เกมวิ่งแข่งบนมือถือ

Running Bot Mobile Game

นายณัฐพล เกงขุนทด รหัส 47360110

นางสาววิมลรัตน์ ไพโรจน์ รหัส 47360185

นายกิตติพัฒน์ บุญพร้อม รหัส 47361829

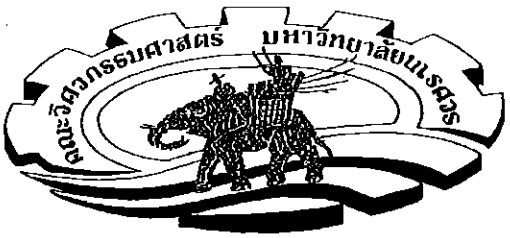
ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 25 W.A. 2553
เลขทะเบียน ท 302 3312 e.2
เลขเรียกหนังสือ ปร
มหาวิทยาลัยนเรศวร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2550



ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	เกมวิ่งแข่งบนมือถือ			
ผู้ดำเนินโครงงาน	นายณัฐพล	เกงขุนทด	รหัส	47360110
	นางสาววิมลรัตน์	ไพโรจน์	รหัส	47360185
	นายกิตติพัฒน์	บุญพร้อม	รหัส	47361829
อาจารย์ที่ปรึกษา	นายเศรษฐา	ตั้งค้าวานิช		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2550			

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(นายเศรษฐา ตั้งค้าวานิช)

.....กรรมการ
(ดร.ไพศาล มุณีสว่าง)

.....กรรมการ
(ดร.พนมชวัญ ริยะมงคล)

หัวข้อโครงงาน	เกมวิ่งแข่งบนมือถือ			
ผู้ดำเนินโครงงาน	นายณัฐพล	เกงขุนทด	รหัส	47360110
	นางสาววิมลรัตน์	ไพโรจน์	รหัส	47360185
	นายกิตติพัฒน์	บุญพร้อม	รหัส	47361829
อาจารย์ที่ปรึกษา	นายเศรษฐา	ตั้งค้าวานิช		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2550			

บทคัดย่อ

โครงงานนี้พัฒนาเกมวิ่งบนมือถือโดยใช้หลักการของปัญญาประดิษฐ์ทำให้ตัวละครในเกมสามารถตอบสนองต่อสิ่งแวดล้อมที่พบได้ อย่างไรก็ตามในช่วงเริ่มต้นเกม ตัวละครแสดงการกระทำได้ไม่เต็มที่ แต่เมื่อตัวละครได้ผ่านอุปสรรคทั้งจากการโจมตี การหลบหลีกสิ่งกีดขวางมาแล้ว ทำให้ตัวละครพัฒนามากยิ่งขึ้น เกมนี้พัฒนาด้วยภาษาจาวาบนเพลตฟอร์ม J2ME (Java 2 Micro Edition) เพื่อให้ใช้ได้กับโทรศัพท์มือถือ จากผลที่ได้แสดงให้เห็นถึงความสามารถของตัวละครที่พัฒนาขึ้นในการผ่านสิ่งกีดขวาง

Project Title	Running Bot Mobile Game.		
Name	Mr. Nattapon	Kengkhuntod	ID. 47360110
	Miss Wimonrat	Phairot	ID. 47360185
	Mr. Kittipat	BunPromt	ID. 47361829
Project Advisor	Mr. Sattha	Tangkawanit	
Major	Computer Engineering.		
Department	Electrical and Computer Engineering.		
Academic Year	2007		

.....

ABSTRACT

This project developed running game on mobile by principle of Artificial Intelligence (AI) that makes characters in game respond to condition. However, in the beginning of the game, characters cannot act smoothly, but after being attacked, passing the obstacles, they can perform efficiently. This game is developed on platform J2ME (Java 2 Micro Edition) for using on mobile. From result show us the ability of characters to pass the obstacles.

กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์เศรษฐา ตั้งค้าวานิช ที่กรุณาเป็นอาจารย์ที่ปรึกษา และอีกท่านที่ได้สละเวลาอันมีค่า ดร.สุรเดช จิตประไพกุลศาล ให้คำปรึกษา คำแนะนำและให้แนวคิดต่าง ๆ เพื่อนำมาพัฒนาเกม ขอขอบพระคุณอาจารย์ทุกท่านที่กรุณาสละเวลาอันมีค่าเพื่อชี้แนะแนวทางและให้คำปรึกษา อีกทั้งเพื่อน ๆ ที่กรุณาช่วยทดสอบเกม และให้ข้อเสนอแนะเพื่อการปรับปรุงเกมให้เหมาะสมกับความต้องการของผู้เล่นมากที่สุด และพี่อารักษ์ คเชนทร์ ที่ช่วยให้คำแนะนำต่าง ๆ เกี่ยวกับการเขียนโปรแกรมด้วย J2ME ซึ่งเป็นส่วนสำคัญที่ทำให้โครงงานนี้สำเร็จได้ สุดท้ายขอขอบพระคุณบิดามารดา ที่คอยให้กำลังใจ และสนับสนุนการศึกษาของพวกเรามาตลอด จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

มหาวิทยาลัยนเรศวร

สารบัญ

หน้า

บทคัดย่อ ..... ก

ABSTRACT ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบข่ายของโครงงาน 2

1.4 ขั้นตอนการดำเนินการ 2

1.5 แผนการดำเนินงาน 2

1.6 ผลที่คาดว่าจะได้รับ 2

1.7 งบประมาณที่ใช้ 3

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง 4

2.1 ปัญญาประดิษฐ์ 4

2.1.1 นิยามของปัญญาประดิษฐ์ 4

2.1.2 Specifying the task environment 6

2.1.3 Properties of task environments 7

2.1.4 Agent Program 7

2.2 J2ME Wireless Toolkit 8

2.2.1 ไฟล์ *.jar 9

2.2.2 ไฟล์ *.jad 9

บทที่ 3 วิธีการดำเนินงาน 10

3.1 สตอรี่บอร์ด (Story Board) 10

สารบัญ(ต่อ)

หน้า

3.2 การออกแบบเกม 11
3.2.1 ความสามารถของตัวละคร 11
3.2.2 ข้อมูลพื้นฐานของตัวละครและคะแนนการแข่งขัน 19
3.2.3 รูปภาพและส่วนการแสดงผล 19

บทที่ 4 ผลการทดลอง 24
4.1 วิธีการเล่นเกม 24
4.2 การทดสอบเกม 25

บทที่ 5 บทสรุป 31
5.1 วิเคราะห์ผลการทดลอง 31
5.2 สรุปผลการทดลอง 31
5.3 ข้อเสนอแนะ 31

เอกสารอ้างอิง 32
ภาคผนวก ก. ตัวอย่างโปรแกรม 33
ประวัติผู้เขียนโครงงาน 56

สารบัญตาราง

ตารางที่ หน้า

ตารางที่ 3.1 รายละเอียด PEAS....11

ตารางที่ 3.2 Task environment 12

ตารางที่ 4.1 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 1 25

ตารางที่ 4.2 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 2 26

ตารางที่ 4.3 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 3 26

ตารางที่ 4.4 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 4 27

ตารางที่ 4.5 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 5 28

ตารางที่ 4.6 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 6 29

ตารางที่ 4.7 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 7 29

สารบัญรูป

รูปที่ หน้า

รูปที่ 2.1 Simple reflex agents 8

รูปที่ 2.2 โฟลเดอร์ที่เก็บ MIDlet Suite 8

รูปที่ 2.3 ตัวอย่างไฟล์ *.mf 9

รูปที่ 2.4 ตัวอย่างไฟล์ *.jad 9

รูปที่ 3.1 สตอรี่บอร์ดของเกมวิ่งแข่งบนมือถือ 10

รูปที่ 3.2 โครงสร้างของเกม 11

รูปที่ 3.3 แผนผังการทำงาน (Flow chart) แสดงสิ่งที่ตัวละครคิดเมื่อวิ่งปกติ 12

รูปที่ 3.4 แผนผังการทำงาน (Flow chart) แสดงสิ่งที่ตัวละครทำเมื่อมีคู่แข่งเข้ามาโจมตี 13

รูปที่ 3.5 แผนผังการทำงาน (Flow chart) แสดงการหลบสิ่งกีดขวาง 14

รูปที่ 3.6 แผนผังการทำงาน (Flow chart) แสดงการกระโดดผ่านสิ่งกีดขวาง 15

รูปที่ 3.7 แผนผังการทำงาน (Flow chart) แสดงการไล่โจมตีและฟันคู่แข่ง 16

รูปที่ 3.8 แผนผังการทำงาน (Flow chart) แสดงการเลือกการหลบ 17

รูปที่ 3.9 แผนผังการทำงาน (Flow chart) เมื่อคู่ต่อสู้อยู่ด้านหลังลู่บนสามารถโจมตีได้ 18

รูปที่ 3.10 แผนผังการทำงาน (Flow chart) เมื่อคู่ต่อสู้อยู่ด้านหลังลู่ล่างสามารถโจมตีได้ 19

รูปที่ 3.11 หน้าตัวละคร 20

รูปที่ 3.12 พื้น (ลู่วิ่ง) 20

รูปที่ 3.13 วิวท้องฟ้า 20

รูปที่ 3.14 วิวต้นไม้ 21

รูปที่ 3.15 ตัวละคร 21

รูปที่ 3.16 สิ่งกีดขวาง 21

รูปที่ 3.17 อาวุธ 21

รูปที่ 3.18 ฉากเริ่มเกม 22

รูปที่ 3.19 ฉาก loading 22

รูปที่ 3.20 ฉากสนทนา 22

รูปที่ 3.21 ฉากเริ่มการแข่งขัน 23

รูปที่ 3.22 ฉากระหว่างการแข่งขัน 23

รูปที่ 3.23 ฉากแสดงคะแนน 23

รูปที่ 4.1 แสดงปุ่ม Soft Key และ Key number 24

รูปที่ 4.2 กราฟแสดงค่าการหลบได้ หลบไม่ได้ และไม่หลบเลยเป็นเปอร์เซ็นต์..... 30

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เกมที่รู้จักกันในปัจจุบันมีหลายรูปแบบ ซึ่งหนึ่งในนั้นเป็นเกมที่ผู้เล่นต้องแข่งขันกับเครื่องที่ตั้งโปรแกรมไว้แล้ว เป็นเกมที่ต้องอาศัยทักษะในการคิดหาวิธีทำสิ่งที่เกมกำหนดให้เร็วที่สุด ผู้ที่เล่นจะเพลิดเพลินในการคิดและเล่นอย่างสนุกสนาน โดยที่เกมในลักษณะที่ตัวเกมสามารถพัฒนาความสามารถได้เองนั้นไม่มีให้เห็นมากนัก เกมแนวนี้หลังจากการเล่นมักจะต้องมีการบันทึก ทำให้เปลืองหน่วยความจำ ทำให้ไม่สามารถลงโปรแกรมอื่นหรือเกมอื่นได้มากเท่าที่ต้องการ เกมในมือถือของเครื่องรุ่นใหม่ ๆ ที่ออกมาในปัจจุบัน มีความแตกต่างจากเกมในอดีตที่ได้เพิ่มฟังก์ชัน สีสัน หรือแม้กระทั่งการดำเนินเรื่องของเกมเองก็มีอยู่หลากหลาย ถ้ารุ่นไหนไม่มีเกมแล้วรุ่นนั้นมักจะมียอดขายที่น้อยกว่ารุ่นที่มีเกม ในปัจจุบันผู้ใช้ส่วนใหญ่ โดยเฉพาะกลุ่มวัยรุ่นมักเน้นไปที่เกมที่ติดมากับเครื่องหรือรุ่นที่สามารถดาวห์โหลด (Download) เกมได้จากแหล่งต่าง ๆ ได้รับความนิยมจนทำให้ผู้ผลิตโทรศัพท์มือถือยี่ห้อหนึ่ง ผลิตมือถือที่เน้นเกมโดยเฉพาะ เพื่อเอาใจคนในกลุ่มนี้

