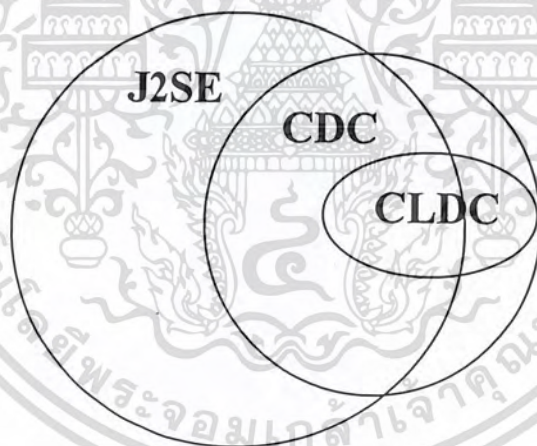
คอนฟิกูเรชันและโพรไฟล์เป็นองค์ประกอบหลักของ J2ME โดยมีจุดสำคัญ คือ เพื่อปรับแต่งเวอร์ชวลแมชชีน และคลาสไลบรารีให้เหมาะสมกับอุปกรณ์แต่ละประเภท คอนฟิกูเรชัน คือ ชุดที่มีคุณสมบัติขั้นต่ำของ จาวาเวอร์ชวลแมชชีน และจาวาคลาสสำหรับอุปกรณ์แต่ละประเภท เป็นตัวแทนของอุปกรณ์ที่มีลักษณะคล้ายคลึงกัน อาจกล่าวได้ว่า คอนฟิกูเรชันเป็นตัวกำหนดคุณสมบัติหรือไลบรารีขั้นต่ำของแพลตฟอร์มจาวาที่นักพัฒนาคาดว่าจะต้องมีในทุกอุปกรณ์ ขณะที่คลาสไลบรารีที่กำหนดในคอนฟิกูเรชันจะมีในทุกอุปกรณ์ที่อยู่ในกลุ่มเดียวกัน

ปัจจุบันคอนฟิกูเรชันใน J2ME แบ่งเป็น 2 ประเภทคือ CDC (Connected Device Configuration) และ CLDC (Connected Limited Device Configuration) ดังนี้

2.4.1 CDC เป็นอุปกรณ์ใช้ร่วมกับอุปกรณ์อื่น คิดตั้งตายตัว และใช้เชื่อมต่อข้อมูล โดยปรกติมักมีส่วนติดต่อกับผู้ใช้หลากหลายแบบ มีหน่วยความจำประมาณ 2-16 เม็กกะไบต์ ใช้หน่วยประมวลผลแบบ32 บิตหรือมากกว่า เชื่อมต่อเครือข่ายที่มีแบนด์วิดท์ (Bandwidch) สูงอย่างต่อเนื่องโดยอาศัยพอร์ต TCP/IP ตัวอย่างอุปกรณ์ประเภทนี้ ได้แก่ โทรศัพท์มือถือขนาดเล็ก อินเทอร์เน็ต

เน็ตทีวี โทรศัพท์พร้อมจอภาพที่เชื่อมต่อกับอินเทอร์เน็ต อุปกรณ์สื่อสารที่มีความซับซ้อนสูง อุปกรณ์บันทึกและระบบนำทางในรถยนต์

2.4.2 CLDC เป็นอุปกรณ์ส่วนบุคคล พกพาได้ และใช้เชื่อมต่อข้อมูล โดยปรกติมักมีส่วนติดต่อกับผู้ใช้แบบง่ายๆเมื่อเทียบกับระบบบนเครื่องคอมพิวเตอร์ตั้งโต๊ะ มีหน่วยความจำประมาณ 128 กิโลไบต์ – 1 เมกกะไบต์ ใช้หน่วยประมวลผลแบบ 16 หรือ 32 บิต เชื่อมต่อกับเครือข่ายที่มีแบนด์วิดท์ต่ำเป็น ระยะเวลาสั้นๆ โดยไม่อาศัยพอร์ต TCP/IP ตัวอย่างอุปกรณ์ประเภทนี้ ได้แก่ โทรศัพท์มือถือแบบไม่ซับซ้อนมากนัก เพจเจอร์รับส่งข้อความ เครื่องปาล์มโอเอสแบบพกพา ในชั้นของคอนฟิгурชัน มีคลาส 2 ประเภทด้วยกัน คือ คลาสที่นำมาจาก J2SE และที่ออกแบบเฉพาะ อุปกรณ์ขนาดเล็ก คลาสที่นำมาจาก J2SE จะมีคุณสมบัติอย่างเดียวกันกับคลาสใน J2SE หรือเป็น ชั้นคลาสของ J2SE เช่น แพคเกจ `java.io` และ `java.util` จากรูป 2.1 แสดงความสัมพันธ์ระหว่าง J2SE และ คลาสไลบรารีของ CDC และ CLDC จากภาพจะเห็นว่าคลาสใน CLDC ส่วนใหญ่ทำงานร่วมกับ คลาสใน CDC ได้ดีเช่นเดียวกับกรณีของคลาสใน CLDC และคลาสใน J2SE



รูปที่ 2.1 ความสัมพันธ์ระหว่าง J2SE และ คลาสไลบรารีใน CDC และ CLDC

คลาสที่ไม่ได้นำมาจาก J2SE และออกแบบเพื่อใช้เฉพาะอุปกรณ์มักทำงานร่วมกับ J2SE ได้ไม่ค่อยดีใน CLDC คลาสเหล่านี้จะอยู่ในกลุ่มกรอบการติดต่อสื่อสารทั่วไป (Generic Connection Framework) โดยระบุไว้ในแพคเกจ `javax.microedition.io`

คอนฟิгурชันยังระบุคุณสมบัติของจาวาเวอร์ชวลแมชชีน ในเลขอร์ด้านล่างอีกด้วย ในโครงสร้างปัจจุบัน CDC และ CLDC มีเวอร์ชวลแมชชีนที่ปรับแต่งมาเฉพาะตัวอยู่แล้ว เวอร์ชวลแมชชีนของ CDC คือ C Virtual Machine (CVM) มีคุณสมบัติครบถ้วนเหมือน Java 2 Virtual Machine แต่ขนาดเล็กกว่าออกแบบมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับอุปกรณ์ที่สลับซับซ้อน CVM มีความต้องการหน่วยความจำ 256 กิโลไบต์ ขณะที่หน่วยความจำแบบ ROM ของ CDC มีขนาด 1 เมกกะไบต์ เวอร์ชันของ CLDC คือ K Virtual Machine (KVM) แม้จะมีขนาดเล็กแต่มีความสามารถในการทำงานสูง ออกแบบมาเพื่อใช้งานกับอุปกรณ์ที่มีทรัพยากรจำกัด K ใน KVM หมายถึง กิโล โดยเรียกตามหน่วยความจำที่นับเป็นกิโลไบต์ ส่วนคอมพิวเตอร์แบบตั้งโต๊ะนั้นหน่วยนับเป็นเมกกะไบต์ KVM เหมาะจะนำมาใช้กับหน่วยประมวลผล RISC/CISC แบบ 16/32 บิต ซึ่งมีหน่วยความจำทั้งหมดไม่กี่ร้อยกิโลไบต์เท่านั้นประมาณ 128 กิโลไบต์ ปัจจุบัน KVM มีความต้องการหน่วยความจำอยู่ในช่วง 40 – 80 กิโลไบต์

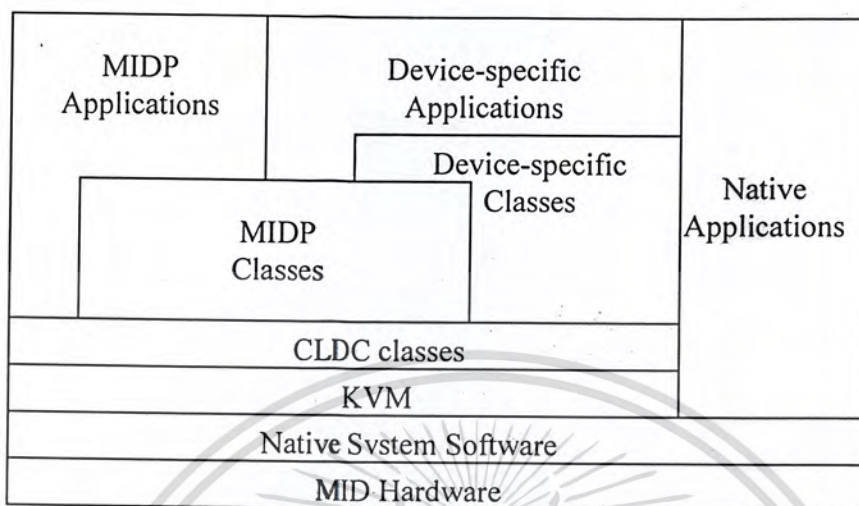
2.5 โพรไฟล์ของ J2ME

โพรไฟล์จะกำหนดชุดของ API ที่ต้องใช้เพิ่มเติม ตลอดจนกำหนดคุณสมบัติที่เป็นที่ต้องการของตลาดเฉพาะกลุ่มหรืออุปกรณ์เฉพาะประเภท คลาสไลบรารีในโพรไฟล์ช่วยให้นักพัฒนาสร้างฟังก์ชันเฉพาะอุปกรณ์ เช่น ส่วนติดต่อกราฟิกกับผู้ใช้การเชื่อมต่อเครือข่าย หน่วยเก็บข้อมูล Persistent Storage ฯลฯ ตามปรกติแล้วจะไม่สามารถนำคลาสไลบรารีที่สร้างเพื่อใช้งานกับโพรไฟล์หนึ่งไปใช้กับโพรไฟล์อื่นได้ ขณะนี้ มีบางโพรไฟล์ได้ถูกกำหนดแล้ว แต่ก็มีหลายโพรไฟล์ที่ยังอยู่ในขั้นตอนการดำเนินการอยู่ โพรไฟล์ Foundation และ RMI ซึ่งสร้างบน CDC กำลังเผยแพร่อยู่ ส่วนที่สร้างบน CLDC มีโพรไฟล์เดียวคือ MID Profile (Mobile Information Device Profile : MIDP) มันจะเตรียมส่วนติดต่อผู้ใช้และหน่วยเก็บข้อมูล Persistent Storage ความสามารถด้านเครือข่าย แบบจำลอง API สำหรับแอปพลิเคชัน ไว้ให้อุปกรณ์ไร้สาย เช่น โทรศัพท์ที่ไม่มีซับซ้อนมากนักและเพจเจอร์รับส่งข้อความ ส่วนโพรไฟล์ พีดีเอ (PDAP) กำลังได้รับการวิจัยอยู่

อุปกรณ์หนึ่งๆอาจมีโพรไฟล์ใช้งานมากกว่า 1 ชนิด และบางโพรไฟล์ใช้งานเฉพาะบางอุปกรณ์หรือแอปพลิเคชันเท่านั้น ตัวอย่างเช่น โพรไฟล์บน CDC ส่วนใหญ่ เช่น RMI และ Personal จะสร้างไว้เหนือ Profile Foundation หากไม่มี Profile Foundation และ CDC รองรับแอปพลิเคชันที่เขียนก็ไม่สามารถทำงานได้

2.6 J2ME สำหรับอุปกรณ์ไร้สาย

J2ME ได้ให้กำเนิดแอปพลิเคชันยุคใหม่บนอุปกรณ์ไร้สาย ช่วยให้เกมส์แบบหลายผู้เล่นที่ทำงานผ่านอินเทอร์เน็ตการทำธุรกรรมทางโทรศัพท์มือถือ แอปพลิเคชันสำหรับองค์กรทั้งไคลเอนต์และเซิร์ฟเวอร์เกิดขึ้นได้บนโทรศัพท์มือถือและเพจเจอร์รับส่งข้อความ MIDP CLDC และ KVM ได้กลายมาเป็นรากฐานในการพัฒนาจาวาแอปพลิเคชันบนอุปกรณ์ไร้สายยุคใหม่ มี 3 ปัจจัยในการสร้างแพลตฟอร์มสำหรับแอปพลิเคชันไร้สาย คือ MIDP CLDC KVM



รูปที่ 2.2 โครงสร้างของ MIDP

โครงสร้างของเลเยอร์ต่างๆจากล่างขึ้นบน ดังนี้

1. เลเยอร์ฮาร์ดแวร์ MID หมายถึง โทรศัพท์มือถือ
2. เลเยอร์ซอฟต์แวร์ของระบบที่ติดตั้งมากับอุปกรณ์ หมายถึง ระบบปฏิบัติการและไลบรารีของระบบที่บริษัทผู้ผลิตให้มา
3. เลเยอร์ KVM เป็นส่วนที่เตรียม Runtime Environment ไว้ให้แอปพลิเคชันบนอุปกรณ์ไร้สาย
4. เลเยอร์ CLDC เป็นส่วนที่เตรียม API หลักของจาวาให้แอปพลิเคชันบนอุปกรณ์ไร้สาย
5. เลเยอร์ MIDP เป็นส่วนที่เตรียมไลบรารีสำหรับส่วนติดต่อกราฟิกกับผู้ใช้ และหน่วยเก็บข้อมูล Persistent Storage ระบบเครือข่าย และไทม์เมอร์

นอกจากคลาสไลบรารีสำหรับ MIDP ผู้ผลิตอาจเตรียมคลาสไลบรารีเฉพาะอุปกรณ์ไว้ให้นักพัฒนา เพื่อดึงความสามารถของฟังก์ชันที่มีอยู่แล้วไปใช้งานให้เกิดประโยชน์สูงสุด อาทิ การโทรศัพท์ การแชร์ข้อมูลกับแอปพลิเคชันที่ติดตั้งมาในเครื่อง จำพวก ปฏิทิน สมุดจดที่อยู่ การตรวจสอบข้อมูลอุปกรณ์ เช่น อายุแบตเตอรี่ ความแรงของสัญญาณ ฯลฯ แม้ว่าถ่านแบตเตอรี่เฉพาะอุปกรณ์ที่ผู้ผลิตเตรียมไว้มาใช้งาน จะช่วยเพิ่มความสามารถแก่แอปพลิเคชันบนอุปกรณ์ไร้สาย แต่ไม่สามารถเคลื่อนย้ายไปสู่อุปกรณ์อื่นที่ใช้ MIDP ได้ เนื่องจากคลาสที่นำมาใช้อยู่นอกเหนือขอบเขตของ MIDP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ความต้องการของระบบ

2.7.1 อุปกรณ์ไร้สายจะทำงานสนับสนุน J2ME ได้ดีเมื่อมีคุณสมบัติตรงตามข้อกำหนด หากต้องการให้ KVM ทำงานได้อย่างมีประสิทธิภาพ โดยใช้ไลบรารี CLDC จะต้องมีคุณสมบัติของระบบขั้นต่ำ ดังนี้

- มีหน่วยความจำ 160 – 512 กิโลไบต์ สำหรับสร้างแพลตฟอร์มจาวา
- มีหน่วยประมวลผลแบบ 16-32 บิตความเร็ว 25 เมกกะเฮิร์ซ
- ใช้พลังงาน โดยมากมักทำงานโดยใช้แบตเตอรี่
- เชื่อมต่อกับเครือข่ายได้ในช่วงสั้นๆ อาศัยระบบไร้สายเป็นส่วนใหญ่ แบนด์วิดท์ก่อนข้างจำกัดความเร็ว 9600 ไบต์ต่อวินาทีหรือน้อยกว่า
- มีหน่วยความจำชั่วคราวขนาด 32 กิโลไบต์ สำหรับเก็บจาวา รันไทม์และหน่วยความจำของอ็อบเจกต์

2.7.2 การใช้งาน MIDP ฮาร์ดแวร์จะต้องมีคุณสมบัติดังนี้

2.7.2.1 การแสดงผล

- หน้าจอขนาด 96*54
- ความลึกของสี 1 บิต
- สัดส่วนของภาพ 1:1

2.7.2.2 การรับข้อมูลเข้า

- ใช้กลไกการป้อนข้อมูลอย่างไรอย่างหนึ่ง อาทิ แป้นพิมพ์ หรือจอสัมผัส

2.7.2.3 หน่วยความจำ

- หน่วยความจำถาวรขนาด 128 กิโลไบต์สำหรับเก็บคอมไพเลอร์ของ MIDP
- หน่วยความจำถาวรขนาด 8 กิโลไบต์สำหรับเก็บข้อมูลที่แอปพลิเคชันสร้างขึ้น
- หน่วยความจำชั่วคราวขนาด 32 กิโลไบต์ สำหรับเก็บจาวา รันไทม์ เช่น จาวา ซีป

2.7.2.4 เครือข่าย

- เครือข่ายรับส่งข้อมูลผ่านระบบไร้สาย เชื่อมต่อได้ในช่วงสั้นๆ และมีแบนด์วิดท์

จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการเขียนโปรแกรมบนอุปกรณ์ไร้สายด้วยจาวา

3.1 บทนำ

ในบทนี้จะแสดงให้เห็นภาพรวมของหลักการพัฒนาแอปพลิเคชันบน MIDP ด้วย J2ME โดยชี้ให้เห็นว่าใน MIDP มี จาวาคลาสอะไรบ้าง ตลอดจนแสดงถึงวิธีการเขียน โปรแกรม MIDP เพื่อใช้งานใน MIDP

สภาพแวดล้อมการทำงานของจาวาใน MIDP มักมาพร้อมกับ KVM ไลบารีของ CLDC และ MIDP และซอฟต์แวร์จัดการแอปพลิเคชัน (Application Management Software – AMS) ไลบารีจาวาใน CLDC และ MIDP จะเป็นรากฐานรองรับการเขียนแอปพลิเคชันบนอุปกรณ์ไร้สาย

3.2 ไลบารีของ CLDC

ไลบารีของ CLDC เป็นไลบารีของเครือข่ายและระบบชั้นสูงที่ไม่จำกัดเฉพาะอุปกรณ์ใดอุปกรณ์หนึ่งประกอบด้วย คลาส 2 ประเภท คือ คลาสที่เป็นซัพเซตของ J2SE และคลาสที่เกี่ยวกับการติดต่อสื่อสารทั่วไป

3.2.1 ซับคลาสของ J2SE คลาสประเภทแรกของ CLDC เป็นซับคลาสของไลบารี J2SE กำหนดไว้ในแพ็คเกจ `java.lang.java.util` และ `java.io` โดยคลาสระบบและคลาสชนิดข้อมูลเข้ากันได้กับ J2SE และ J2EE และเพื่อให้เข้ากันได้เคลื่อนย้ายข้ามแพลตฟอร์มได้สะดวก คลาสประเภทนี้จะใช้ชื่อคลาสและชื่อแพ็คเกจตามชื่อคลาสอย่างเดียวกันใน J2SE หรือตามชื่อซับคลาสใน J2SE ที่เกี่ยวเนื่องกัน โดยไม่มีการเพิ่มเมธีด `public` หรือ `protected` หรือฟิลดซึ่งไม่มีในคลาสที่เกี่ยวเนื่องกัน ใน J2SE ความหมายของคลาสและเมธีดจึงไม่เปลี่ยนแปลง

3.2.2 คลาสที่มีเฉพาะใน CLDC คลาสประเภทที่สองของ CLDC กำหนดไว้ในแพ็คเกจ `javax.microedition.io` คลาสเหล่านี้มีเฉพาะใน CLDC จึงไม่สามารถทำงานร่วมกับไลบารีของ J2SE ได้ เป็นคลาสชั้นสูงเกี่ยวกับเรื่องทั่วไปของเครือข่าย กำหนดไว้ในแพ็คเกจ `java.io` และ `java.net` คลาสประเภทนี้จะเตรียมกรอบการติดต่อสื่อสารผ่านเครือข่ายให้กับอุปกรณ์ที่สนับสนุน J2ME โดยกรอบการสื่อสารนี้มักเรียกกันว่า Generic Connection Framework คลาสส่วนใหญ่ได้แก่อินเตอร์เฟสที่แทนการสื่อสารรูปแบบต่างๆ เช่น ซ็อกเก็ต ดาต้าแกรมซีเรียล และ http ขึ้นอยู่กับว่าผู้ผลิตอุปกรณ์หรือผู้ให้บริการเครือข่ายจะติดตั้งอินเตอร์เฟสการเชื่อมต่อเหล่านี้เพียงบางตัวหรือครบทั้งหมด เมื่อพิจารณาจากความสามารถของอุปกรณ์และเครือข่าย

ข้อจำกัดของ CLDC CLDC และ KVM ออกแบบมาเพื่อใช้กับอุปกรณ์ที่มีทรัพยากรจำกัด จึงไม่สามารถสนับสนุนคุณสมบัติและฟังก์ชันได้หลากหลายเท่า J2SE คุณสมบัติบางอย่างที่มีใน J2SE ถูกดึงออก

จาก CLDC และ KVM เพื่อลดขนาดและปรับปรุงประสิทธิภาพในการทำงาน โดยจะต้องศึกษาข้อจำกัดของ J2ME ก่อนที่จะเริ่มออกแบบและพัฒนาแอปพลิเคชัน

3.2.3 ประเภทของข้อมูลพื้นฐานของ CLDC สนับสนุนเฉพาะชนิดของชนิดข้อมูลที่นำมาจาก J2SE ซึ่งได้แก่ byte short int long char และ boolean แต่ไม่สนับสนุนข้อมูลประเภท float และ double เนื่องจากสาเหตุสองประการ คือ อุปกรณ์เป้าหมายที่นำ CLDC ไปใช้ ส่วนใหญ่ไม่มีฮาร์ดแวร์รองรับตัวเลขทศนิยม และหากจะใช้ซอฟต์แวร์เข้าช่วยก็จะต้องเสียค่าใช้จ่ายสูงมาก นอกจากนี้ CLDC ยังกำหนดคลาส type wrapper สำหรับข้อมูลทุกประการ ได้แก่

java.lang.Boolean

java.lang.Byte

java.lang.Character

java.lang.Integer

java.lang.Long

java.lang.Short

คลาส type wrapper กำหนดไว้ในจาวา เนื่องจากจาวามีระบบย่อยหลายระบบซึ่งทำงานได้เฉพาะกับอีโอบเจกต์เท่านั้นในกรณีนี้ คุณสามารถสร้างออบเจกต์โดยใช้คลาส wrapper ซึ่งเก็บชนิดของข้อมูลที่นำมาจาก J2SE เอาไว้

3.3 ไลบรารีของ MIDP

ไลบรารีของ CLDC ช่วยสร้างฟังก์ชันที่ไม่จำกัดเฉพาะอุปกรณ์บางชนิด ไลบรารีของ MIDP กลับตรงกันข้ามได้แก่ การจัดการแอปพลิเคชันบนอุปกรณ์ ส่วนการติดต่อกราฟิกกับผู้ใช้ทั้งแบบพื้นฐานและซับซ้อน พื้นที่เก็บข้อมูลแบบคงตัว และความสามารถเพิ่มเติมด้านเครือข่าย

3.3.1 คลาสโปรแกรมจัดการแอปพลิเคชัน เป็นคลาสที่ติดต่อกับ โปรแกรมจัดการแอปพลิเคชันบนอุปกรณ์ถูกกำหนดไว้ในแพ็คเกจ java.microedition.midlet แอปพลิเคชันทั้งหลายที่เขียนใน MIDP จะต้องขยายคลาส MIDlet ที่อยู่ในแพ็คเกจออกไป และจะต้องนำเมธอดทั้ง 3 ซึ่งได้แก่ startApp() pauseApp() และ destroyApp() เข้ามาใช้

3.3.2 คลาสของส่วนติดต่อกับกราฟิกกับผู้ใช้ (GUI Class) เนื่องจากชุดเครื่องมือ Abstract Windowing Toolkit ใน J2SE ออกแบบมาเพื่อใช้งานกับแอปพลิเคชันบนเครื่องคอมพิวเตอร์แบบตั้งโต๊ะและไม่สามารถนำมาใช้กับอุปกรณ์ไร้สายได้ เนื่องจากบริบททรัพยากรประเภทหน่วยความจำค่อนข้างมาก MIDP มีวิธีที่ต่างกันในการกำหนดไลบรารีสำหรับแพ็คเกจส่วนติดต่อกับกราฟิกกับผู้ใช้ ด้วยการให้ API ชั้นสูงซึ่งเน้นค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถในการเคลื่อนย้ายข้ามอุปกรณ์ และ API ขั้นพื้นฐานซึ่งเน้นองค์ประกอบกราฟิกเฉพาะอุปกรณ์ และ input event ทั่วไป คลาสที่เชื่อมต่อกับส่วนติดต่อกราฟิกกับผู้ใช้ และ event-handling กำหนดไว้ใน แพ็คเกจ java.microedition.lcdui Screen ถือเป็นซูเปอร์คลาสของคอมโปเนนต์ส่วนติดต่อกับผู้ใช้โดยใช้ API ขั้นสูง ประกอบไปด้วย Alert , Form , List , Textbox ฯลฯ Canvas และ Graphic เป็นคลาสหลักของ API ระดับต่ำ (Low Level) แอปพลิเคชันเกมส่วนใหญ่เป็นแอปพลิเคชันที่ใช้ API ระดับต่ำในการสร้างส่วนติดต่อผู้ใช้

3.3.3 คลาสของพื้นที่เก็บข้อมูลแบบคงตัว (Persistent Storage Class) ในบางครั้งแอปพลิเคชันที่เขียนบน MIDP จำเป็นต้องเก็บข้อมูลไว้บนอุปกรณ์อย่างถาวร คลาสที่กำหนดไว้ในแพ็คเกจ java.microedition.rms ให้กลไกการเก็บข้อมูลถาวรที่เรียกว่า RecordStore โดยยอมให้แอปพลิเคชันเขียนลบ และปรับปรุงบันทึกข้อมูลในหน่วยเก็บข้อมูลถาวรในอุปกรณ์