การสร้างเกมหรือโปรแกรมประยุกต์อื่นๆในอุปกรณ์สื่อสารแบบไร้สาย มีเทคโนโลยีหนึ่งในตระกูลจาวา คือ J2ME (Java 2 Micro Edition) สามารถรันและทำงานได้ในอุปกรณ์ที่มีขนาดเล็กและหน่วยความจำที่จำกัด ซึ่งบริษัท Sun Microsystems เป็นผู้เริ่มในการพัฒนาเทคโนโลยีนี้ โดยมี class, packet, และ API พิเศษสำหรับการติดต่อและทำงานกับอุปกรณ์ขนาดเล็กได้สะดวก

แนวเกมที่ออกมามากมาย ดังนี้ แนวเกมแอคชั่น ผจญภัย วางแผนและเกมกระดาน ซึ่งแต่ละเกมมีเสน่ห์ดึงดูดให้ผู้เล่นชวนหลงใหลต่างกันออกไป แบ่งตามความชอบของผู้เล่นเอง กลุ่มของเกมแอคชั่น เป็นเกมที่มีความท้าทาย เมื่อนำมาทำเป็นเกมบนโทรศัพท์มือถือ เนื่องจากเกมแอคชั่นเป็นแนวเกมที่ประกอบไปด้วย การคำนวณทางด้านวิชาฟิสิกส์ การคำนวณการเคลื่อนไหวพื้นฐาน การเคลื่อนที่รูปแบบต่าง ๆ การเคลื่อนไหวของภาพพื้นหลัง ความแม่นยำของการควบคุม ลักษณะของปัญญาประดิษฐ์ เสียงและภาพกราฟฟิคที่สวยงาม เป็นต้น จากข้อกำหนดของเกมลักษณะนี้อาจถูกมองว่าเป็นสิ่งที่ยากต่อการพัฒนาบนโทรศัพท์มือถือ แต่ใช่ว่าจะเป็นไปไม่ได้ เนื่องจากเกมแอคชั่น มีรูปแบบการเล่นที่สนุกสนาน ท้าทาย ทำให้เป็นที่ดึงดูดใจผู้เล่นเกมทั่วไป ดังนั้นถ้าได้สร้างเกมตามมาตรฐานและผสานรูปแบบตามความคิดของเราเองจะทำให้เกิดเกมที่สมบูรณ์แบบขึ้นมาได้

1.2 วัตถุประสงค์ของโครงงาน

1. สร้างเกมที่ตัวละครในเกมสามารถพัฒนาทักษะได้เองบนโทรศัพท์มือถือ
2. ช่วยในการฝึกสมาธิและทำให้ผู้เล่นได้รับความเพลิดเพลินในการเล่นเกม

1.3 ขอบข่ายของโครงงาน

เกมบนมือถือที่ตัวละครในเกมสามารถพัฒนาทักษะได้เองจากการเล่น โดยที่การเก็บข้อมูลเป็นไปอย่างจำกัด ซึ่งรันในมือถือระบบต่าง ๆ ที่มี MIDP 2.0 และรองรับเทคโนโลยี J2ME

1.4 ขั้นตอนการดำเนินการ

1. ออกแบบเกม
2. ศึกษาและรวบรวมข้อมูล
3. เขียนโปรแกรมและทดสอบ
4. ปรับปรุงและแก้ไข
5. ตรวจสอบการใช้งานจริง
6. จัดทำคู่มือโครงงาน

1.5 แผนการดำเนินงาน

กิจกรรม	2549		2550									
	11	12	1	2	3	4	5	6	7	8	9	10
1. ออกแบบเกม	←	—	—	→								
2. ศึกษาและรวบรวมข้อมูล	←	—	—	—	—	→						
3. เขียนโปรแกรมและทดสอบ		←	—	—	—	—	—	→				
4. ปรับปรุงและแก้ไข				←	—	—	—	—	—	—	→	
5. ตรวจสอบการใช้งานจริง						←	—	—	—	—	→	
6.จัดทำคู่มือโครงงาน						←	—	—	—	—	—	→

1.6 ผลที่คาดว่าจะได้รับ

1. กระบวนการคิดเพื่อสร้างวิธีการแก้ปัญหาเกม
2. ความรู้เกี่ยวกับการเขียนเกม

1.7 งบประมาณที่ใช้

- ค่าหนังสือ	1,200	บาท
- ค่าวัสดุสำนักงาน	600	บาท
- ค่าถ่ายเอกสารและจัดทำรูปเล่ม	1,200	บาท
รวม	3,000	บาท

หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้กล่าวถึงหลักการปัญญาประดิษฐ์คือเมื่อคู่แข่งในเกมหลบหรือกระโดดข้ามจะได้รับความสามารถเพิ่ม หากชนเข้ากับสิ่งกีดขวางทำให้ค่าชีวิตลดลง ตามหลักการของ เพื่อมาใช้ในสร้างการพัฒนาทักษะความสามารถของตัวละครในเกม และโปรแกรมที่ใช้คอมไพล์ (Compile) เกมวิ่งแข่งบนมือถือเพื่อสร้างแพคเกจ (Package) ให้สามารถนำไปติดตั้งบนโทรศัพท์มือถือได้คือ J2ME Wireless Toolkit

2.1 ปัญญาประดิษฐ์

ในเกมประกอบด้วยผู้เล่นและตัวละครในเกม ซึ่งความสามารถของตัวละครมาจากหลักการของปัญญาประดิษฐ์

ปัญญาประดิษฐ์ (Artificial Intelligence หรือ AI) หมายถึงความฉลาดเทียมที่ปรุงแต่งให้กับสิ่งที่ไม่มีชีวิต ปัญญาประดิษฐ์เป็นสาขาหนึ่งในด้านวิทยาการคอมพิวเตอร์ และวิศวกรรมเป็นหลัก รวมถึงศาสตร์ในด้านอื่น ๆ เช่น จิตวิทยา ปรัชญา หรือชีววิทยา สาขาปัญญาประดิษฐ์เป็นการเรียนรู้เกี่ยวกับกระบวนการการคิด การกระทำ การให้เหตุผล การปรับตัว หรือการอนุมาน และการทำงานของสมอง แนวคิดหลายๆ อย่างในศาสตร์นี้ได้มาจากการปรับปรุงเพิ่มเติมจากศาสตร์อื่น ๆ เช่น

- การเรียนรู้ของเครื่องมีเทคนิคการเรียนรู้ที่เรียกว่าการเรียนรู้ต้นไม้ตัดสินใจ มาประยุกต์เอาเทคนิคการอุปนัยของ จอห์น สจวร์ต มิลล์ นักปรัชญาชื่อดังของอังกฤษมาใช้
- เครือข่ายประสาทเทียมก็นำเอาแนวคิดของการทำงานของสมองของมนุษย์ใช้ในการแก้ปัญหาการแบ่งประเภทของข้อมูล และแก้ปัญหาอื่นๆ ทางสถิติ เช่น การวิเคราะห์ความถดถอยหรือ การปรับเส้นโค้ง

2.1.1 นิยามของปัญญาประดิษฐ์ [5]

มีคำนิยามของปัญญาประดิษฐ์มากมาย ซึ่งสามารถจัดแบ่งออกเป็น 4 ประเภทโดยมองใน 2 มิติ ได้แก่

- ระหว่าง นิยามที่เน้นระบบที่เลียนแบบมนุษย์ กับ นิยามที่เน้นระบบที่มีเหตุผล (แต่ไม่จำเป็นต้องเหมือนมนุษย์)
- ระหว่าง นิยามที่เน้นความคิดเป็นหลัก กับ นิยามที่เน้นการกระทำเป็นหลัก

ปัจจุบันงานวิจัยหลัก ๆ ของปัญญาประดิษฐ์มีแนวคิดเน้นเหตุผลเป็นหลัก เนื่องจากการนำปัญญาประดิษฐ์ไปประยุกต์ใช้แก้ปัญหา ไม่ได้อาศัยอารมณ์หรือความรู้สึกของมนุษย์มาช่วย

ตัดสินใจอย่างไรก็ตามนิยามทั้ง 4 ไม่ได้ต่างกันโดยสมบูรณ์ นิยามทั้ง 4 ต่างก็มีส่วนร่วมที่คาบเกี่ยวกันอยู่ นิยามดังกล่าวคือ

1. ระบบที่คิดเหมือนมนุษย์ (Systems that think like humans)
    a. [AI คือ] ความพยายามใหม่อันน่าตื่นเต้นที่จะทำให้คอมพิวเตอร์คิดได้ ... เครื่องจักรที่มีสติปัญญาอย่างครบถ้วนและแท้จริง ("The exciting new effort to make computers think ... machines with minds, in the full and literal sense." [Haugeland, 1985])
    b. [AI คือ กลไกของ]กิจกรรมที่เกี่ยวข้องกับความคิดมนุษย์ เช่น การตัดสินใจ การแก้ปัญหา การเรียนรู้ ("[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning." [Bellman, 1978])