3.3.4 คลาสของเครือข่าย แม้ว่ากรอบการติดต่อสื่อสารทั่วไปที่กำหนดไว้ใน CLDC จะประกอบไปด้วยชุดของอินเทอร์เน็ตเฟสการเชื่อมต่อเครือข่าย แต่ก็ไม่มีโปรโตคอลอยู่เบื้องหลังอินเทอร์เน็ตเฟสการเชื่อมต่อจริงๆ หากแต่ปล่อยให้ให้เป็นหน้าที่ของ MIDP ในบรรดาอินเทอร์เน็ตเฟสการเชื่อมต่อเครือข่ายเหล่านี้ HttpURLConnection ถือเป็นอินเทอร์เน็ตเฟสหลักที่ต้องมีใน MIDP เสมอ คลาสของอินเทอร์เน็ตเฟสเหล่านี้กำหนดในแพ็คเกจ java.microedition.io

3.4 MIDlet

MIDlet หมายถึงแอปพลิเคชันบน MIDP MIDlet มีส่วนคล้ายกับจาวาแอปเพล็ต แม้จะไม่มีเมธอด main() แต่ MIDlet ก็นำคลาส java.microedition.midlet.MIDlet ตลอดจนเมธอด startApp() pauseApp() และ destroyApp() เข้ามาใช้ นอกจากนี้ MIDlet ยังกำหนด constructor แบบ public ที่ไม่มีอาร์กิวเมนต์ใดๆอีกด้วย เราสามารถให้คำนิยามของคลาส javax.microedition.midlet.MIDlet ได้ดังนี้