<u>หมายเหตุ</u> ก่อนที่จะทำให้เครื่องคิดอย่างมนุษย์ได้ ต้องรู้ก่อนว่ามนุษย์มีกระบวนการคิดอย่างไร ซึ่งการวิเคราะห์ลักษณะการคิดของมนุษย์ เป็นศาสตร์ด้าน Cognitive science เช่น ศึกษาการเรียงตัวของเซลล์สมองในสามมิติ ศึกษาการถ่ายเทประจุไฟฟ้า และวิเคราะห์การเปลี่ยนแปลงทางเคมีไฟฟ้าในร่างกาย ระหว่างการคิด จนถึงปัจจุบัน (พ.ศ. 2548) เราก็ยังไม่รู้แน่ชัดว่า มนุษย์เรา คิดได้อย่างไร

2. ระบบที่กระทำเหมือนมนุษย์ (Systems that act like humans)
    a. [AI คือ] วิชาของการสร้างเครื่องจักรที่ทำงานในสิ่งที่อาศัยปัญญาเมื่อกระทำโดยมนุษย์ ("The art of creating machines that perform functions that requires intelligence when performed by people." [Kurzweil, 1990])
    b. [AI คือ] การศึกษาวิธีทำให้คอมพิวเตอร์กระทำในสิ่งที่มนุษย์ทำได้ดีกว่าในขณะนั้น ("The study of how to make computers do things at which, at the moment, people are better." [Rich and Knight, 1991])

<u>หมายเหตุ</u> การกระทำเหมือนมนุษย์ เช่น

- สื่อสารได้ด้วยภาษาที่มนุษย์ใช้ เช่น ภาษาไทย ภาษาอังกฤษ ตัวอย่างคือ การแปลงคำพูดเป็นข้อความและการแปลงข้อความเป็นคำพูด
- มีประสาทรับสัมผัสคล้ายมนุษย์ เช่น คอมพิวเตอร์รับภาพได้โดยอุปกรณ์รับสัมผัสแล้วนำภาพไปประมวลผล
- เคลื่อนไหวได้คล้ายมนุษย์ เช่น หุ่นยนต์ช่วยงานต่าง ๆ เคลื่อนย้ายสิ่งของ
- เรียนรู้ได้โดยสามารถตรวจจับรูปแบบการเกิดของเหตุการณ์ใด ๆ แล้วสามารถปรับตัวสู่สิ่งแวดล้อมที่เปลี่ยนไปได้

3. ระบบที่คิดอย่างมีเหตุผล (Systems that think rationally)

a. [AI คือ] การศึกษาความสามารถในด้านสติปัญญา โดยการใช้โมเดลของการคำนวณ ("The study of mental faculties through the use of computational model." [Charniak and McDermott, 1985])

b. [AI คือ] การศึกษาวิธีการคำนวณที่สามารถรับรู้ ใช้เหตุผล และกระทำ ("The study of the computations that make it possible to perceive, reason, and act" [Winston, 1992])

หมายเหตุ คิดอย่างมีเหตุผล หรือคิดถูกต้อง เช่น ใช้หลักตรรกศาสตร์ในการคิดหาคำตอบอย่างมีเหตุผล เช่น ระบบผู้เชี่ยวชาญ

4. ระบบที่กระทำอย่างมีเหตุผล (Systems that act rationally)

a. ปัญญาประดิษฐ์คือการศึกษาเพื่อออกแบบ Agent ที่มีปัญญา ("Computational Intelligence is the study of the design of intelligent agents" [Poole et al., 1998])

b. AI เกี่ยวข้องกับพฤติกรรมที่แสดงปัญญาในสิ่งที่มนุษย์สร้างขึ้น ("AI ... is concerned with intelligent behavior in artifacts" [Nilsson, 1998])

หมายเหตุ กระทำอย่างมีเหตุผล เช่น Agents (โปรแกรมที่มีความสามารถในการกระทำ หรือเป็นตัวแทนในระบบอัตโนมัติต่าง ๆ) สามารถกระทำอย่างมีเหตุผลเพื่อบรรลุเป้าหมายที่ได้ตั้งไว้ เช่น Agent ในระบบขับรถอัตโนมัติ ที่มีเป้าหมายว่าต้องไปถึงเป้าหมายในระยะทางที่สั้นที่สุด ต้องเลือกเส้นทางที่ไปยังเป้าหมายที่สั้นที่สุดที่เป็นไปได้ จึงจะเรียกได้ว่า Agent กระทำอย่างมีเหตุผล อีกตัวอย่างเช่น Agent ในเกมหมากรุก ที่มีเป้าหมายว่าต้องเอาชนะคู่ต่อสู้ ก็ต้องเลือกเดินหมากที่จะทำให้คู่ต่อสู้แพ้ให้ได้ เป็นต้น

2.1.2 Specifying the task environment [4]

การออกแบบความสามารถของตัวละครในเกมวิ่งแข่งบนมือถือ โดยใช้หลักการ Intelligent Agents ในการออกแบบขั้นแรก ต้องกำหนด PEAS ซึ่งประกอบไปด้วย Performance, Environment, Actuators, Sensors โดยที่

- Performance หมายถึงประสิทธิภาพของ Agents ที่สามารถทำอะไรได้บ้าง เหมือนเป็นจุดมุ่งหมายของ Agents ซึ่งถ้า Agent ทำได้ตามจุดมุ่งหมายแสดงว่าประสบความสำเร็จ
- Environment หมายถึง สภาพแวดล้อมภายในเกม ที่ส่งผลกระทบต่อการตัดสินใจของ Agents ว่าต้องทำอะไรเพื่อให้บรรลุเป้าหมาย
- Actuators หมายถึง อุปกรณ์ที่ตอบสนองต่อสภาพแวดล้อมในลักษณะการกระทำ (Actions) ในที่นี้หมายถึงการกระทำของ Agents ที่แสดงออกมา

- Sensors หมายถึง อุปกรณ์ที่เอาไว้รับรู้สภาพแวดล้อมเพื่อนำข้อมูลไปประมวลผล

2.1.3 Properties of task environments [4]

เป็นการระบุคุณลักษณะของสภาพสิ่งแวดล้อมที่ตัว Agents ให้ได้รับรู้เพื่อนำไปประมวลผลสามารถจำแนกคุณลักษณะได้ดังนี้

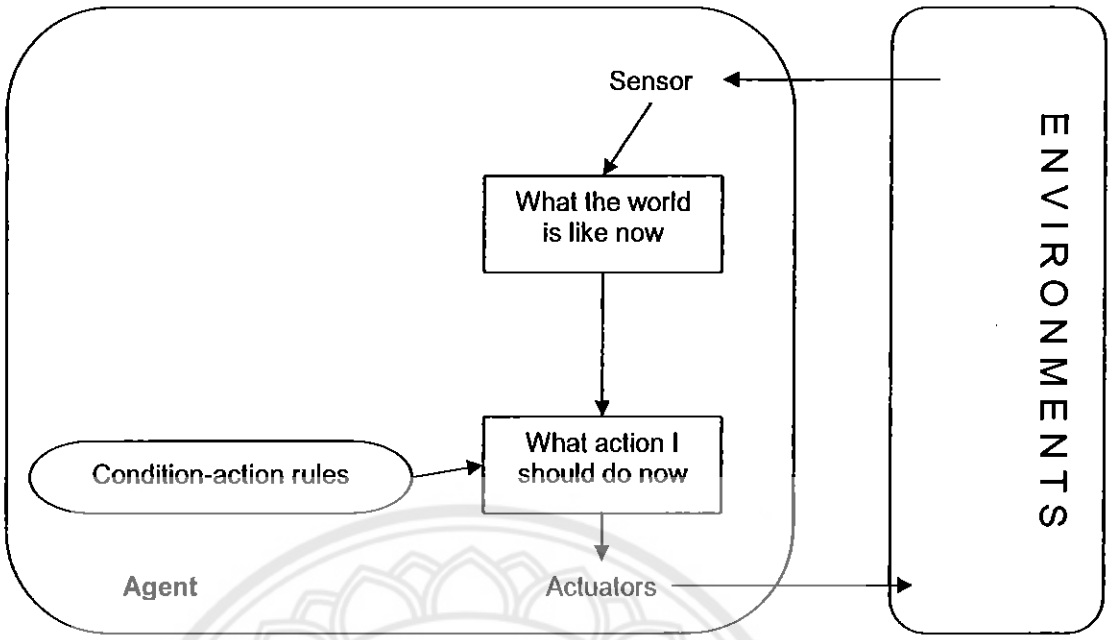
- Fully observable vs. Partially observable หมายความถึงสิ่งแวดล้อมที่ Agents ได้เจอ ตัว Agents สามารถรู้ทั้งหมดเลยหรือไม่ว่าสภาพแวดล้อมเป็นอย่างไร
- Deterministic vs. Stochastic หมายความถึงสิ่งแวดล้อมที่ Agents ได้เจอ Agents มีความสามารถที่จะคาดการณ์ได้ล่วงหน้าหรือไม่
- Episodic vs. Sequential หมายความถึงสิ่งแวดล้อมที่ Agents ได้เจอ เป็นแบบช่วง หรือแบบต่อเนื่องกัน
- Static vs. Dynamic หมายความถึงสิ่งแวดล้อมที่ Agents ได้เจอนั้นมีความสามารถเปลี่ยนแปลงไปเรื่อย ๆ หรือว่าเป็นแบบเดิมตลอด
- Discrete vs. Continuous หมายความถึงสิ่งแวดล้อมที่ Agents ได้เจอ สามารถหยุดเพื่อคำนวนได้ หรือว่าสภาพแวดล้อมดำเนินหรือเปลี่ยนแปลงไปเรื่อย ๆ ในขณะที่ Agents ยังคำนวณอยู่
- Single agent vs. Multiagent หมายความว่าสิ่งแวดล้อมที่ Agents ได้เจอนั้น มีเพียงอย่างเดียวที่ส่งผลกระทบต่อการคำนวนของ Agents หรือไม่

2.1.4 Agent Program [4]

มี 4 แบบดังนี้

1. Simple reflex agents เลือกทำตามสิ่งที่รับรู้ไม่สนใจอดีต
2. Model-based reflex agents นำค่าในอดีตมามีส่วนร่วมในการตัดสินใจ
3. Goal-based agents เน้นการทำตามเป้าหมายเป็นหลัก
4. Utility-based agents นำทุกสิ่งมารวมกันเพื่อใช้ในการตัดสินใจทำต่อไป

ในการออกแบบปัญญาประดิษฐ์ของเกมเลือก Simple reflex agents มาใช้ ซึ่ง Agents มีการเลือกการกระทำตามสิ่งที่รับรู้ได้ทั่วไปโดยไม่สนใจสิ่งที่ผ่านมาในอดีต วิธีคิดของ Agents สามารถอธิบายได้เป็นแผนผัง โดยที่รูปก้อนสี่เหลี่ยมด้านในแสดงถึงสิ่งที่อยู่ภายในของกระบวนการการตัดสินใจของ Agents และรูปวงรีแสดงถึงข้อมูลที่ใช้ในกระบวนการของ Agent Program