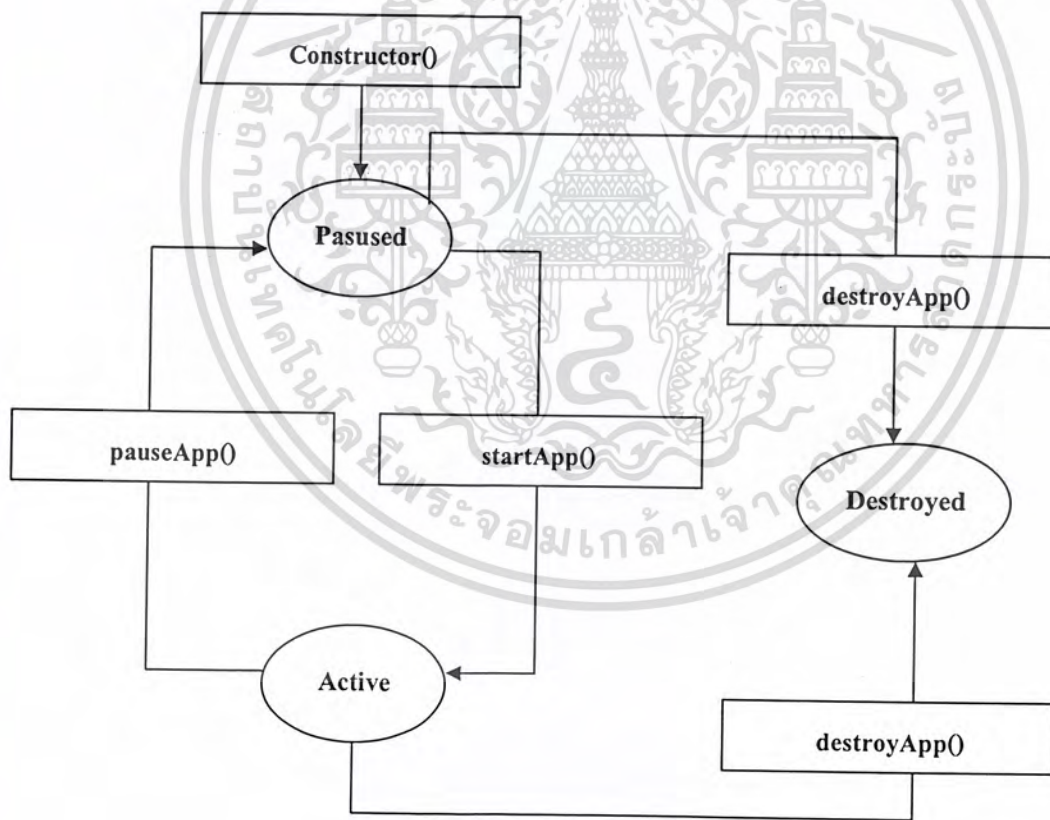
Public abstract คลาส MIDlet extends Object

```
{
    protected MIDlet()
    protected abstract void startApp() throws MIDletStateException
    protected abstract void pauseApp()
    protected abstract void destroyApp(boolean unconditional)
    throws MIDletStateException
    public final String getAppProperty(String key)
    public final void notifyDestroyed()
    public final void notifyPaused()
    public final String getAppProperty(String key)
    public final void resumeRequest()
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 วงจรการทำงานของ MIDlet

การกระทำของ MIDlet ประกอบไปด้วย 3 สถานะ คือกำลังทำงาน หยุดชั่วคราวและถูกทำลาย อาศัยซอฟต์แวร์จัดการแอปพลิเคชัน เป็นตัวควบคุมการเปลี่ยนสถานะหนึ่งไปยังอีกสถานะหนึ่งควบคุมด้วยเมธอด startApp() pauseApp() และ destroyApp() ที่มาพร้อมกับMIDlet จากรูปที่ 3.1 แสดงให้เห็นจุดเปลี่ยนระหว่างสถานะทั้ง 3 โดยการเรียกใช้เมธอดข้างต้น เมื่อ MIDlet พร้อมตั้งกระทำการ ซอฟต์แวร์จัดการแอปพลิเคชันจะสร้างตัวอย่าง MIDlet ขึ้นมาก่อน โดยใช้constructor แบบ public ที่ไม่มีอาร์กิวเมนต์ใดๆ โดย MIDlet จะอยู่ในสถานะหยุดชั่วคราว จากนั้นซอฟต์แวร์จัดการแอปพลิเคชันจะเรียกเมธอด startApp() ขึ้นมา และ MIDlet จะเข้าสู่สถานะกำลังทำงานเปิดรับทรัพยากรที่ต้องการและเริ่มต้นการทำงาน ในสถานะนี้ MIDlet จะทำงานและดึงทรัพยากรที่ต้องการไว้ใช้งาน เมื่อซอฟต์แวร์จัดการแอปพลิเคชันไม่ต้องการให้ MIDlet ทำงานต่อไป ก็จะเรียกเมธอด pauseApp() จากนั้น MIDlet จะหยุดทำงานและเข้าสู่สถานะหยุดชั่วคราว คินทรัพยากรที่ดึงมาใช้งาน และเข้าสู่ภาวะไม่ทำงาน MIDlet สามารถกลับไปอยู่ที่สถานะกำลังทำงาน ได้เมื่อซอฟต์แวร์จัดการแอปพลิเคชันเรียกเมธอด startApp() ขึ้นมา



รูปที่ 3.1 วงจรการทำงานของ MIDlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ท้ายที่สุดเมื่อซอฟต์แวร์จัดการแอปพลิเคชันไม่ต้องการเรียกใช้งาน MIDlet หรือต้องการเคลียร์หน่วยความจำเพื่อให้โปรแกรมอื่นได้ใช้งาน ก็จะส่งสัญญาณเตรียมทำลาย MIDlet ที่ด้วยการเรียกเมธอด `destroyApp()` และเข้าสู่สถานะถูกทำลาย ซึ่ง MIDlet จะปล่อยทรัพยากรทั้งหมด ทำการจัดเก็บข้อมูลต่างๆ และหยุดการทำงานทั้งหมด หาก MIDlet อยู่ระหว่างสังเคราะห์ขั้นตอนสำคัญอยู่ ก็อาจร้องขอไม่เข้าสู่สถานะถูกทำลายได้โดยเรียกใช้ `MIDletStateChangeException` อย่างไรก็ตามซอฟต์แวร์จัดการแอปพลิเคชันอาจปฏิเสธหรือยินยอมตามคำร้องขอนี้ก็ได้ ตัวแปรบูลีน `unconditional` ในลายเซ็นของเมธอด `destroyApp()` จะเป็นตัวกำกับว่าการร้องขอนี้สมควรหรือไม่ ถ้าตัวบ่งชี้มีค่าเท่ากับ `true` การร้องขอเป็นอันตกไป ตรงกันข้ามหากมีค่าเท่ากับ `false` ก็จะได้รับคำตอบรับและจะเรียกเมธอด `destroyApp()` ขึ้นมาใหม่ในคราวต่อไป หาก MIDlet ต้องการเข้าสู่สถานะหยุดชั่วคราวหรือถูกทำลายด้วยตัวเอง ก็สามารถทำได้โดยเรียกเมธอด `notifyPause()` หรือ `notifyDestroy()` ตามลำดับ โดยเมธอดทั้ง 2 จะแจ้งให้ซอฟต์แวร์จัดการแอปพลิเคชันทราบว่า MIDlet ได้เข้าสู่สถานะหยุดชั่วคราว/ถูกทำลายแล้ว ในกรณีนี้ซอฟต์แวร์จัดการแอปพลิเคชันจะไม่เรียกเมธอด `pauseApp()` หรือ `destroyApp()` อีก MIDlet จะเข้าสู่สถานะหยุดทำงานได้ก็ต่อเมื่อผ่านสถานะการดำเนินงานมาก่อนแล้ว ในทางตรงกันข้าม หากต้องการเข้าสู่สถานะถูกทำลาย สามารถเข้าได้โดยตรงจากสถานะหยุดทำงานชั่วคราว และสถานะกำลังทำงานนอกจากนี้ยังสามารถเข้าสู่สถานะหยุดการทำงานได้ในขณะที่กำลังทำงาน หรือเมื่อได้รับคำสั่งจากซอฟต์แวร์จัดการแอปพลิเคชัน

3.5 ซอฟต์แวร์จัดการแอปพลิเคชัน (Application Management Software)

ซอฟต์แวร์แอปพลิเคชันที่มาพร้อมกับ MIDP ทำหน้าที่ควบคุมการติดตั้ง สังเคราะห์การ และลบ MIDlet บางครั้งเราเรียกซอฟต์แวร์จัดการแอปพลิเคชันว่าเป็น ซอฟต์แวร์จัดการ MIDlet (MIDlet Management Software) หรือ โปรแกรมบริหารจาวาแอปพลิเคชัน (Java Application Manager)

การติดตั้งซอฟต์แวร์จัดการแอปพลิเคชันแตกต่างกันออกไปตามแต่ประเภทของอุปกรณ์ แต่ว่าให้บริหารจัดการขั้นพื้นฐานเหมือนกัน ได้แก่

- ยอมให้ผู้ใช้งานติดตั้งและถอดถอน MIDlet จากอุปกรณ์ไร้สาย ทั้งผ่านสายเคเบิลที่เชื่อมต่อเข้ากับเครื่องคอมพิวเตอร์ส่วนบุคคล หรือ จากระยะไกลโดยผ่านอินเทอร์เน็ต
- เตรียมสภาพแวดล้อม สังเคราะห์การให้แก่ MIDlet หลังจาก MIDlet เริ่มทำงานซอฟต์แวร์จัดการแอปพลิเคชันจะจัดเตรียมทรัพยากรของระบบ เช่น คลาสของ CLDC และ MIDlet และ KVM ไว้ให้ MIDlet ใช้งาน นอกจากนี้ยังเตรียมคลาสไฟล์ทรัพยากรและไฟล์ descriptor ของ แออปพลิเคชันไว้ให้ MIDlet ได้ใช้เป็นรันไทม์ในการทำงานอีกด้วย
- ซอฟต์แวร์จัดการแอปพลิเคชันยังจัดการแก้ไขข้อผิดพลาดทั้งหมดที่เกิดขึ้นระหว่างการจัดตั้ง และสังเคราะห์การแอปพลิเคชัน โดยไม่ทำให้ระบบล่มอีกด้วย

3.6 ข้อจำกัดต่างๆ

3.6.1 ข้อจำกัดในการจัดการข้อผิดพลาด CLDC นั้นรองรับความผิดพลาดในการทำงานในวงจำกัด โดยกำหนดคลาสแสดงข้อผิดพลาดไว้เพียง 2 คลาส คือ `java.lang.VirtualMachineError` และ `java.lang.OutOfMemoryError` ขณะที่ดึงคลาสส่วนใหญ่ออกไป เนื่องจาก

3.6.1.1 ในระบบอุปกรณ์ฝังตัว การคืนสู่ภาวะทำงานผิดพลาด เป็นความสามารถเฉพาะบางอุปกรณ์ เท่านั้น ไม่ควรคาดหวังให้ผู้เขียนแอปพลิเคชันจัดการกับข้อบกพร่องเฉพาะอุปกรณ์เหล่านี้

3.6.1.2 ความผิดปกติประเภท Error มักไม่สามารถแก้ไขได้ การบรรจุความสามารถในการจัดการข้อผิดพลาดอย่างเต็มรูปแบบทำให้ต้องเสียค่าใช้จ่ายมาก และยังคงกำหนดลำดับความสำคัญไว้สูงทั้งที่อุปกรณ์ CLDC มีทรัพยากรค่อนข้างจำกัด

3.6.2 ข้อจำกัดของ KVM เนื่องจาก KVM เป็นเลขอร์ Java Virtual Machine ที่รองรับไลบรารีของ CLDC คุณสมบัติใดที่ CLDC ไม่สนับสนุน ก็จะถูกดึงออกจาก KVM ด้วยเช่นกัน อาทิ ตัวเลขทศนิยม finalization และการจัดการข้อผิดพลาดเฉพาะอุปกรณ์

3.6.2.1 ไม่สนับสนุน Java Native Interface (JNI) ด้วยสาเหตุ 2 ประการคือในแบบจำลองความปลอดภัยของ CLDC ผู้เขียนแอปพลิเคชันไม่สามารถดาวน์โหลดไลบรารีใหม่ๆที่ยังมีฟังก์ชันเดิมของอุปกรณ์อยู่ และไม่สามารถเข้าถึงฟังก์ชันเดิมของอุปกรณ์ที่ไม่มีในไลบรารีของจาวาได้ นอกจากนี้การนำ JNI มาใช้ยังเพิ่มภาระให้กับหน่วยความจำบนอุปกรณ์ CLDC

3.6.2.2 ไม่สนับสนุนโปรแกรมบรรจุคลาสที่ผู้ใช้งานกำหนดขึ้นเอง และโปรแกรมบรรจุคลาสใน KVM ก็ไม่อนุญาตให้ผู้ใช้งานลบล้าง เขียนทับ หรือกำหนดค่าใหม่ด้วย

3.6.2.3 ไม่สนับสนุน Reflection , RMI และ การเรียงลำดับอ็อบเจกต์ ทำให้โปรแกรม CLDC ไม่สามารถตรวจสอบเนื้อหาในคลาสอ็อบเจกต์ เมธอด ฯลฯ

3.6.2.4 ไม่สนับสนุน Thread Group หรือ Daemon คือ KVM สนับสนุนแอปพลิเคชันแบบทำงานคู่ขนาน (Multi-Thread) แต่ไม่รองรับกลุ่มงานย่อย (Thread Object) หรืองานเบื้องหลัง (Daemon Thread)

3.7 ระบบความปลอดภัยใน J2ME

วัตถุประสงค์สำคัญของการรักษาความปลอดภัยบนระบบคอมพิวเตอร์คือ การป้องกันทรัพยากรของระบบจากการเจาะเข้ามาทั้งที่ไม่ประสงค์ดีและโดยบังเอิญ การดาวน์โหลดเนื้อหาและแอปพลิเคชันตามข้อกำหนดของ CLDC ระบุไว้ ทำให้การรักษาความปลอดภัยบนระบบเครือข่ายทวีความสำคัญมากขึ้นจาวานั้น ออกแบบมาโดยคำนึงถึงความปลอดภัยมาตั้งแต่ต้น โดย J2SE เองมีคุณสมบัติด้านความปลอดภัยที่หลากหลาย เช่น การตรวจไบต์โค้ด และโปรแกรมจัดการความปลอดภัย อย่างไรก็ตาม เนื่องจากรหัสที่ J2SE ใช้ในการรักษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความปลอดภัยมีขนาดใหญ่เกินกว่าหน่วยความจำของอุปกรณ์ไร้สายที่มีทรัพยากรจำกัด จึงต้องลดทอนคุณสมบัติดังกล่าวลงให้เหมาะสมกับความต้องการระบบของอุปกรณ์ไร้สาย

3.8 การเวอริฟายและฟรีเวอริฟายไฟล์คลาส

เนื่องจากทรัพยากรที่มีจำกัดบนอุปกรณ์ไร้สายการเวอริฟายคลาสของ KVM จึงต้องแยกกันกระทำ โดยบางส่วนกระทำนอกตัวอุปกรณ์นั้น เรียกว่า ฟรีเวอริฟิเคชัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

คอมโพเนนต์หลักของส่วนติดต่อกับผู้ใช้ในภาษาจาวา

4.1 บทนำ

J2ME สามารถทำงานบนอุปกรณ์ที่หลากหลาย ตั้งแต่สมาร์ตการ์ดและโทรศัพท์เคลื่อนที่ ไปจนถึง อุปกรณ์ set-top-box สำหรับโทรทัศน์และเครื่องใช้ไฟฟ้าในบ้าน อุปกรณ์เหล่านี้มีความสามารถในการแสดงผลและทรัพยากรที่แตกต่างกันค่อนข้างมาก บริษัทซัมซุง ไมโครซิสเต็มส์ตระหนักถึงปัญหานี้จึงไม่ได้กำหนดคลาสส่วนติดต่อกับผู้ใช้ซึ่งสมบูรณ์แบบใน J2ME หากแต่กำหนด API ของส่วนติดต่อกับผู้ใช้ในลักษณะโพรไฟล์ของอุปกรณ์เคลื่อนที่ (Mobile Information Device Profile : MIDP) หรือถ้าเป็น API ส่วนติดต่อกับผู้ใช้สำหรับพีดีเอ ก็จะระบุไว้ในโพรไฟล์ของพีดีเอ (Personal Digital Assistant Profile:PDAP) เป็นต้น

API ของส่วนติดต่อกับผู้ใช้ใน J2SE กำหนดไว้ในชุดเครื่องมือ Abstract Window Toolkit(AWT) ซึ่งออกแบบสำหรับเครื่องคอมพิวเตอร์ตั้งโต๊ะที่มีอุปกรณ์การชี้ (Pointer Device) จอภาพขนาดใหญ่พร้อมหน่วยความจำที่พอเพียง ขณะที่หน้าจอแสดงผลไร้สายมีขนาดเล็กกว่า และใช้เป็นกดเพื่อป้อนข้อมูลเข้าแทนที่จะเป็นนิ้วชี้ ดังนั้นกลุ่มผู้เชี่ยวชาญการออกแบบ MIDP จึงได้กำหนด API ของส่วนติดต่อกับผู้ใช้ขึ้นมาใหม่ทั้งหมด โดยไม่ได้อ้างอิงซบเซตของ AWT เลยซึ่ง API ของส่วนติดต่อกับผู้ใช้เหล่านี้กำหนดไว้ในแพ็คเกจ `javax.microedition.lcdui`

API ของส่วนติดต่อกับผู้ใช้ใน MIDP มี 2 กลุ่มหลักๆ ด้วยกันคือ API ระดับสูงและระดับล่าง API ระดับสูงออกแบบให้เป็น Abstract เน้นความยืดหยุ่นในการโยกย้ายไปใช้งานในระบบอื่น แอปพลิเคชันที่ใช้ API ระดับสูงจะควบคุมวิธีแสดงผลได้น้อยมากและเข้าถึงได้เฉพาะอีเวนต์ระดับสูงเท่านั้น ในทางตรงกันข้าม การใช้ API ระดับล่างจะสามารถควบคุมการแสดงผลได้เต็มที่ สามารถเข้าถึงอุปกรณ์ป้อนข้อมูล และจัดการอีเวนต์พื้นฐานที่เกิดจากการตอบโต้ของผู้ใช้ได้ อย่างไรก็ดีโปรแกรมที่ใช้ API ระดับล่างนี้จะขาดความยืดหยุ่น ยึดติดกับอุปกรณ์มากจนเกินไป โดยจะต้องพิจารณาถึงข้อดีข้อเสียในการออกแบบโปรแกรมด้วย

4.2 Displayable และ Display

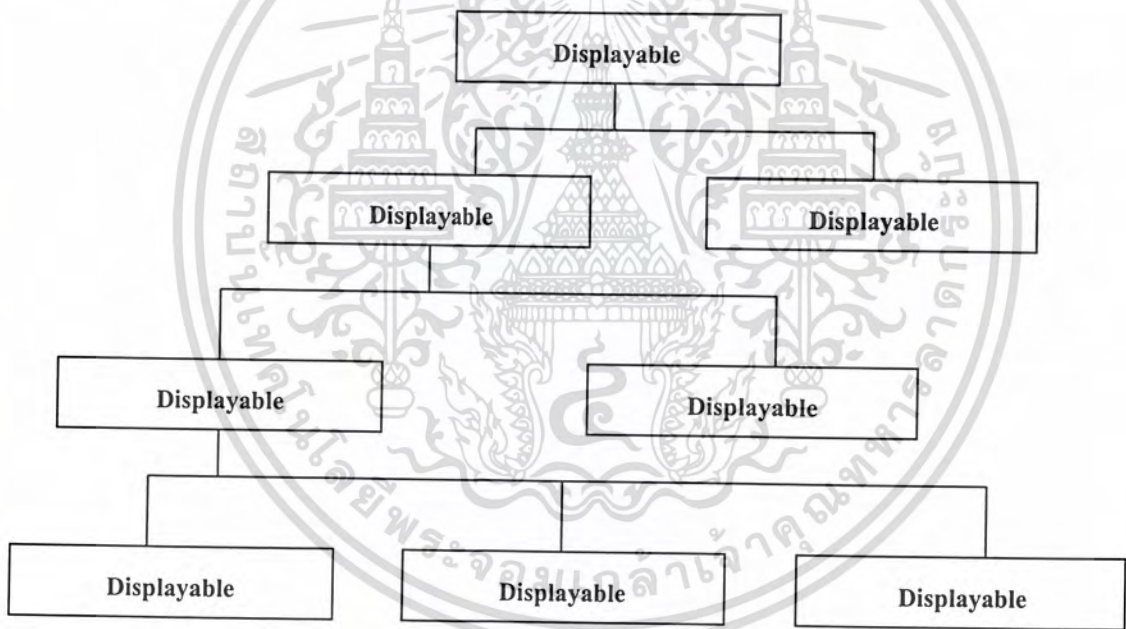
ฟังก์ชันหลักอย่างหนึ่งของส่วนติดต่อกับผู้ใช้ ก็คือ การแสดงข้อมูลบนหน้าจอใน J2ME MIDP ออบเจกต์ Displayable จะเก็บข้อมูลที่ต้องการนำเสนอ และออบเจกต์ Display จะนำออบเจกต์ Displayable นั้นไปแสดงบนจอให้ผู้ใช้เห็น

4.2.1 Displayable