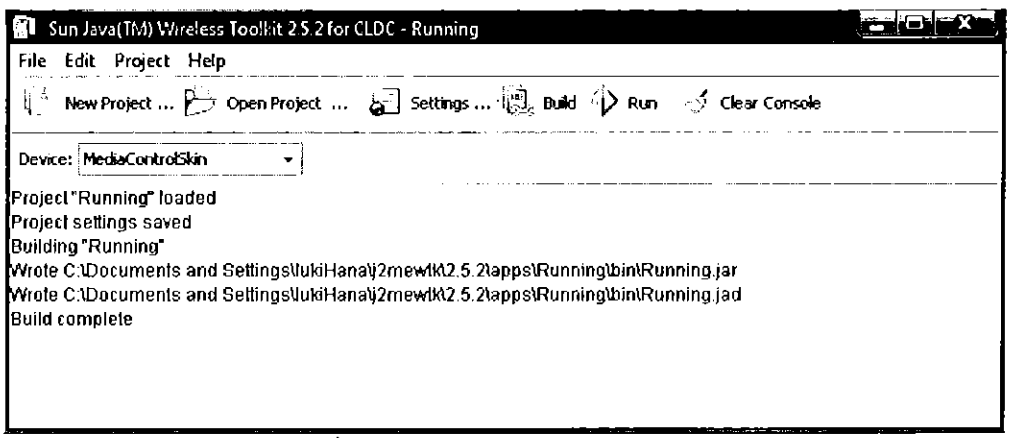


รูปที่ 2.1 Simple reflex agents [4]

2.2 J2ME Wireless Toolkit [1, 2]

ในการพัฒนาแอพพลิเคชันใด ๆ เราจำเป็นต้องใช้เครื่องมือช่วยในการพัฒนาอยู่เสมอ เครื่องมือในการพัฒนา MIDlet ก็เช่นกัน J2ME Wireless Toolkit ถือว่าเป็นเครื่องมือชนิดแรก ๆ ที่เปิดให้ดาวน์โหลดได้ฟรี ซึ่งโปรแกรมนี้มีหน้าที่คอมไพล์ (Compile) และสร้างแพคเกจ (Package) เพื่อไปติดตั้งบนโทรศัพท์มือถือ

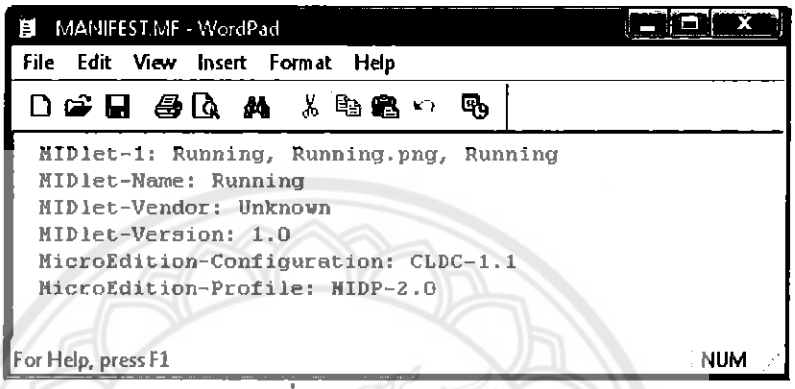
เมื่อทำการติดตั้งโปรแกรมเรียบร้อยแล้ว ทุกครั้งที่สร้าง MIDlet ไฟล์ทั้งหมดต้องถูกเก็บไว้ใน C:\...\j2mewtk\2.5.2\apps\ (ProjectName) เมื่อคอมไพล์ (Compile) เรียบร้อยแล้วจะได้ไฟล์ *.jar และ *.jad โดยเก็บเอาไว้ในไดเร็คทอรี bin ของโปรเจ็กต์ดังรายละเอียดที่บอกไว้ในหน้าต่างหลักของ J2ME WTK หลังจากสร้างแพคเกจ (Package) เสร็จแล้ว



รูปที่ 2.2 โฟลเดอร์ที่เก็บ MIDlet Suite

2.2.1 ไฟล์ *.jar

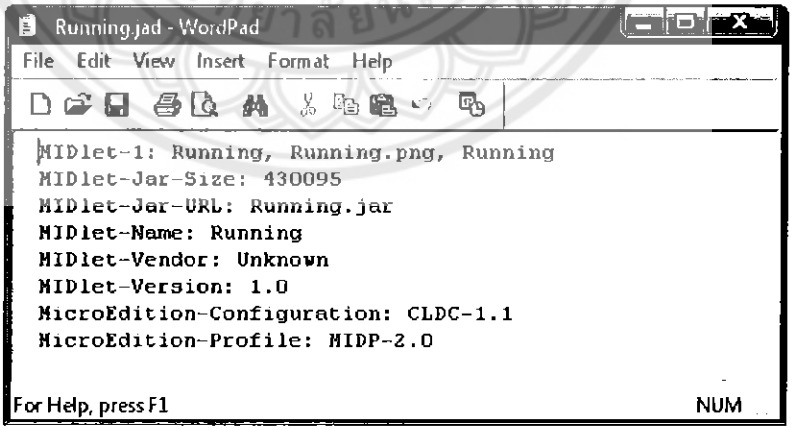
ไฟล์ *.jar ประกอบไปด้วยไฟล์ *.class ซึ่งคอมไพล์ (Compile) และผ่านการตรวจสอบแล้ว ไฟล์ Resource ต่าง ๆ เช่น รูปภาพและไฟล์เสียง (Audio) เป็นต้น รวมทั้งไฟล์ *.mf ซึ่งเป็นตัวบอกว่าในไฟล์ *.jar นั้นมีเนื้อหาหาอะไรบ้าง นอกจากนี้ยังประกอบไปด้วยข้อมูลอื่น ๆ เช่น ชื่อ เวอร์ชัน ผู้ผลิต (Vendor) ของ MIDlet suite อีกด้วย



รูปที่ 2.3 ตัวอย่างไฟล์ *.mf

2.2.2 ไฟล์ *.jad

ไฟล์ *.jad จะมีข้อมูลเกี่ยวกับตัว MIDlet รวมถึงชื่อและข้อมูลคุณสมบัติต่าง ๆ ซึ่งสามารถใช้ Notepad หรือ Text Editor ตัวใดเปิดดูก็ได้



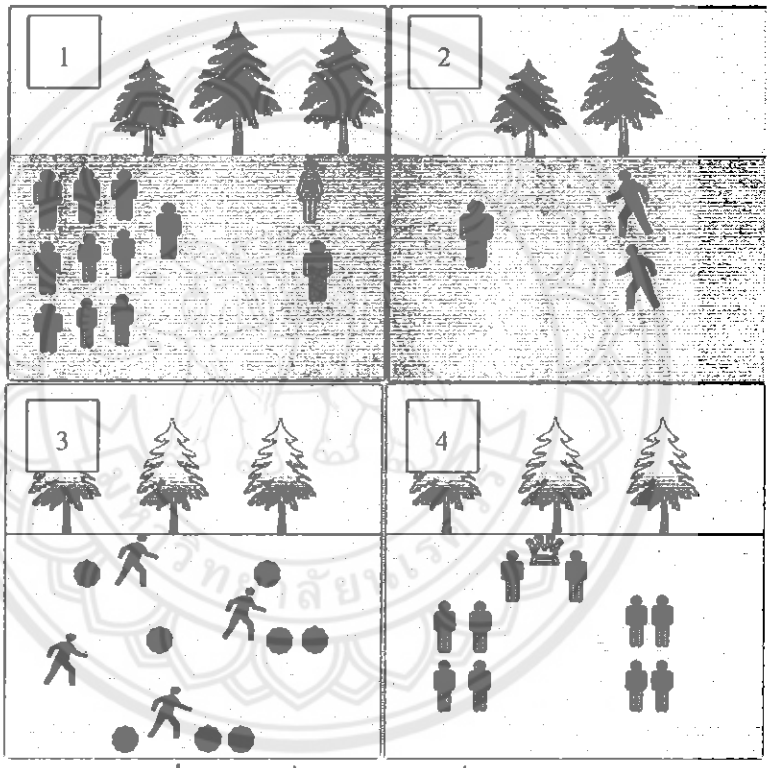
รูปที่ 2.4 ตัวอย่างไฟล์ *.jad

หลังจากสร้างแพคเกจ (Package) แล้วสามารถดาวน์โหลด *.jad และ *.jar ลงในอุปกรณ์ได้โดยอาจผ่านทางสายดาต้าลิงค์ซึ่งต่อเข้าโดยตรงกับคอมพิวเตอร์ ใช้อินฟราเรดผ่านพอร์ตอินฟราเรด หรือใช้ Wap ผ่านเครือข่ายของผู้ให้บริการ

บทที่ 3
วิธีการดำเนินงาน

ในบทนี้กล่าวถึงขั้นตอนในการดำเนินการออกแบบเกมวิ่งแข่งบนมือถือ โดยเริ่มจากการเขียนสตอรี่บอร์ด (Story board) ของเกม

3.1 สตอรี่บอร์ด (Story board)



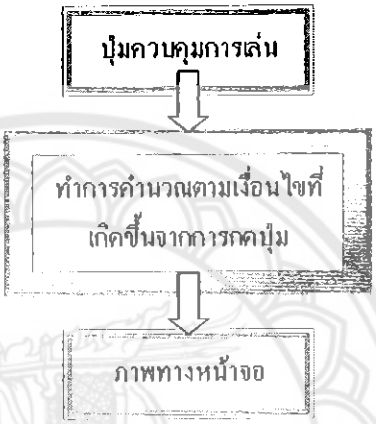
รูปที่ 3.1 สตอรี่บอร์ดของเกมวิ่งแข่งบนมือถือ

จากรูปสามารถอธิบายได้ดังนี้ เกมที่พัฒนาลักษณะของเกมเป็นเกมวิ่งแข่งที่มีเรื่องราว โดยบอกเล่าเรื่องราวผ่านบทสนทนาต่าง ๆ ในระหว่างการแข่งขันเกิดการโจมตีคู่แข่งโดยใช้ขวานฟันและยังต้องหลบสิ่งกีดขวางในลู่วิ่งอีกด้วย การแข่งขันมีทั้งหมด 18 ครั้งแต่มีการพิจารณาทุก ๆ 6 ครั้งหากผู้เล่นอยู่ในลำดับที่ 3 และ 4 ถือว่า "ตกรอบ" (Game Over)

3.2 การออกแบบเกม

จากสตอรี่บอร์ดและขอบเขตของโครงงานสามารถวิเคราะห์โครงสร้างของเกมซึ่งสามารถแบ่งเป็นส่วนย่อย ๆ ดังนี้

- ส่วนความสามารถของตัวละคร
- ส่วนการคำนวณค่าพื้นฐานของตัวละครและคะแนนการแข่งขัน
- ส่วนของรูปภาพและการแสดงผล



รูปที่ 3.2 โครงสร้างของเกม

ซึ่งในส่วนของความสามารถของตัวละครนั้นถือเป็นหัวใจหลักของการพัฒนาเนื่องจากส่วนนี้เป็นส่วนที่แสดงศักยภาพในการเพิ่มทักษะของตัวละครนั้น ๆ

3.2.1 ความสามารถของตัวละคร

ใช้หลักการจากปัญญาประดิษฐ์ (AI) สร้างวิธีคิดให้แก่ตัวละครในเกม โดยที่ตัวละครสามารถพัฒนาขีดความสามารถได้ตามสถานการณ์ที่ตัวละครได้พบเจอตลอดระยะเวลาการเล่นเกมในแต่ละครั้ง ผู้เล่นสามารถเล่นกับตัวละครที่มีความสามารถแตกต่างกันทุกครั้งที่มีการเล่นใหม่

จากหลักการ Intelligent Agents สามารถกำหนด PEAS ของ Agent จากสตอรี่บอร์ดและขอบเขตของโครงงานได้ดังตารางที่ 3.1 และวิเคราะห์ Task environment ได้ดังตารางที่ 3.2

ตารางที่ 3.1 รายละเอียด PEAS

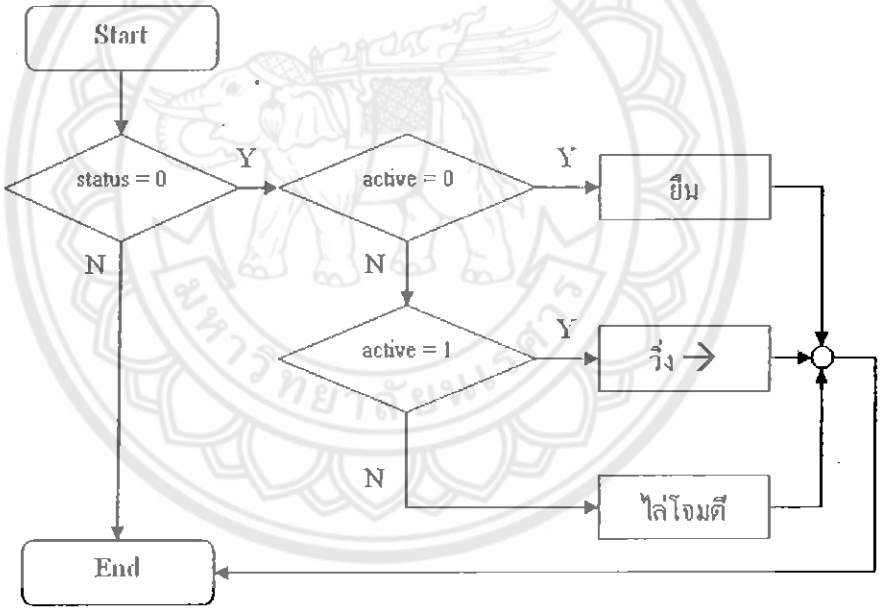
Agent Type	Performance Measure	Environment	Actuators	Sensors
ตัวละคร	เข้าเส้นชัยเป็นที่ 1, เอาชีวิตรอดได้, ทำลายคู่ต่อสู้ได้	สนามแข่ง, คู่แข่ง, สิ่งกีดขวาง	ขา, อาวุธ	ระยะการมองเห็น

ตารางที่ 3.2 Task environment

Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Running Bot	Partially	Stochastic	Sequential	Dynamic	Continuous	Multi

จากการวิเคราะห์ PEAS และ Task environment พบว่า AI ของเกมเหมาะสมกับ Simple reflex agents เนื่องจาก Agents program แบบนี้มีการตรวจสอบสภาพแวดล้อมตลอดเวลาเพื่อเลือกการกระทำให้เหมาะกับสถานการณ์ในเวลานั้น ๆ ตาม Condition-action rules ที่กำหนดไว้ซึ่งเงื่อนไขที่กำหนดไว้มีดังนี้

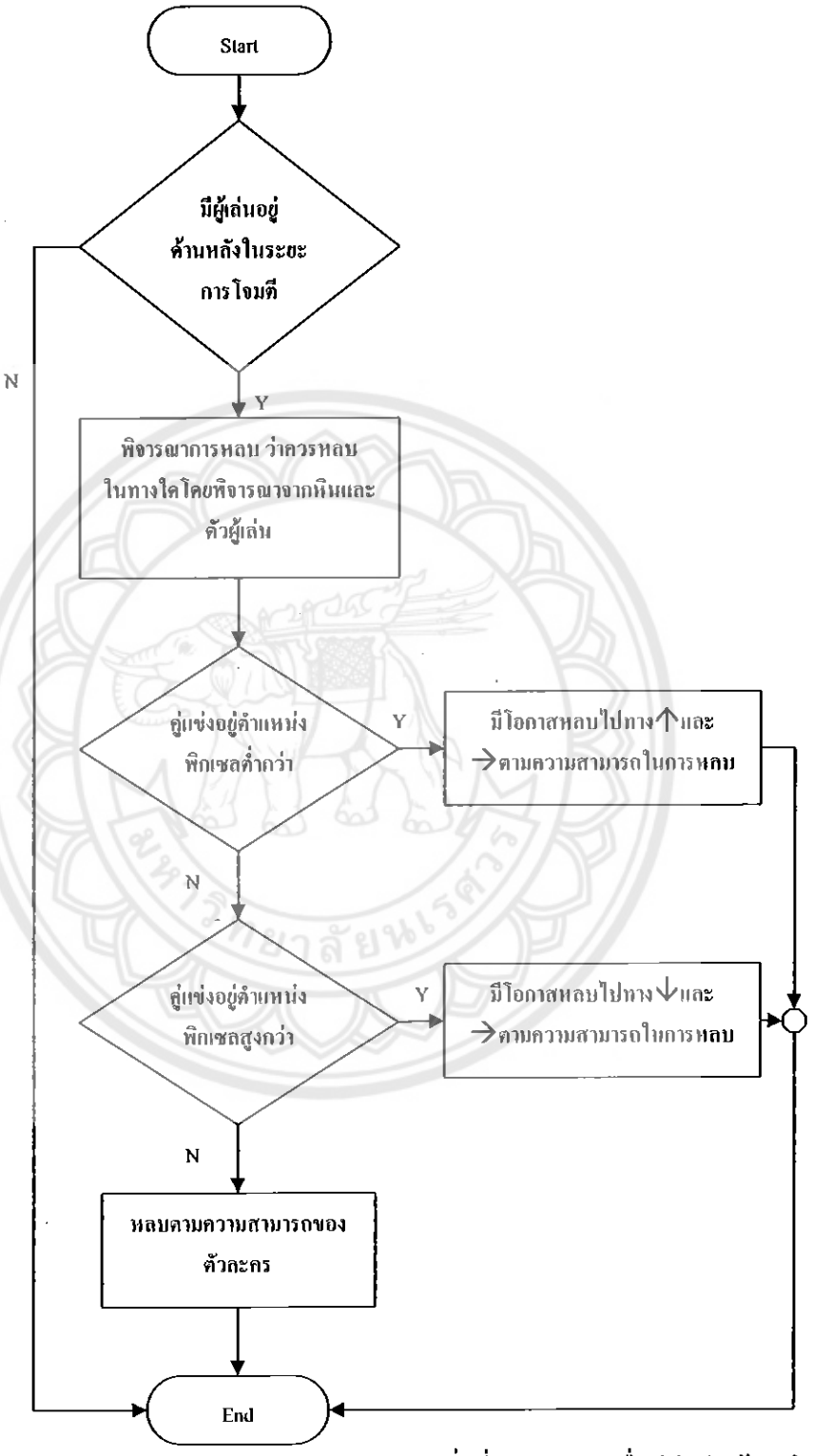
ส่วนนี้แสดงถึงความคิดของตัวละครเมื่อตัวละครวิ่งปกติ (วิ่งปกติคือการวิ่งที่ไม่มีการกระโดดและโจมตีคู่ต่อสู้)



รูปที่ 3.3 แผนผังการทำงาน (Flow chart) แสดงสิ่งที่ตัวละครคิดเมื่อวิ่งปกติ

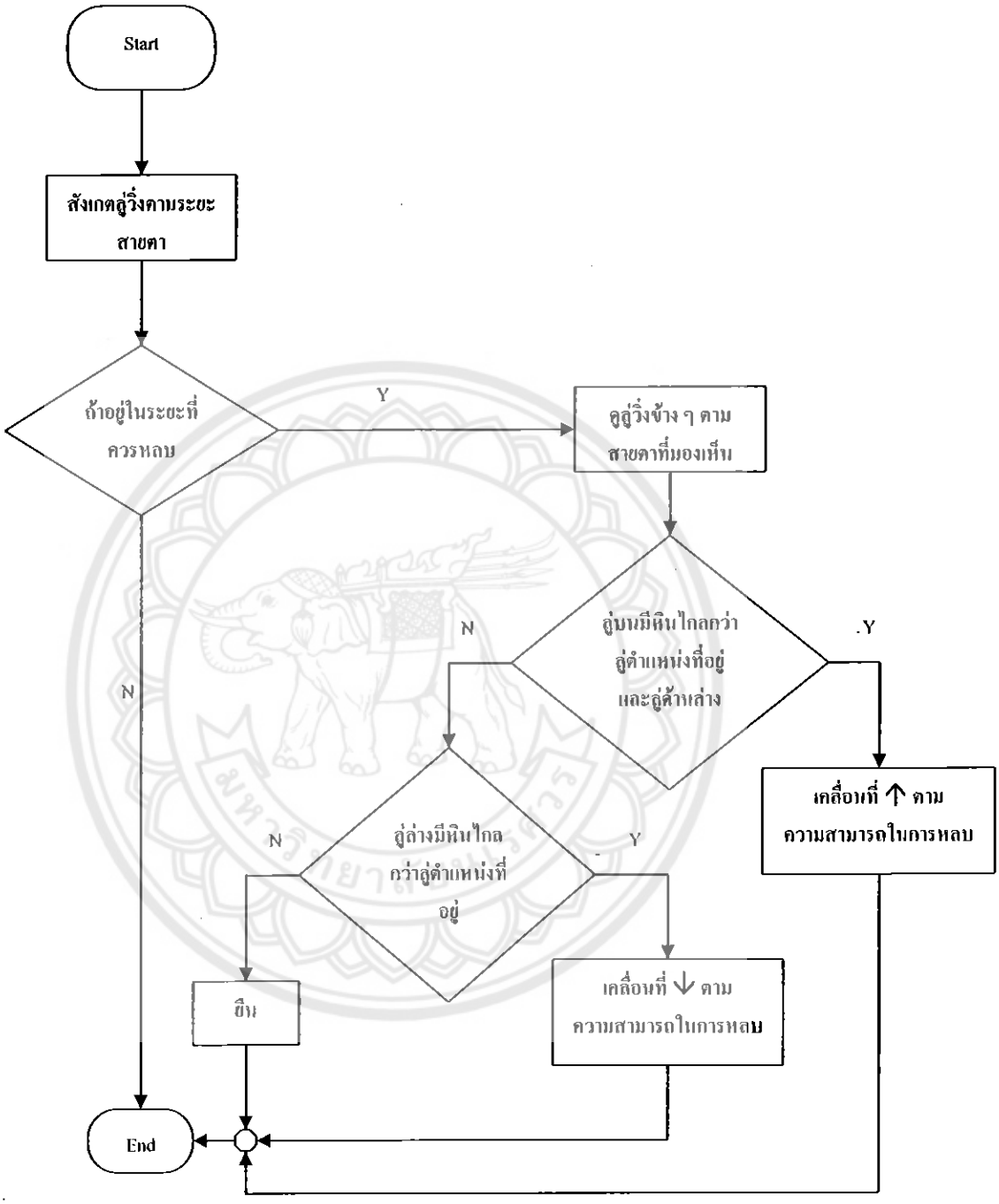
เนื่องจากสิ่งที่ตัวละครต้องทำคือเอาชีวิตรอดจากการถูกโจมตี โจมตีผู้อื่นและเข้าเส้นชัยอันดับ 1 ให้ได้ ทำให้ในช่วงที่ตัวละครวิ่งปกตินั้นต้องมีการตัดสินใจเพื่อไล่โจมตีผู้อื่นหรือวิ่งขึ้นหน้าเพื่อเข้าเส้นชัยอยู่ด้วย