เป็นคลาสหลักของส่วนติดต่อกับผู้ใช้ของ MIDP สามารถเก็บภาพกราฟิกที่สร้างเฉพาะอุปกรณ์และนำไปแสดงบนจอ ตัว MIDP กำหนด API ของส่วนติดต่อกับผู้ใช้ออกเป็น 2 กลุ่มคือ API ระดับสูงและระดับล่าง คลาส Screen เป็นคลาสย่อยของ Displayable ที่ใช้ API ระดับสูง ส่วนคลาส Canvas เป็นคลาสย่อยของ Displayable ที่ใช้ API ระดับล่าง

ขนาดและรูปร่างของหน้าจออุปกรณ์จะแตกต่างกันออกไป เนื่องจาก API ระดับสูงจะจัดการกับความแตกต่างของขนาดหน้าจอ ดังนั้น ในการเขียนแอปพลิเคชันจึงไม่ต้องพะวงเรื่องความแตกต่างของขนาดหน้าจอมากนัก แต่หากเป็นแอปพลิเคชันที่ใช้ API ระดับล่างจะต้องระวังเรื่องความแตกต่างของหน้าจอแสดงผลเหล่านี้

Displayable ทุกตัวต้องวิธีสแตนด์ Command และ CommandListener เพื่อที่จะสามารถรับอีเวนต์ของคอมมานด์ได้



รูปที่ 4.1 แสดงโครงสร้างลำดับชั้นของคลาส Displayable

4.2.1.1 Screen

เป็นซับคลาสของ Displayable ซึ่งนำ API ระดับสูงมาใช้ประกอบด้วยชูปเปอร์คลาส Alert, Form, List และ TextBox อาจมีชื่อหรือแถบตัวอักษรวิ่ง (ticker tape)

กำกับหรือจะไม่มีก็ได้

Screen เป็นคลาสแบบ Abstract เมธอดซึ่งกำหนดวิธีการแสดงผลและการโต้ตอบกับเนื้อหาในออบเจกต์ก็เป็นเมธอดแบบ Abstract ด้วย นำมาใช้โดยซับคลาสของ Screen มีเพียงเมธอดสำหรับปรับปรุงหน้าจอ

เวลาเมื่อต้องการแสดงผลเท่านั้นที่อยู่ในคลาส Screen อย่างไรก็ดี ไม่สามารถเพิ่มเติมอะไรในคลาส Screen ได้เอง เนื่องจากใช้เมธอดแบบ Abstract ซึ่งไม่สามารถเข้าถึงได้จากภายนอกแพ็คเกจ javax.microedition.lcdui

4.2.1.2 Canvas

เป็นซับคลาสของ Displayable ซึ่งใช้ API ระดับล่างแอปพลิเคชันสามารถควบคุมได้ว่าแสดงผลอะไรและอย่างไรบ้าง และยังสามารเข้าถึงอีเวนต์ระดับล่างเช่น การกดปุ่ม ได้ด้วย Canvas เป็นคลาส Abstract แอปพลิเคชันต้องสร้างซับคลาสย่อยเพื่อใช้งาน

4.2.2 Display

เป็นตัวจัดการการแสดงผล และใช้เมธอดเหล่านี้สำหรับอ่านค่าพารามิเตอร์ของอุปกรณ์และนำออบเจกต์ไปแสดงผลบนหน้าจอ

ในโปรแกรม MIDlet หนึ่งๆ จะมีออบเจกต์ของ Display ได้เพียงตัวเดียวเท่านั้นแอปพลิเคชันสามารถเข้าถึงออบเจกต์ Display ได้ด้วยเมธอด getDisplay() และสามารถเรียกเมธอดนี้ได้ทุกเมื่อตั้งแต่เริ่มเรียกใช้ MIDlet.startApp() ไปจนถึง MIDlet.destroyApp() ออบเจกต์ Display ทั้งหมดที่ได้จาก getDisplay() จะยังคงอยู่ตลอด จนกว่าจะเสร็จสิ้น โปรแกรม MIDlet

ในแอปพลิเคชันหนึ่งๆจะมีออบเจกต์ Displayable ที่จะแสดงผลบนอุปกรณ์หรือมีการทำงานโต้ตอบกับผู้ใช้ได้เพียงออบเจกต์เดียวเท่านั้น โดยเรียก Displayable นี้ว่า Displayable ปัจจุบัน ผู้ใช้จะป้อนข้อมูลหรือโต้ตอบได้เฉพาะกับ Displayable ปัจจุบันเท่านั้น คลาส Displayable มีเมธอด getCurrent() สำหรับดึง Displayable ปัจจุบัน และมีเมธอดไว้สำหรับแสดงผลคือ setCurrent(Displayable nextDisplayable)

แอปพลิเคชัน สามารถควบคุม Displayable ปัจจุบัน และ เรียก เมธอด Display.setCurrent() ได้ตลอดเวลา เรดของ MIDlet สามารถกำหนดหรือดึงค่า Displayable ปัจจุบัน ด้วย เมธอด Display.setCurrent() และ Display.getCurrent()

แอปพลิเคชันหน้าฉากและหลังฉาก แม้สามารถสั่งให้แอปพลิเคชันหลายตัวทำงานพร้อมกันบนอุปกรณ์หนึ่งๆ อุปกรณ์ก็จะแสดงผลได้เฉพาะ Displayable ปัจจุบันเท่านั้น ดังนั้นแอปพลิเคชันใดที่มี Displayable ปัจจุบันแสดงผลอยู่บนจอจะเรียกว่าเป็นแอปพลิเคชันหน้าฉาก (foreground) ซึ่งสามารถรับ อีเวนต์การป้อนข้อมูลเข้าจากผู้ใช้ได้ แอปพลิเคชันที่เหลือซึ่งไม่ได้แสดงผลจะเรียกว่าแอปพลิเคชันหลังฉาก (background)

MIDlet สามารถวางตัวเองให้อยู่หลังฉากได้โดยใช้ Display.setCurrent(null) แต่การใช้เมธอดนี้ไม่ได้กำหนด Displayable ปัจจุบันให้เป็น null จริงๆ สมมติว่า Displayable ปัจจุบันคือ cD หลังจากแอปพลิเคชันเรียก Display.setCurrent(null) แล้ว การเรียกเมธอด getCurrent() ก็ ยังได้ค่าเป็น cD แอปพลิเคชัน จะกลับคืนสู่หน้าฉากโดยใช้ Display.setCurrent(Display.getCurrent())

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แม้ว่า Displayable ปัจจุบันของแอปพลิเคชันจะไม่ได้วาดอะไรบนจอก็ตาม แอปพลิเคชันก็ยังรับรู้ว่ามี Displayable ใดเป็น Displayable ปัจจุบัน ซึ่งสำคัญมากแม้แต่กับแอปพลิเคชันหลังจาก เนื่องจากเมื่อแอปพลิเคชันหลังจากกลายเป็นแอปพลิเคชันหน้าจอก ก็จะสามารถแสดงผล Displayable ที่ ถูกต้องได้

โปรแกรม MIDlet แต่ละตัวจะมี Displayable ปัจจุบันของตัวเอง เมธอด `getCurrent()` จะให้ Displayable ปัจจุบัน โดยใช้เมธอด `setCurrent()` ก็ไม่มีผลอะไรกับ Displayable ปัจจุบันของ MIDlet อื่นๆ

System Screen โดยปรกติแล้ว จะสามารถมองเห็นการแสดงผลบนหน้าจอปัจจุบันของ MIDlet ที่ทำงานหน้าจอกได้ อย่างไรก็ตามในบางสถานการณ์ระบบอาจจะสร้างหน้าจอชั่วคราวขึ้นมาบังหน้าจอปัจจุบันของแอปพลิเคชันเอาไว้ คือ หน้าจอของระบบ(system screen) โดยจะแสดงขึ้นมาเมื่อระบบต้องการแสดงผลเพิ่มเติม นอกเหนือไปจากการแก้ไขฟิลด์ข้อความต่างๆใน Form

แม้ว่าหน้าจอของระบบ จะโผล่ขึ้นมาบังหน้าจอของแอปพลิเคชัน Displayable ปัจจุบันก็ยังเป็นอันเดิมไม่เปลี่ยนแปลง การเรียกใช้เมธอด `getCurrent()` ในขณะที่มองเห็นหน้าจอของระบบ ก็ยังได้ค่า Displayable ปัจจุบันของแอปพลิเคชัน และไม่ใช่หน้าจอของระบบถ้าจะดูว่า Displayable ปัจจุบัน ถูกบังด้วยหน้าจอของระบบหรือไม่ สามารถตรวจสอบได้จาก `currentDisplayable.isShown()`

4.3 รูปภาพ (Image)

คลาส Image ใช้เก็บข้อมูลรูปภาพ ออบเจกต์ Image จะอยู่ในหน่วยความจำที่แยกจากส่วนแสดงผล ออบเจกต์ Image สามารถวาดลงบน Canvas หรือวางใน Form , Alert , List หรือ ChoiceGroup ก็ได้ Image อาจเป็นรูปภาพที่แก้ไขได้ หรือแก้ไขไม่ได้ ขึ้นอยู่กับวิธีการสร้าง

4.3.1 รูปที่แก้ไขไม่ได้ มักมีที่มาจากภาพโหลดภาพจากไฟล์ จากแหล่งอื่นๆที่ใส่มาในแพ็คเกจ หรือจากเครือข่าย เมื่อสร้างรูปเสร็จแล้ว จะไม่สามารถแก้ไขได้อีก รูปที่ใส่ใน Alert ,Choice หรือ Item จะต้องเป็นรูปที่แก้ไขไม่ได้เท่านั้น

4.3.2 รูปที่แก้ไขได้ จะสร้างในหน่วยความจำที่อยู่นอกจอ เสมือนกับว่าเป็นหน้าจอหรือ Canvas อีกอันหนึ่ง แอปพลิเคชันสามารถวาดรูปที่แก้ไขนี้ได้จากที่สร้างขึ้นมาแล้ว ด้วยเมธอด `Image.createImage(int width,int height)` โดยรูปที่แก้ไขได้จะใช้ในเทคนิคแบบบัพเฟอร์ 2 ชั้น

พารามิเตอร์ของรูป ออบเจกต์ของ image จะมีพารามิเตอร์ 3 แบบด้วยกันคือ height width และ isMutable แอปพลิเคชันสามารถอ่านค่าพารามิเตอร์ได้โดยใช้เมธอด `getHeight()` `getWidth()` และ `isMutable()` ตามลำดับ ถ้ารูปนั้นแก้ไขได้ แอปพลิเคชันจะสามารถสร้างออบเจกต์ Graphics เพื่อสร้างรูปโดยใช้ `getGraphics()`

4.4 อีเวนต์และการจัดการอีเวนต์

ส่วนติดต่อกับผู้ใช้ประกอบด้วย 2 ส่วนหลักคือ หน้าจอที่ทำหน้าที่แสดงผล และตอบสนองการโต้ตอบของผู้ใช้ ส่วนติดต่อกับผู้ใช้ในอีเวนต์จะทำหน้าที่ประมวลผลอีเวนต์ที่เกิดขึ้น อีเวนต์และตัวจัดการอีเวนต์มีทั้งระดับสูงและระดับต่ำเช่นเดียวกันกับ API ของDisplayable



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การใช้ API ระดับสูงของภาษาจาวา

5.1 บทนำ

API ระดับสูงเน้นความสามารถในการโยกย้ายไปใช้กับอุปกรณ์อื่นๆ ชับคลาสของ Screen มี 2 กลุ่ม กลุ่มแรก ได้แก่ List TextBox และ Alert ซึ่งมีโครงสร้างที่กำหนดไว้ล่วงหน้าแล้ว กับกลุ่มที่สองได้แก่คลาส Form ซึ่งกำหนดโครงสร้างโดยแอปพลิเคชัน

5.2 List และ Choice

เมื่อเริ่มการทำงาน แอปพลิเคชันมักจะแสดงเมนูซึ่งประกอบด้วยฟังก์ชันต่างๆ ให้คุณได้เลือกตามต้องการ List และ ChoiceGroup ต่างแสดงรายการทางเลือก โดยทั้งคู่ใช้อินเทอร์เฟซ Choice ที่กำหนดไว้ในแพ็คเกจ javax.microedition.lcdui

5.2.1 List

เป็น Screen ซึ่งมีโครงสร้างที่กำหนดไว้แล้วโดยใช้อินเทอร์เฟซ Choice API ของแพ็คเกจ javax.microedition.lcdui กำหนด constructor ของ list ไว้ 2 แบบ โดยตัวแรกจะสร้าง List เปล่าๆขึ้นมา ซึ่งสามารถเพิ่มรายการของ choice เข้าไปได้ทีหลัง constructor ตัวที่สองจะสร้าง List ที่มีค่าเริ่มต้นอยู่ข้างใน อาร์เรย์ stringElements ต้องไม่เป็น null และสมาชิกทุกตัวในอาร์เรย์ต้องไม่เป็น null ด้วยเช่นกัน ความยาวของอาร์เรย์ stringElements จะกำหนดจำนวนรายการใน List อาร์เรย์ imageElements อาจมีค่าเป็น null ซึ่งหมายความว่ารายการใน List นั้นไม่มีภาพอยู่ ถ้าอาร์เรย์ imageElements ไม่เป็น null ก็จะต้องมีความยาวเท่ากับอาร์เรย์ stringElements รายการแต่ละรายการในอาร์เรย์ imageElements อาจจะเป็น null ได้ซึ่งหมายความว่าไม่มีภาพในรายการของ List ที่ตรงกัน

5.2.2 Choice

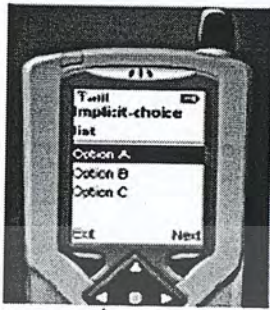
ออบเจกต์ Choice มี 3 ประเภทได้แก่ implicit-choice (ใช้ได้กับ List เท่านั้น) exclusive-choice และ multiple-choice

5.2.2.1 Exclusive-choice จะต้องเลือก รายการใดรายการหนึ่ง ในแต่ละครั้งเว้นเสียแต่ว่าไม่มีรายการใดๆให้เลือก

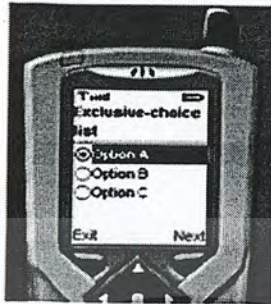
5.2.2.2 Implicit-choice เป็นกรณีพิเศษของ exclusive-choice เมื่อรายการที่อยู่ในโฟกัสถูกเลือกโดยอัตโนมัติเมื่อเริ่มต้น Command โดย choice ประเภทนี้ใช้ได้กับ List เท่านั้น

5.2.2.3 Multiple-choice สามารถเลือกหลายรายการพร้อมกันได้ในแต่ละครั้งหรือไม่เลือกเลยก็ได้

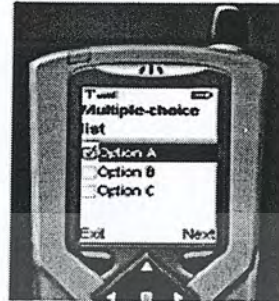
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 List แบบ
Implicit-choice



รูปที่ 5.2 List แบบ
exclusive-choice



รูปที่ 5.3 List แบบ
multiple-choice

ออบเจกต์ Choice จะแสดงรายการทางเลือกแบบต่างๆ ให้ผู้ใช้เลือก แต่ละรายการประกอบด้วยสตริงข้อความและรูปซึ่งอาจจะมีหรือไม่มีก็ได้

จำนวนรายการในออบเจกต์ Choice สามารถดูได้โดยใช้เมธอด size() แต่ละรายการสามารถอ้างอิงได้ตามดัชนีซึ่งเป็นตัวเลขจำนวนเต็มมีค่าตั้งแต่ 0 จนถึง size() - 1 โดยที่ 0 อ้างถึงรายการแรกและ size() - 1 อ้างถึงรายการสุดท้าย

ภาพในรายการ Choice อาจเว้นว่างได้ถ้าแอปพลิเคชันนั้นไม่ได้กำหนดไว้ หรือแม้จะกำหนดไว้ก็อาจเลือกไม่แสดงภาพเลย หากอุปกรณ์นั้นๆ ไม่สามารถแสดงผลได้ ถ้าเลือกที่จะแสดงภาพภาพจะติดอยู่กับสตริงข้อความและถือเสมือนว่าทั้งภาพและข้อความนั้นเป็นหน่วยเดียวกัน ภาพไม่สามารถเปลี่ยนได้ แอปพลิเคชันสามารถกำหนดภาพให้กับรายการ elementNum choice ได้โดย Void set(int elementNum, String stringPart, Image imagePart) และอ่านภาพจาก elementNum choice ได้โดย Image getImage(int elementNum)

5.3 Textbox

เป็น Screen ที่มีโครงสร้างกำหนดไว้แล้ว ผู้ใช้งานสามารถป้อนหรือแก้ไขข้อความใน TextBox ได้ การสร้างออบเจกต์ TextBox ทำได้โดยใช้ constructor ดังนี้