วิธีตัดสินใจของตัวละครในเกมเมื่อมีคู่แข่งเข้ามาด้านหลังเพื่อโจมตี



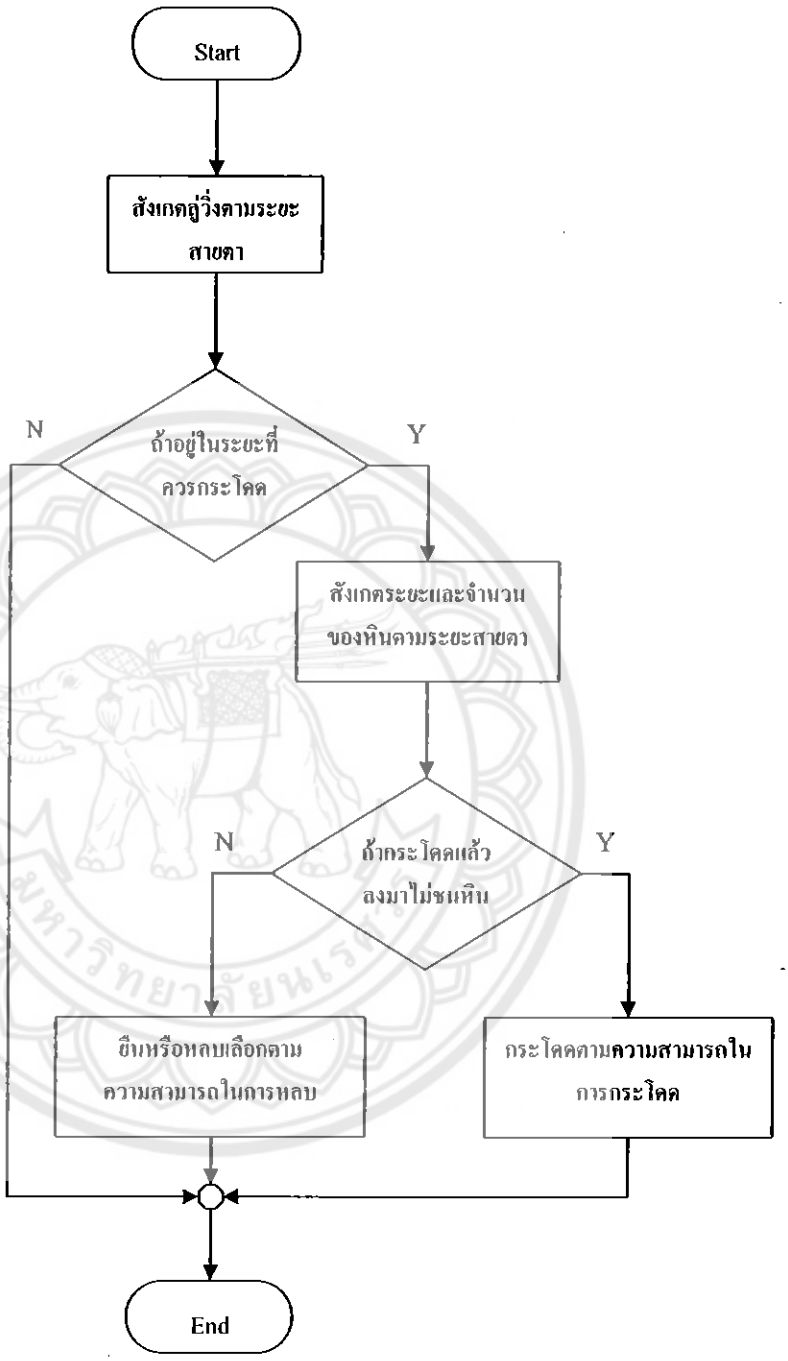
รูปที่ 3.4 แผนผังการทำงาน (Flow chart) แสดงสิ่งที่ตัวละครทำเมื่อมีคู่แข่งเข้ามาโจมตี

วิธีหลบสิ่งกีดขวาง เมื่อตัดสินใจได้เคลื่อนที่หลบแล้ว



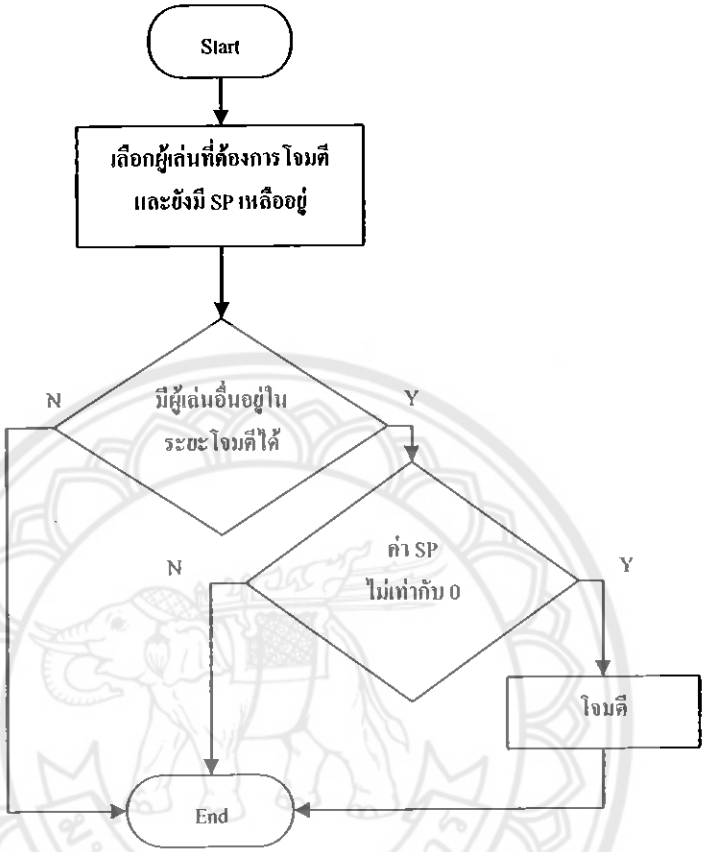
รูปที่ 3.5 แผนผังการทำงาน (Flow chart) แสดงการหลบสิ่งกีดขวาง

วิธีกระโดดผ่านสิ่งกีดขวาง เมื่อตัดสินใจได้หลบด้วยการกระโดดแล้ว



รูปที่ 3.6 แผนผังการทำงาน (Flow chart) แสดงการกระโดดผ่านสิ่งกีดขวาง

จากรูปที่ 3.3 เมื่อเลือกได้ไล่โจมตีตัวละครและเมื่อระยะของคู่แข่งอยู่ด้านหน้าในระยะที่ตัวละครสามารถฟันคู่แข่งได้แล้ว ทำตามแผนผังการทำงาน (Flow chart) ดังนี้