```
TextBox(String title, String text, int maxSize, int constraints)
```

ข้อความใน TextBox จะเก็บเป็นอาร์เรย์ของ char ดังนั้นจึงมีขนาด (ความจุ) ในการเก็บกลุ่มตัวอักษรไว้ในออบเจกต์อย่างจำกัด โดยสามารถกำหนดขนาดสูงสุดของ TextBox ในขณะที่สร้างออบเจกต์หรืออาจใช้เมธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อด `setMaxSize()` กำหนดหลังจากที่สร้างออบเจกต์ `TextBox` แล้ว การดูขนาดความจุทำได้โดยใช้เมธอด `getMaxSize()` เนื่องจากออบเจกต์ `TextBox` มีหน่วยความจำจำกัด ชิดจำกัดสูงสุดที่แท้จริงขึ้นอยู่กับค่าสูงสุดที่ `MIDP` กำหนดไว้ ดังนั้นขนาดสูงสุดจริงของ `TextBox` จะจำกัดอยู่ภายในกรอบที่กำหนดโดย `MIDP`

การตรวจสอบขนาดความจุสูงสุดเกิดขึ้นเมื่อสร้างออบเจกต์ `TextBox` เมื่อผู้ใช้งานแก้ไขข้อความภายใน `TextBox` และเมื่อโปรแกรมเรียกใช้เมธอดเพื่อแก้ไขเนื้อหาของ `TextBox` เมื่อใดก็ตามที่ข้อความใน `TextBox` มีขนาดเกินกว่าที่กำหนดไว้ จะเกิด `IllegalArgumentException` ขึ้น

ข้อความใน `TextBox` อาจมีความยาวเกินกว่าจะแสดงผลได้หมดในครั้งเดียว ในกรณีนี้ผู้ใช้สามารถเลื่อนหน้าจอขึ้น-ลง เพื่อดูหรือแก้ไขข้อความส่วนต่างๆ การเลื่อนหน้าจอนี้ไม่ทำให้เกิดอีเวนต์ `application-visible` แต่อย่างไร

5.4 Alert

เป็น `Screen` ที่มีโครงสร้างกำหนดไว้แล้ว ใช้สำหรับแสดงข้อมูล (ข้อความและภาพ) ให้ผู้ใช้และจะคอยเป็นเวลาคู่หนึ่ง (กำหนดโดย `timeout`) ก่อนที่จะแสดงหน้าจอถัดไป มี `Constructor 2` เมธอด ดังนี้

`Alert(String title)`

`Alert(String title, String alertText, Image alertImage, AlertType alertType)`

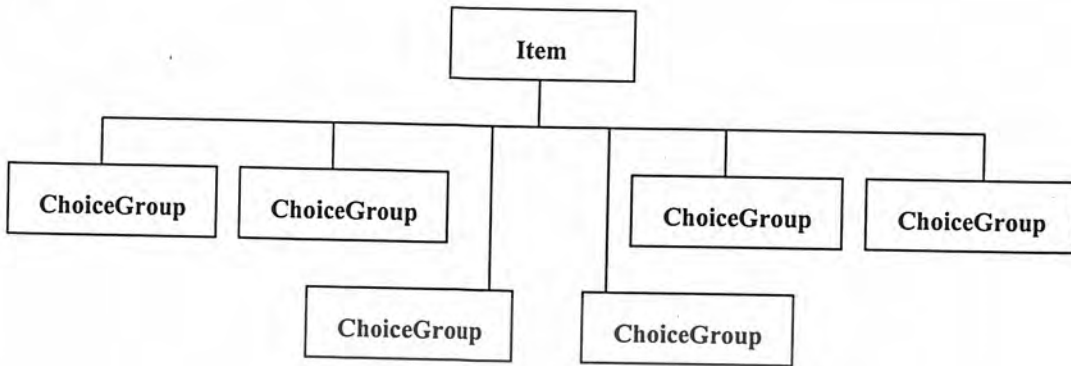
5.5 Form และ Item

`Form` เป็นออบเจกต์ `Screen` ซึ่งสามารถบรรจุ `Item` ต่างๆ เช่น ภาพ ข้อความ หรือ `choice` จำนวนเท่าใดก็ได้ การสร้างออบเจกต์ `Form` ทำได้ 2 แบบ คือ สร้าง `Form` เปล่าที่ไม่มี `Item` ใดๆ โดยใช้ `Form(String title)` หรือสร้าง `Form` ที่มี `Item` เริ่มต้นโดยใช้ `Form(String title, Item[] item)`

สามารถดูจำนวนของ `Item` ใน `Form` ได้โดยใช้เมธอด `size()` การอ้างถึง `Item` ทำได้โดยใช้ดัชนีตัวเลข ซึ่งมีค่าตั้งแต่ 0 ถึง `size() - 1` อ้างถึงตัวสุดท้าย สามารถดึงออบเจกต์ `Item` จาก `Form` ได้โดยใช้ `get(int itemNum)`

`Item` หนึ่งจะอยู่ได้ใน `Form` เดียวเท่านั้น หากแอปพลิเคชันโปรแกรมพยายามใส่ `Item` ที่มีแล้วใน `Form` นี้หรือ ในฟอร์มอื่นๆ ลงไปในอีก `Form` หนึ่งก็จะเกิด `IllegalStateException` ขึ้น ดังนั้นจะต้องย้าย `Item` ออกจาก `Form` ที่อยู่เดิมก่อนจะสามารถใส่เข้าไปใน `Form` ใหม่ได้

5.5.1 `Item` เป็นซูเปอร์คลาสสำหรับคอมโพเนนต์ต่างๆที่สามารถใส่ลงไปใน `Form` ได้ แต่ละ `Item` จะมีป้ายข้อความกำกับ ซึ่งเป็นฟิลด์สตริงข้อความที่ติดอยู่กับ `Item` โดยปกติจะแสดงไว้ใกล้ๆ กับคอมโพเนนต์ที่ปรากฏในหน้าจอ



รูปที่ 5.4 โครงสร้างลำดับชั้นของคลาส Item

5.5.1.1 **ChoiceGroup** คือ Item ที่แสดงผลแบบ Choice มีลักษณะคล้ายกับ List แต่ที่ไม่มีแบบ implicit-choice การแสดงผลแบบ exclusive หรือ multiple-choice ของ ChoiceGroup จะเหมือนกับ List การสร้าง ChoiceGroup ทำได้ดังนี้ `ChoiceGroup(String label, int choiceType)`

5.5.1.2 **ImageItem** อาจมีออบเจกต์ Image ซึ่งสามารถเปลี่ยนแปลงได้ การสร้างออบเจกต์ ImageItem ทำได้ดังนี้ `ImageItem(String label, Image img, int layout, String altText)`

5.5.1.3 **StringItem** ประกอบด้วยป้ายและสตริงข้อความที่แสดงผลอย่างเดียวยังผู้ใช้ไม่สามารถแก้ไขค่าใน StringItem ได้โดยตรง ต้องทำผ่านแอปพลิเคชันเท่านั้นโดยใช้เมธอด `setText()` การสร้างออบเจกต์ของ StringItem ทำได้ดังนี้ `StringItem(String label, String text)`

5.5.1.4 **Gauge** เป็น Item ที่แสดงผลแบบกราฟแท่งโดยมีค่าระหว่าง 0 ถึง `maxValue` การสร้าง ออบเจกต์ Gauge ทำได้ดังนี้ `Gauge(String label, Boolean interactive, int maxValue, int initialValue)`

5.5.1.5 **DateField** คือ Item ที่แก้ไขได้ ใช้สำหรับแสดงค่าวัน เดือน ปี และ เวลา สามารถสร้างออบเจกต์ DateField โดยใช้ `DateField(String label, int mode, TimeZone timeZone)`

5.5.1.6 **TextField** เป็น Item ที่มีสตริงข้อความซึ่งผู้ใช้สามารถแก้ไขได้ การสร้างออบเจกต์ TextField ทำได้ดังนี้ `TextField(String label, String text, int maxSize, int constraints)`

ข้อแตกต่างระหว่าง TextField และ TextBox คือ TextBox เป็น subclass ของ Screen ดังนั้นจึงสามารถแสดงผลบนหน้าจอได้ทันที ทว่า TextField เป็น subclass ของ Item จะสามารถแสดงผลได้ก็ต่อเมื่อนำไปไว้ใน Form

5.5.2 **Form** เป็นคอมโพเนนต์ที่สำคัญที่สุดในการพัฒนาส่วนติดต่อกับผู้ใช้สำหรับอุปกรณ์ MIDP เพราะช่วยให้ผู้พัฒนาสามารถสร้างหน้าจอได้อย่างยืดหยุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการนำ MIDP มาใช้งาน ได้ระบุการวางโครงร่างของ Item ใน Form สำหรับอุปกรณ์ต่างๆ ไปไว้ว่าต้องเป็นแนวตั้ง Item ที่โฟกัสได้เช่น TextField, DateField, Gauge หรือ ChoiceGroup จะขึ้นบรรทัดใหม่ เสมอ สำหรับ StringItem และ ImageItem ซึ่งไม่เกี่ยวข้องกับการโต้ตอบของผู้ใช้จะวางในแนวนอน นอกเสียจากว่าสตริงหรือคำสั่ง Layout ของ ImageItem จะกำหนดให้ขึ้นบรรทัดใหม่ ข้อความใน StringItem หรือ ภาพใน ImageItem จะถูกตัดขึ้นบรรทัดใหม่(กรณีข้อความ) หรือถูกคลิปบางส่วนออก (กรณีภาพ) เพื่อให้พอดีกับความกว้างของหน้าจอ หากจำเป็นต้องเลื่อนจอ ทำได้เฉพาะในแนวตั้งเท่านั้น

5.6 อีเวนต์ระดับสูงและการจัดการอีเวนต์

โมเดลการจัดการอีเวนต์ระดับสูงเป็นแบบ delegation-based เหมือนกับใน AWT ของ J2SE ในโมเดลนี้ประกอบด้วย 2 ส่วนคือ แหล่งกำเนิดอีเวนต์ และตัว event listener แหล่งกำเนิดอีเวนต์จะสร้างอีเวนต์แล้วส่งไปยัง event listener จากนั้นเมื่อบริษัทที่จัดการอีเวนต์ของ listener ก็จะมีประมวลผลอีเวนต์นั้น

ออบเจกต์ของส่วนติดต่อกับผู้ใช้ใน MIDP จะใช้อีเวนต์ระดับสูง 2 ตัวคือ อีเวนต์ command และ ItemStateChanged ดังนั้นจึงมีอีเวนต์ listener 2 ตัวคือ commandListener และ ItemStateListener ตามลำดับ

5.6.1 Command เป็นโครงสร้างที่เก็บความหมายของแอคชันหรืออีเวนต์ แอคชันจริง (หรือการจัดการอีเวนต์) ซึ่งเกิดเมื่อมี command เกิดขึ้นได้กำหนดไว้ในอินเทอร์เฟซของ CommandListener เมื่อ command เริ่มต้นจะเกิดอีเวนต์แล้วส่งไปยังออบเจกต์ CommandListener ที่รีจิสเตอร์ไว้ ออบเจกต์ Command มีข้อมูล 3 อย่างคือ เลขเบต , ชนิด และ ลำดับความสำคัญ

5.6.2 ItemStateChanged เมื่อ Form แสดงผลบนจอ ผู้ใช้สามารถเลื่อน โฟกัสจาก Item หนึ่งไปยังอีก Item หนึ่งภายใน Form ได้ โดยไม่เกินอีเวนต์ให้แอปพลิเคชันสามารถตรวจเห็นได้แต่หากผู้ใช้แก้ไขเปลี่ยนแปลงสถานะของ Item ที่ได้ตอบโต้ซึ่งอยู่ภายใน Form จะทำให้เกิด ItemStateChanged ขึ้น แต่ถ้าเป็นการเปลี่ยนแปลงค่า Item โดยแอปพลิเคชันด้วยเมธอด setString() ก็จะไม่ทำให้เกิดอีเวนต์ ItemStateChanged

บทที่ 6

การออกแบบเกม

6.1 การออกแบบลักษณะเกม

เริ่มออกแบบโดยการคิดแนวเกม มุมมองที่ผู้เล่นจะเห็นการดำเนินเนื้อเรื่องในเกม ข้อจำกัดต่างๆ ของเกมก่อนแล้วค่อยเลือกเนื้อหาของการเล่นเนื้อเรื่องในเกม ซึ่งในที่นี้ออกแบบให้เกมมีลักษณะเป็นเกมแนว Shooting โดยที่ให้มุมมองของผู้เล่นนั้นมองเห็นแบบ 2 มิติ

6.1.1 รูปแบบของเกมที่ออกแบบ

โดยตัวละครของเราจะเป็นยาน ต้องยิงศัตรูที่จะบุกเข้ามาโจมตี และต้องระวังไม่ให้ศัตรูมาโดนยานของเรา ถ้าศัตรูมาโดนยานของเราจะทำให้จบเกม

6.2 การทำงานของคลาส ต่างๆ

6.2.1 Galaxy

ในส่วนนี้จะมีการทำงานในส่วนของคลาสหลักนั่นก็คือคลาส Galaxy ซึ่งทำงานในส่วนของ MIDlet ทั้งหมด ที่จะทำงานว่าต้องการจะให้ปัจจุบันบนหน้าจอ โทรศัพท์มือถือนั้นจะโชว์หน้าจอในรูปแบบ FORM หรือ CANVAS หน้าใด