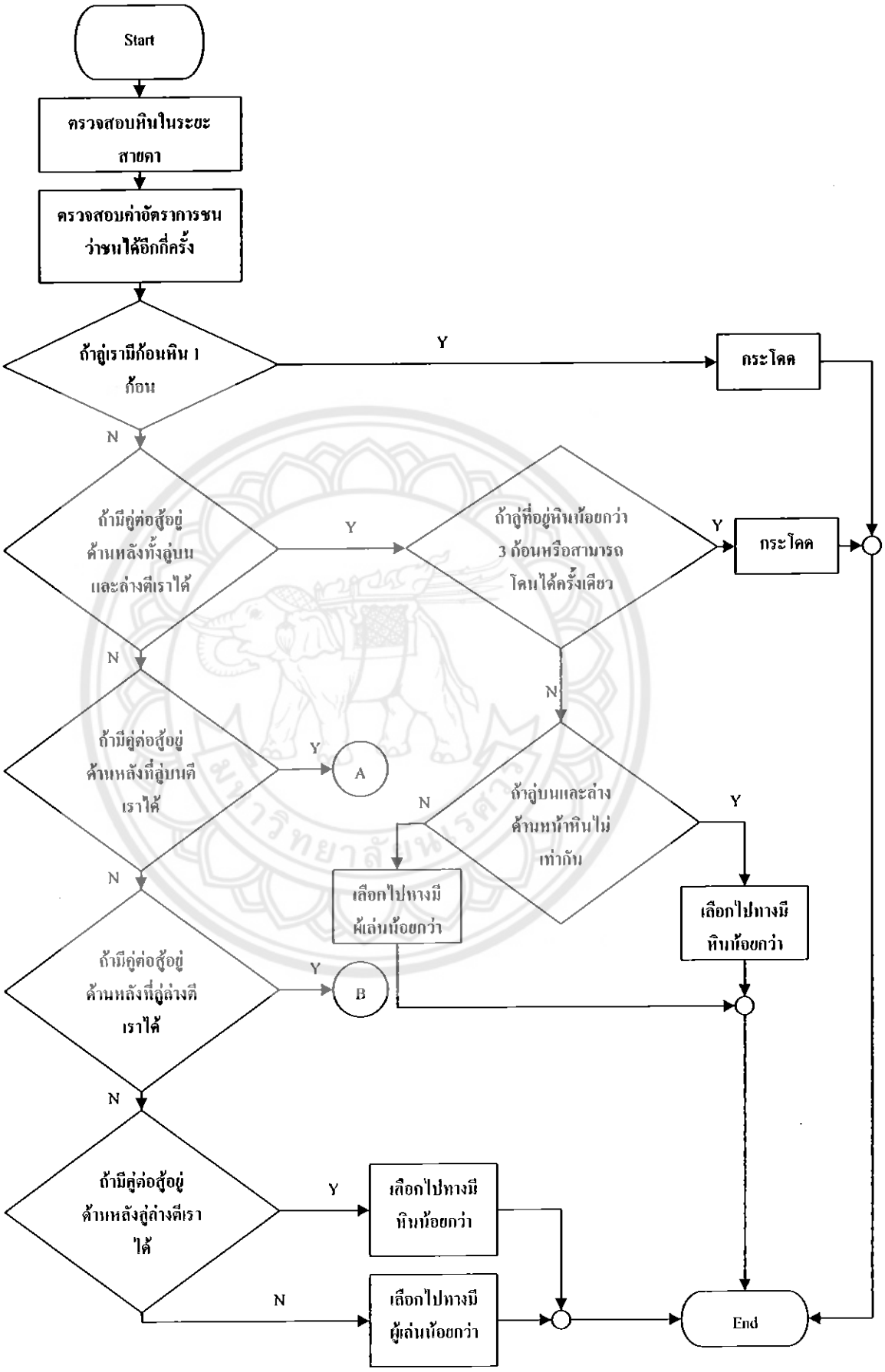
รูปที่ 3.7 แผนผังการทำงาน (Flow chart) แสดงการไล่โจมตีและฟันคู่แข่ง

การไล่โจมตีคู่แข่ง ตัวละครเลือกจากคู่แข่งที่อยู่ใกล้และอยู่ด้านหน้าของตำแหน่งที่ตัวเองอยู่

การเพิ่มทักษะของตัวละครแทรกอยู่ในทุกการกระทำของตัวละคร ดังนี้

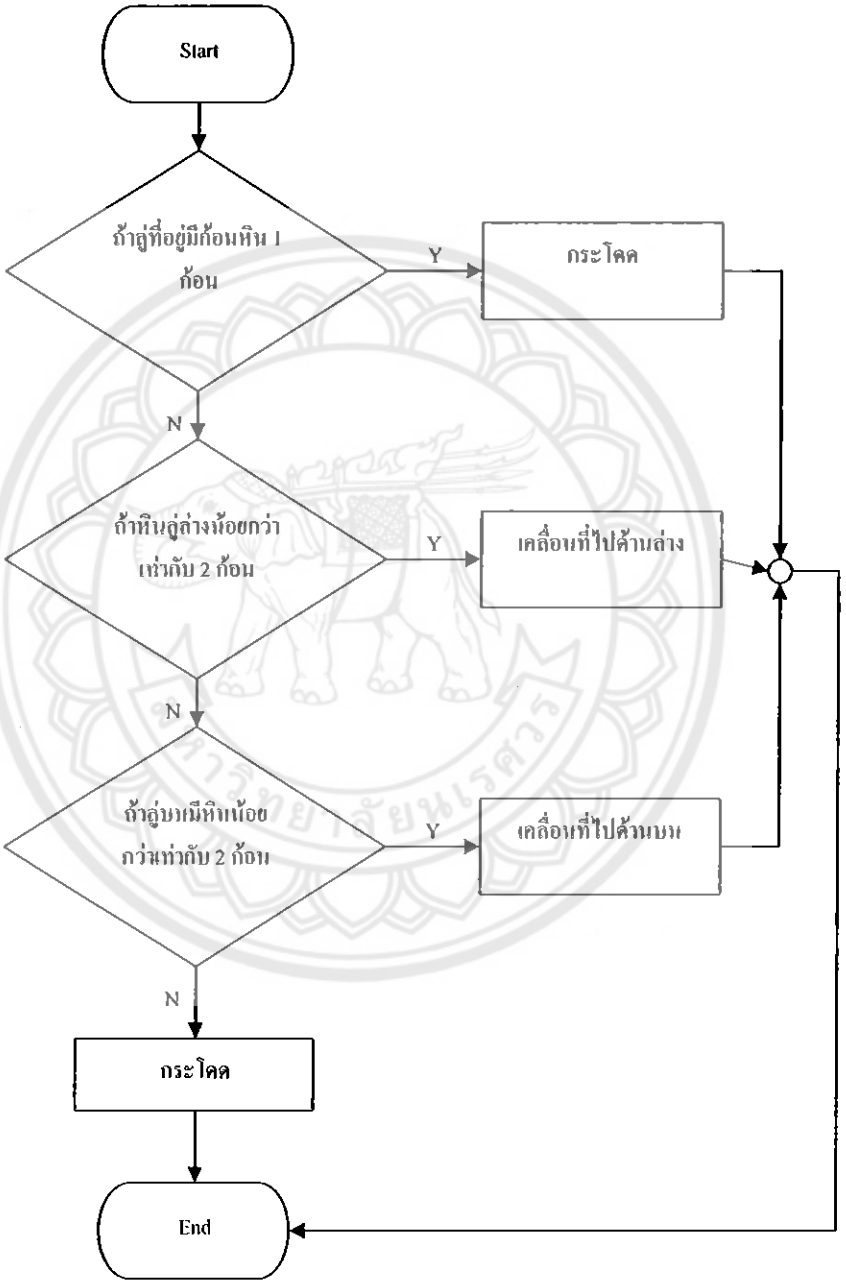
- ตัวละครตัดสินใจหลบสิ่งกีดขวางด้วยการกระโดด ค่าความสามารถในการตัดสินใจกระโดดจะเพิ่มขึ้น
- ตัวละครตัดสินใจหลบสิ่งกีดขวางด้วยการเคลื่อนที่หลบ ค่าความสามารถในการตัดสินใจการเคลื่อนที่หลบจะเพิ่มขึ้น
- ตัวละครตัดสินใจได้การโจมตี ค่าความสามารถในการตัดสินใจการโจมตีเพิ่มขึ้น

เพิ่มที่ค่าขีดความสามารถในการตัดสินใจของตัวละคร โดยค่าขีดความสามารถมีค่าทั้งหมด 1000 ในช่วงเริ่มต้นเกมตัวละครแต่ละตัวได้รับค่าขีดความสามารถในการตัดสินใจที่ต่ำ ๆ อยู่ในอัตราส่วน 1/20 ถึง 1.5/20 จากนั้นหากมีการกระทำที่ทำให้เกิดการเพิ่มขีดความในการตัดสินใจขึ้นมาแล้ว ความสามารถของตัวละครแต่ละตัวจะแตกต่างกันออกไป



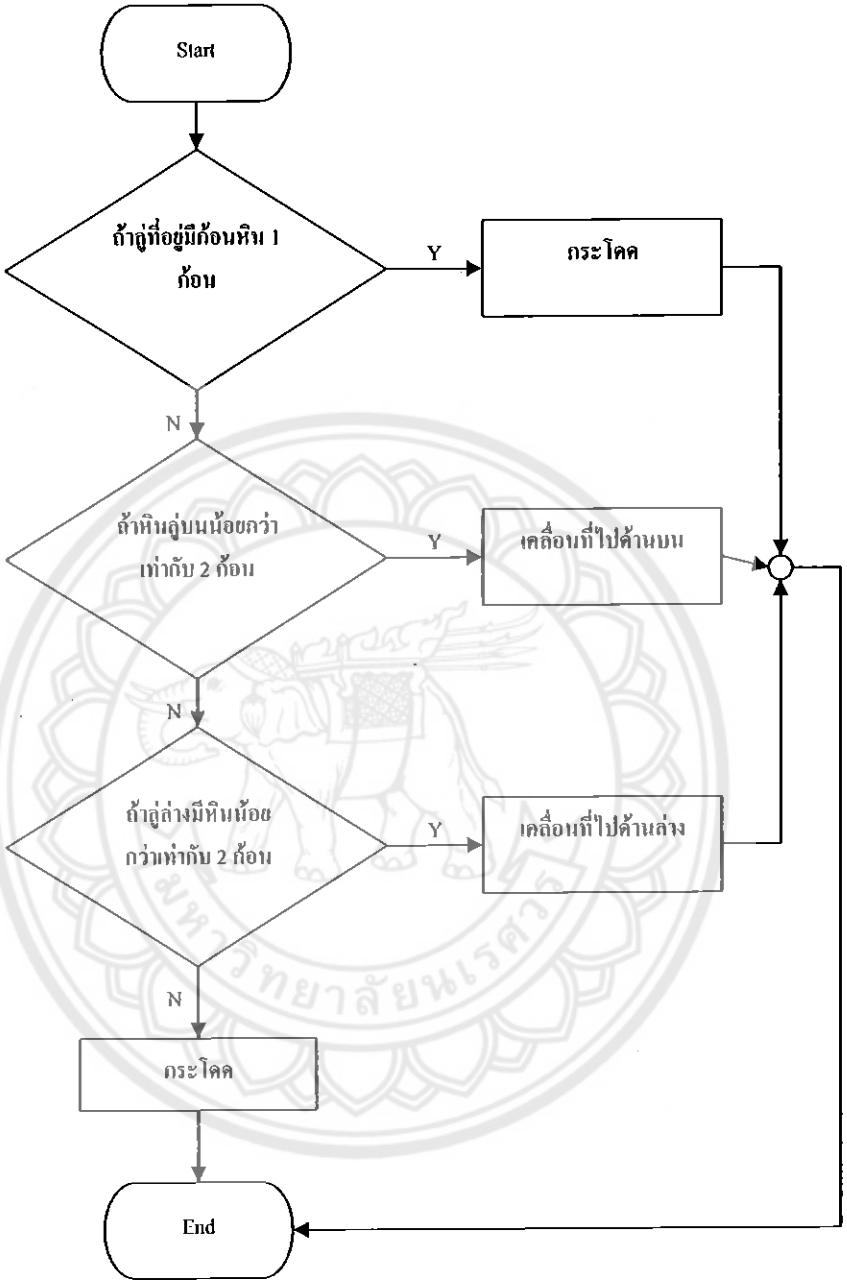
รูปที่ 3.8 แผนผังการทำงาน (Flow chart) แสดงการเลือกการหลบ