```
public class Galaxy extends MIDlet implements CommandListener
```

```
{ ..... }
```

6.2.2 GameThread

ควบคุมการทำงานของคลาส GalaxyCanvas สืบทอดมาจากคลาส Thread ในเมธอด run() ประกอบไปด้วยลูปหลักซึ่งจะสร้างคลาส GalaxyCanvas ขึ้นมาคอยตรวจสอบว่าปุ่มไหนถูกกดและเปลี่ยนแปลงรูปภาพฟีกให้เคลื่อนตามกับปุ่มกดไปเรื่อยๆจนกว่าเกมจะถูกหยุดพักชั่วคราวหรือจบ

```
public class GameThread extends Thread
```

```
{ ..... }
```

6.2.3 GalaxyCanvas

การทำงานของคลาสนี้จะทำงานด้านเกี่ยวกับการวาดภาพทั้งหมดในตัวเกมหลังจากการเข้าจากเล่นเกมเรียบร้อยแล้ว รวมการทำงานของคิดต่อบนปุ่มกดต่างๆของผู้เล่นที่กดจาก โทรศัพท์มือถือ ซึ่งการทำงานของคลาสนี้จะใช้ LOW LEVEL คือการจัดการด้านกราฟฟีกและการคิดต่อจากผู้ใช้ด้วยการเขียนโปรแกรมจัดการสิ่งต่างๆเองทั้งหมดโดยผู้พัฒนาโปรแกรม

```
public class JumpCanvas extends javax.microedition.lcdui.game.GameCanvas
```

```
{ ..... }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.4 GalaxyManager

ทำหน้าที่คอยจัดการส่วนของกราฟฟิกต่างๆ สืบทอดมาจากคลาส LayerManager

```
public class GalaxyManager extends javax.microedition.lcdui.game.LayerManager
{
    .....
}
```

6.2.5 Ship,Enemy,Bullet,Boss

ทำการเก็บรูปภาพและจัดการกับการเคลื่อนไหว สืบทอดมาจากคลาส Sprite ซึ่งใช้สร้างออบเจกต์ที่เคลื่อนไหวอิสระบนพื้นหลังหรือออบเจกต์อื่น

```
public class Ship extends Sprite
{
    .....
}

public class Enemy extends Sprite
{
    .....
}

public class Bullet extends Sprite
{
    .....
}

public class Boss extends Sprite
{
    .....
}
```

6.2.6 Background

จะเห็นได้ว่าคลาสนี้ก็จะทำการเก็บรูปภาพต่างที่เป็นภาพพื้นหลังในฉาก โดยการแยกคลาสนี้ออกมาเก็บภาพเพื่อเป็นการแยกส่วนของการโหลดภาพเพื่อให้ลดหน่วยความจำให้ลดน้อยลงในการเก็บภาพของเกมไว้เพื่อที่จะแสดงผล สืบทอดมาจากคลาส TiledLayer

```
public class Background extends TiledLayer
{
    .....
}
```

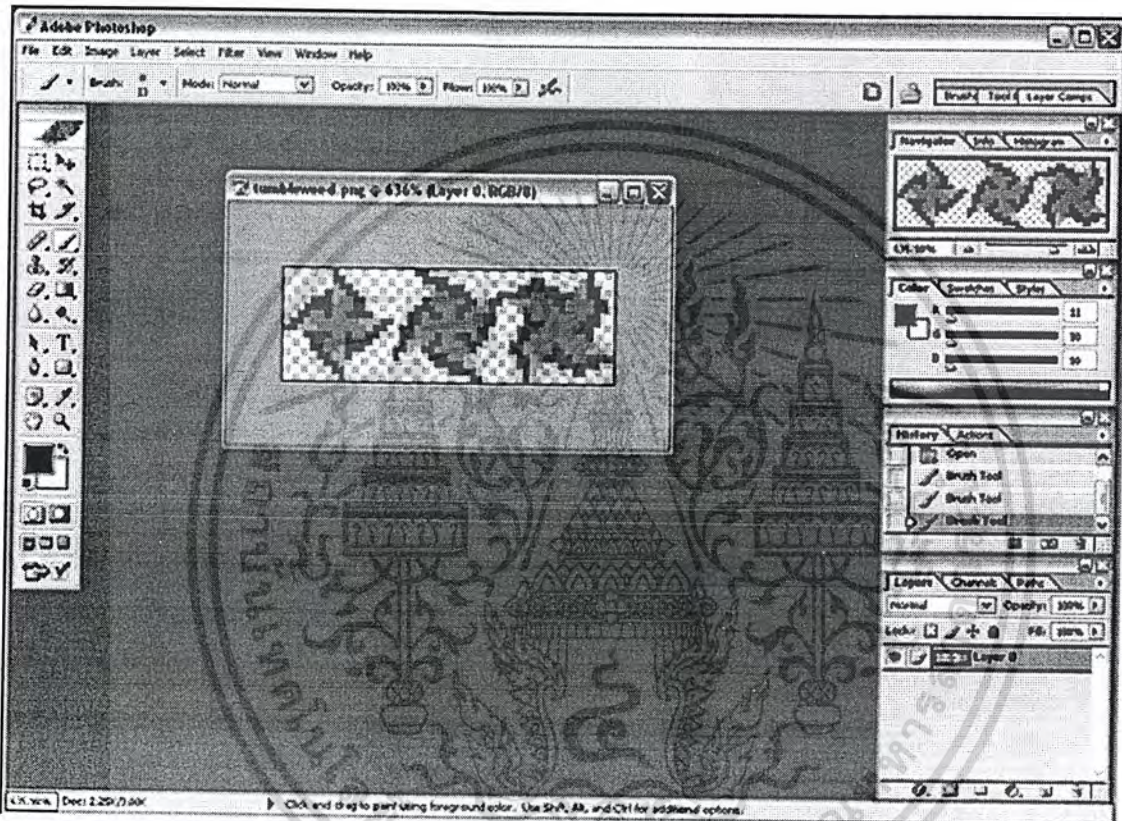
6.3 เทคนิคต่างๆที่นำมาใช้ในเกม

6.3.1 เทคนิคการวาดแผนที่

เราจะทำการสร้างรูปขึ้นมาให้อยู่ในรูปแบบ *.png เพื่อที่จะนำมาเรียกใช้ในโปรแกรมได้

6.3.2 เทคนิคการทำภาพแอนิเมชันของตัวละคร

ในส่วนของสัคตรูนั้น จะมีการเคลื่อนไหวของตัวละครด้วย การจะทำให้รูปมีการเคลื่อนไหวนั้นต้องทำการสร้างภาพขึ้นมาเป็นเฟรมๆแล้วทำการวาดภาพทีละเฟรม



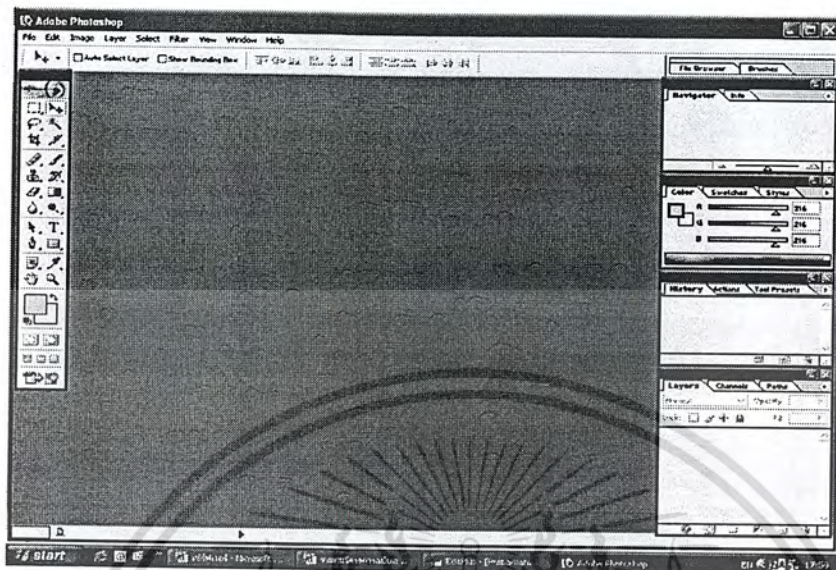
รูปที่ 6.3 ภาพในแต่ละเฟรม

6.4 โปรแกรมที่ใช้ในการพัฒนา

6.4.1 ADOBE PHOTOSHOP

เป็นโปรแกรมที่ใช้ในการออกแบบและตัดต่อ ตกแต่งภาพต่างๆ ในเกมเช่น ภาพฉากการต่อสู้ ภาพตัวละคร ภาพสัคตรู รวมทั้งการออกแบบแอนิเมชันในเกม ซึ่งจะใช้โปรแกรมนี้ในการออกแบบภาพเหล่านี้โดยที่ลักษณะของภาพนั้นมีลักษณะเป็นมูมมิง มิติ การใช้โปรแกรมนั้นก็เป็นโปรแกรมที่มีความนิยมนิยมสามารถใช้ได้ง่ายรวมทั้งหาหนังสือเพื่อที่จะศึกษาเพื่อค้นคว้าในการที่ใช้งานที่จำซ้อกว่านี้ได้

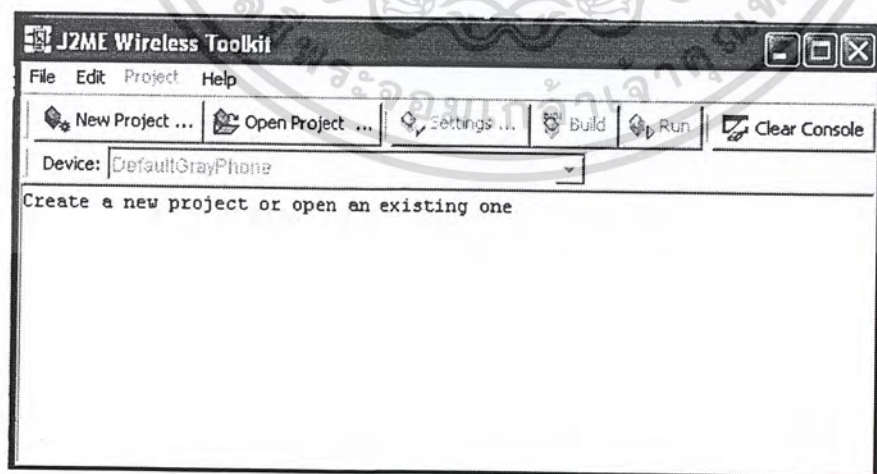
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.7 โปรแกรม Adobe PhotoShop

6.4.2 J2ME Wireless Toolkit

เป็นโปรแกรมอิมูเลเตอร์ที่ใช้ในการสร้างโปรเจกต์ของแอปพลิเคชันที่เขียนด้วยภาษา J2ME สามารถใช้คอมไพล์ และรันแอปพลิเคชันนั้นๆ โดยสามารถเลือกอุปกรณ์ที่ต้องการจำลองการทำงานได้ซึ่งมีทั้งโทรศัพท์มือถือรุ่นต่างๆ หรืออาจเป็นพีดีเอก็ได้ โดยเราสามารถนำอิมูเลเตอร์ของอุปกรณ์ที่เราต้องการมาทำการใส่เข้าไปก็ได้โดยการใส่เข้าไปในโฟลเดอร์ อุปกรณ์ เท่านั้นก็สามารถเลือกใช้งานได้แล้ว



รูปที่ 6.8 โปรแกรม J2ME Wireless Toolkit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.9 อุปกรณ์จำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การทดสอบการทำงาน

7.1 ผลการทดสอบการทำงาน

เมื่อเริ่มต้นเปิดแอปพลิเคชัน จะปรากฏหน้าจอ ดังรูปที่ 7.1

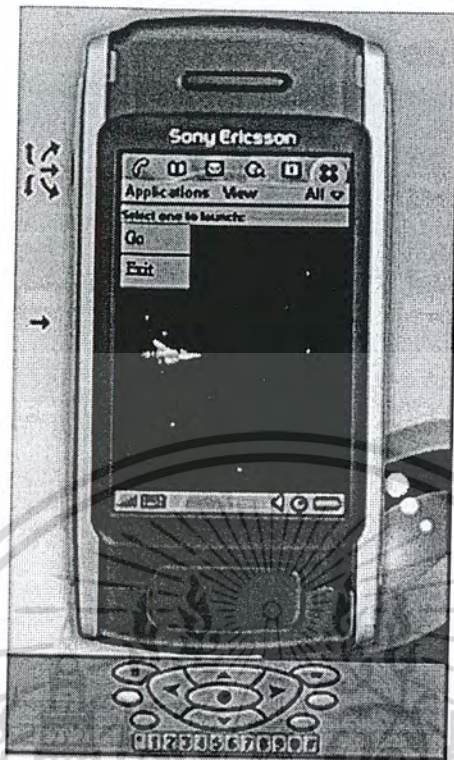


รูปที่ 7.1 หน้าจอเมื่อเริ่มต้นเลือกเข้าสู่เกม

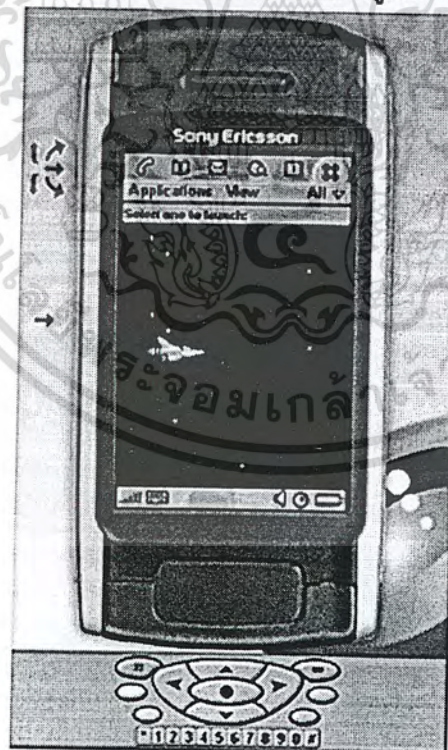
เมื่อเข้าสู่เกมจะปรากฏหน้าจอดังรูป 7.2 ซึ่งจะมี LIST ให้เลือก 2 ตัวเลือกนั้นก็คือ

- Go คือ การเริ่มต้นเล่นเกม
- Exit คือ ออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.2 หน้าจอเมื่อเริ่มต้นเข้าสู่เกม



รูปที่ 7.3 หน้าจอเมื่อเลือก Go เพื่อเล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มเกมแล้วเราสามารถยิงลูกกระสุนและบังคับให้ยานขับในจอภาพ และจะมีศัตรูออกมา โดยเราต้องยิง และต้องระวังไม่ให้ยานของเราโดนศัตรู



รูปที่ 7.4 หน้าจอเมื่อกดยิง



รูปที่ 7.5 หน้าจอเมื่อมีศัตรู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อศัตรูโคนฮานของเราจะเป็นการจบเกม



รูปที่ 7.6 หน้าจอเมื่อจบเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

บทวิจารณ์และสรุป

8.1 สรุปผลการทดลอง

เกมสามารถทำตามรูปแบบของเกมที่ออกไว้ได้ทั้งหมด แต่ถึงการทำงานจะเสร็จสมบูรณ์ตามที่คาดไว้นั้นก็ยังมีปัญหาที่ควรคำนึงถึงนั่นก็คือเนื่องจากการเขียนโปรแกรมใน อุปกรณ์ ขนาดเล็กนั้นมีข้อจำกัดทางด้านความเร็วและหน่วยความจำ ดังนั้นการเขียนภาษา J2ME จึงมีข้อจำกัดต่างๆมากกว่าภาษาจาวาธรรมดา ทำให้การเขียนเป็นไปได้ยากขึ้นกว่าธรรมดาที่ภาษาเขียนบนเครื่องคอมพิวเตอร์

ปัญหาสำคัญที่เกิดขึ้นอีกประการก็คือ ขนาดของหน้าจอการแสดงผลนั้นไม่ว่าจะเป็นปาล์มรุ่นต่างๆ หรือมือถือแต่ละรุ่นก็จะมีหน้าจอการแสดงผลไม่เท่ากันดังนั้นการพัฒนา แอปพลิเคชัน นั้นต้องเลือกพัฒนา จำกัดว่าควรใช้กับ อุปกรณ์ ยี่ห้ออะไรรุ่นอะไรไปเลยไปและการพัฒนาจะต้องทำการพัฒนาใน อิมูเลเตอร์ ของ อุปกรณ์ ยี่ห้อนั้นรุ่นนั้น แต่ก็อาจเกิดปัญหาที่ว่าสามารถใช้ใน อิมูเลเตอร์ ได้แค่พอนามารันใน อุปกรณ์ จริง กลับรันไม่ได้ ทำให้ต้องเปลี่ยนรุ่นในการพัฒนาดังนั้น โปรแกรมที่เขียนเสร็จก็ต้องนำมาปรับปรุงอีกเพื่อให้ขนาดการแสดงผลต่างๆเหมาะสมกับรุ่นที่เราจะเปลี่ยนมาใช้

8.2 แนวทางการพัฒนาและประยุกต์ใช้งานร่วมกับงานอื่น

ในส่วนของโปรแกรมที่สร้างสามารถแบ่งได้ออกเป็น 2 ส่วน

1. ส่วนของ High-Level API อินเทอร์เฟซระดับสูงซึ่งในส่วนนี้เป็นการติดต่อกับผู้เล่นเกมในตอนเริ่มแรกเพื่อที่จะเริ่มเกม ในส่วนนี้สามารถนำไปประยุกต์ใช้งานกับการป้อนข้อมูลต่างๆของผู้ใช้งาน แอปพลิเคชัน นั้นๆเพื่อให้นำข้อมูลเหล่านั้นไปใช้ประโยชน์ต่อไป

2. ส่วนของ Low-Level API อินเทอร์เฟซระดับต่ำเป็นซึ่งเป็นส่วนหลักของเกมในการทำ animation ต่างๆของเกมซึ่งในส่วนนี้จะสามารถนำไปประยุกต์ต่อไปในการที่จะทำด้านกราฟิกซึ่งในส่วนนี้ผู้เขียน แอปพลิเคชัน สามารถออกแบบทุกอย่างเองได้ตามใจชอบ

ในส่วนของตัวเกมนั้นถือว่าสมบูรณ์ตามเป้าหมายที่ได้คาดหวังไว้ แต่ถ้าต้องการที่จะพัฒนาต่อโดย ส่วนตัวของเกมเพื่อความสนุกหรือความน่าสนใจให้มากขึ้น ควรเพิ่มทางด้านรายละเอียดของตัวเกมเช่น ควรมีแผนที่เพิ่มขึ้น , เพิ่มของพิเศษต่างๆเพื่อช่วยในการเล่นให้สนุกและหลากหลาย, เพิ่มเสียงในการเล่นหรือ อาจเพิ่มตัวละครให้มากขึ้น ดังนั้นการพัฒนาค่อยๆที่สุดแล้วแต่เทคนิคของแต่ละผู้พัฒนาซอฟต์แวร์ ว่าต้องการให้เป็นแนวใดต่อไปเพื่อที่จะให้เกิดการพัฒนาในการเขียน โปรแกรมเพื่อนำมาใช้กับชีวิตจริงให้มากที่สุด และสามารถเป็นแนวทางในการพัฒนาการเขียน โปรแกรมต่างๆได้อีกด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ภาษาไทย

- [1] หนังสือ J2ME wireless java programming with Java 2 Micro Edition
แต่งโดย Yu Feng, Jun Zhu
- [2] เก่ง J2ME ให้ครบสูตร สำนักพิมพ์ WITTY GROUP แต่งโดย ทรงเกียรติ ภาวดี
- [3] เขียนโปรแกรมและเกมบนมือถือ สำนักพิมพ์ Infopress แต่งโดย กาญจนา ดันวิสุทธิ์
เว็บไซต์
- [4] MrPalm website “www.mrpalm.com” (ค้นคว้าเกี่ยวกับการอุปกรณ์ไร้สาย)
- [5] Midlet “www.MidletZone.com” (ตัวอย่าง MIDLET)
- [6] Wireless Toolkit “www.java.sun.com” (ดาวน์โหลดอิมเพลเตอร์)
- [7] <http://java.sun.com/j2me/>
- [8] <http://java.sun.com/product/cldc/>
- [9] <http://java.sun.com/products/javaphone/>
- [10] <http://sun.java.com/products/j2mewtoolkit/>
- [11] <http://java.sun.com/product/midp/>

ภาคผนวก ก
แสดงคลาสไลบรารีของ CLDC

Java.lang

class java.lang.Object
class java.lang.Boolean
class java.lang.Byte
class java.lang.Character
class java.lang.Class
class java.lang.Integer
class java.lang.Long
class java.lang.Math
class java.lang.Runtime
class java.lang.Short
class java.lang.String
class java.lang.StringBuffer
class java.lang.System
class java.lang.Thread (implements java.lang.Runnable)
class java.lang.Throwable
class java.lang.Error
class java.lang.VirtualMachineError
class java.lang.OutOfMemoryError
class java.lang.Exception
class java.lang.ClassNotFoundException
class java.lang.IllegalAccessException
class java.lang.InstantiationException
class java.lang.InterruptedExecution
class java.lang.RuntimeException
class java.lang.ArithmeticException
class java.lang.ArrayStoreException
class java.lang.ClassCastException
class java.lang.IllegalArgumentException

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

class java.lang.IllegalThreadStateException
class java.lang.NumberFormatException
class java.lang.IllegalMonitorStateException
class java.lang.IllegalStateException
class java.lang.IndexOutOfBoundsException
class java.lang.ArrayIndexOutOfBoundsException
class java.lang.StringIndexOutOfBoundsException
class java.lang.NegativeArraySizeException
class java.lang.NullPointerException
class java.lang.SecurityException

```

java.io

```

class java.lang.Object
class java.io.InputStream
class java.io.ByteArrayInputStream (implements java.io.DataInput)
class java.io.OutputStream
class java.io.ByteArrayOutputStream
class java.io.DataOutputStream (implements java.io.DataOutput)
class java.io.PrintStream
class java.io.Reader
class java.io.InputStreamReader
class java.io.Throwable
class java.io.Exception
class java.io.IOException
class java.io.EOFException
class java.io.InterruptedIOException
class java.io.UnsupportedEncodingException
class java.io.UTFDataFormatException
class java.io.Write
class java.io.OutputStreamException
Interface java.io.DataInput
Interface java.io.DataOutput

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Java.util

```

class java.lang.Object
class java.util.Calendar
class java.util.Date
class java.util.Hashtable
class java.util.Random
class java.lang.Throwable
class java.lang.Exception
class java.lang.RuntimeException
class java.util.EmptyStackException
class java.util.NoSuchElementException
class java.util.Timer
class java.util.TimerTask(implements java.lang.Runnable)
class java.util.TimeZone
class java.util.Vector
class java.util.Stack

```

javax.microedition.io

```

class java.lang.Object
class javax.microedition.io.Connector
class java.lang.Throwable
class java.io.IOException
class javax.microedition.io.ConnectionNotFoundException
interface javax.microedition.io.Connection
interface javax.microedition.io.DatagramConnection
interface javax.microedition.io.InputConnection
interface javax.microedition.io.StreamConnection
(also extends javax.microedition.io.OutputConnection)
interface javax.microedition.io.ContentConnection
interface javax.microedition.io.OutputConnection
interface javax.microedition.io.StreamConnection
(also extends javax.microedition.io.InputConnection)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

interface javax.microedition.io.ContentConnection

interface javax.microedition.io.StreamConnectionNotifier

interface javax.io.DataInput

interface javax.microedition.io.Datagram(also extends java.io.DataOutput)

interface javax.io.DataOutput

interface javax.microedition.io.Datagram(also extends java.io.DataInput)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
 คลาสไลบรารีของ MIDP

Javax.microedition.lcdui

```

class java.lang.Object
class javax.microedition.lcdui.AlertType
class javax.microedition.lcdui.Command
class javax.microedition.lcdui.Display
class javax.microedition.lcdui.Displayable
class javax.microedition.lcdui.Canvas
class javax.microedition.lcdui.Screen
class javax.microedition.lcdui.Alert
class javax.microedition.lcdui.Form
class javax.microedition.lcdui.List(implements class javax.microedition.lcdui.Choice)
class javax.microedition.lcdui.TextBox
class javax.microedition.lcdui.Font
class javax.microedition.lcdui.Graphics
class javax.microedition.lcdui.Image
class javax.microedition.lcdui.Item
class javax.microedition.lcdui.ChoiceGroup(implements class javax.microedition.lcdui.Choice)
class javax.microedition.lcdui.DateField
class javax.microedition.lcdui.Gauge
class javax.microedition.lcdui.ImageItem
class javax.microedition.lcdui.StringItem
class javax.microedition.lcdui.TextField
class javax.microedition.lcdui.Ticker
Interface javax.microedition.lcdui.Choice
Interface javax.microedition.lcdui.CommandListener
Interface javax.microedition.lcdui.ItemStateListener
  
```

Javax.microedition.rms

```

class java.lang.Object
class javax.microedition.rms.RecordStore
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
class java.lang.Throwable
class java.lang.Exception
class java.microedition.rms.RecordStoreException
class java.microedition.rms.InvalidRecordIDException
class java.microedition.rms.RecordStoreFullException
class java.microedition.rms.RecordStoreNotFoundException
class java.microedition.rms.RecordStoreNotOpenException
Interface javax.microedition.rms.RecordComparator
Interface javax.microedition.rms.RecordEnumeration
Interface javax.microedition.rms.RecordFilter
Interface javax.microedition.rms.RecordListener
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้