แผนผังความคิดนี้แสดงเมื่อมีคู่แข่งเข้ามาทางด้านหลังของตัวละครและทางด้านหน้ามีสิ่งกีดขวางอยู่ ตัวละครเลือกตัดสินใจจากค่าของพลังชีวิต ซึ่งค่าพลังชีวิตลดลงจากการโจมตีจากคู่ต่อสู้อื่นมากกว่าการชนสิ่งกีดขวาง ทำให้เลือกทำการหลบคู่ต่อสู้อื่นก่อนคิดหลบสิ่งกีดขวาง ต่อจากรูปที่ 3.8 ในสัญลักษณ์วงกลม A ถ้ามีคู่ต่อสู้อยู่ด้านหลังลู่บนสามารถโจมตีได้ ตัวละครจะทำดังนี้



รูปที่ 3.9 แผนผังการทำงาน (Flow chart) เมื่อคู่ต่อสู้อยู่ด้านหลังลู่บนสามารถโจมตีได้

ในสัญลักษณ์วงกลม B ถ้ามีคู่ต่อสู้อยู่ด้านหลังคู่ล่างสามารถโจมตีได้ ตัวละครจะทำดังนี้



รูปที่ 3.10 แผนผังการทำงาน (Flow chart) เมื่อคู่ต่อสู้อยู่ด้านหลังคู่ล่างสามารถโจมตีได้

3.2.2 ข้อมูลพื้นฐานของตัวละครและคะแนนการแข่งขัน

ข้อมูลพื้นฐานของตัวละครได้แก่ HP และ SP ทั้งสองเป็นพลังชีวิตและค่าความสามารถแสดงท่าทางต่าง ๆ ในการเล่นเกม

1. ค่าพลัง HP (Health Points) ค่าพลัง HP ถูกกำหนดให้ค่าพลังเต็มทุกครั้งที่เข้าฉากใหม่ และลดลงเมื่อชนกับสิ่งกีดขวางหรือถูกคู่ต่อสู้โจมตี

2. ค่าพลัง SP (Stamina Points) ค่าพลัง SP ถูกกำหนดให้ค่าพลังเต็มทุกครั้งที่เข้าฉากใหม่ เช่นเดียวกับค่าพลัง HP ลดลงเมื่อมีการกระโดดหรือโจมตีคู่ต่อสู้ และเพิ่มขึ้นเมื่อกลับมาเคลื่อนที่ปกติ

คะแนนในการแข่งขันให้ลำดับคะแนนดังนี้

- อันดับ 1 ได้ 4 คะแนน
- อันดับ 2 ได้ 2 คะแนน
- อันดับ 3 ได้ 1 คะแนน
- อันดับ 4 ได้ 0 คะแนน

เมื่อทำการแข่งขันครบทุก 6 ฉาก รวมคะแนนเพื่อประมวลผลเกมให้เข้าสู่ฉากต่อไป หรือจบเกม หากลำดับการแข่งขันของผู้เล่น อยู่ในลำดับที่ 3 และ 4 ถือว่าเกมสิ้นสุดลง

3.2.3 รูปภาพและการแสดงผล

จากสตอรี่บอร์ดรูปภาพที่นำมาใช้ประกอบภายในเกมประกอบไปด้วย รูปหน้าตัวละคร รูปพื้น (ลู่วิ่ง) รูปวิวท้องฟ้า รูปวิวต้นไม้ รูปตัวละคร รูปสิ่งกีดขวาง รูปอาวุธ ดังนี้

รูปที่ 3.11 หน้าตัวละคร

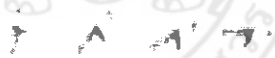
รูปที่ 3.12 พื้น (ลู่วิ่ง)

รูปที่ 3.13 วิวท้องฟ้า

รูปที่ 3.14 วิวต้นไม้

รูปที่ 3.15 ตัวละคร

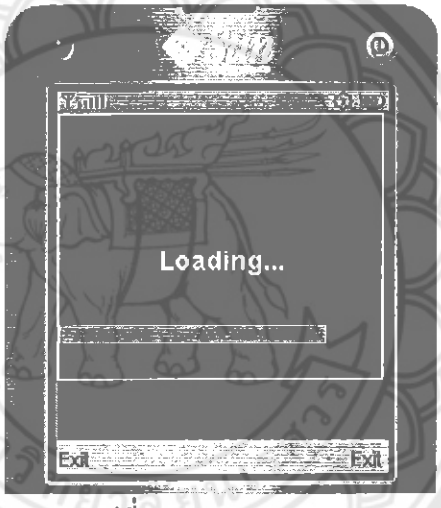
รูปที่ 3.16 สิ่งกีดขวาง



รูปที่ 3.17 อาวุธ

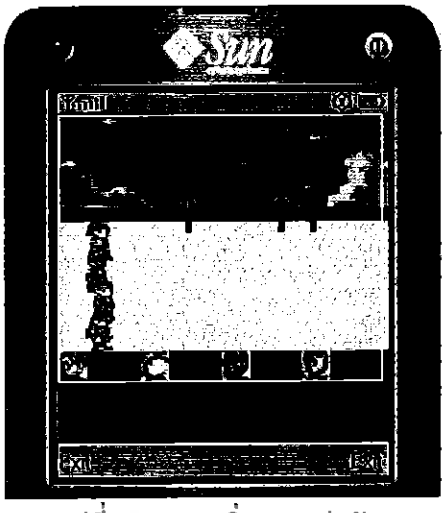
จากรูปที่เตรียมไว้และสตอรี่บอร์ดทำให้ได้อินเตอร์เฟส (Interface) ของเกมทำให้ได้รูปแบบของฉากต่าง ๆ ได้แก่ ฉากเริ่มต้นเกม ฉากสนทนา ฉากการแข่งขัน และฉากแสดงคะแนน ดังตัวอย่างในรูป

รูปที่ 3.18 ฉากเริ่มเกม

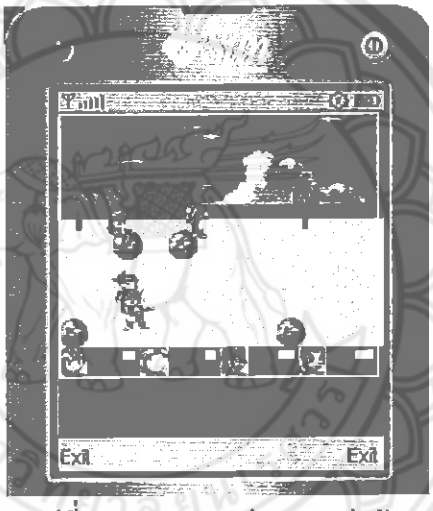


รูปที่ 3.19 ฉาก Loading

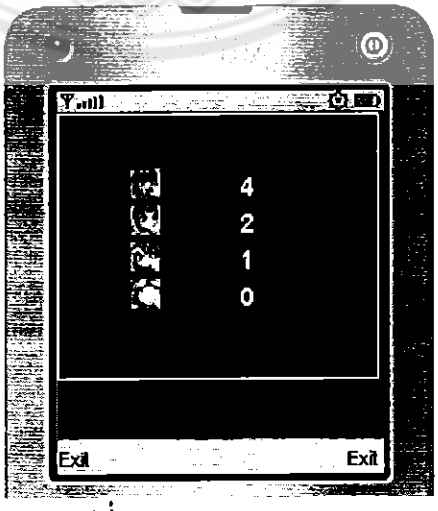
รูปที่ 3.20 ฉากสนทนา



รูปที่ 3.21 ฉากเริ่มการแข่งขัน



รูปที่ 3.22 ฉากระหว่างการแข่งขัน



รูปที่ 3.23 ฉากแสดงคะแนน

บทที่ 4

ผลการทดลอง

ในบทนี้กล่าวถึงวิธีเล่นเกม การทดสอบเกม และการวิเคราะห์ผล การทดสอบเกมสามารถแบ่งเป็นส่วนการแสดงภาพออกทางหน้าจอ และความสามารถของตัวละครในเกม ซึ่งการวิเคราะห์ผลนำเอาความสามารถของตัวละครในเกมมาวิเคราะห์

4.1 วิธีการเล่นเกม

เมื่อติดตั้งเกมเรียบร้อยแล้ว กดเริ่มโปรแกรม

- เคลื่อนที่ ↑ กดปุ่ม 2 หรือ soft key ไปในทิศ↑
- เคลื่อนที่ ← กดปุ่ม 4 หรือ soft key ไปในทิศ←
- เคลื่อนที่ → กดปุ่ม 6 หรือ soft key ไปในทิศ→
- เคลื่อนที่ ↓ กดปุ่ม 8 หรือ soft key ไปในทิศ↓
- กระโดดกดปุ่ม 3
- โจมตีคู่แข่งกดปุ่ม 9
- หยุดชั่วคราวกดปุ่ม 0

รูปที่ 4.1 แสดงปุ่ม Soft Key และ Key number

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ใช้ 023312 ค.2

4.2 การทดสอบเกม

ความสามารถของตัวละครที่นำมาวัดคือการหลบหลีก ซึ่งจากการทดสอบเกม ได้ทำการศึกษาและเก็บข้อมูลจากการเล่น 7 ครั้ง โดยเล่นทั้ง 18 ฉาก ได้ผลตามตารางดังนี้

ชร. ฉ 34 20 2550

ตารางที่ 4.1 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 1

ฉาก	จำนวนก้อนหิน	ไม่หลบเลย	หลบได้		หลบไม่ได้	
			กระโดด	เคลื่อนที่	กระโดด	เคลื่อนที่
1	18	15	1	1	0	1
2	24	17	2	3	1	1
3	18	12	2	2	0	2
4	22	14	3	3	0	2
5	22	8	1	5	2	6
6	25	8	4	7	1	5
7	26	8	7	6	1	4
8	23	7	2	11	0	3
9	25	5	2	15	0	3
10	30	4	8	11	1	6
11	27	3	8	11	2	3
12	20	3	5	9	1	2
13	32	4	6	15	2	5
14	28	3	7	12	2	4
15	32	3	13	12	2	2
16	37	4	10	13	4	6
17	36	2	14	14	2	4
18	37	3	11	16	3	4

ตารางที่ 4.2 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 2

ฉาก	จำนวนก้อนหิน	ไม่หลบเลย	หลบได้		หลบไม่ได้	
			กระโดด	เคลื่อนที่	กระโดด	เคลื่อนที่
1	23	22	1	0	0	0
2	22	16	4	2	0	0
3	21	12	3	2	3	1
4	27	11	5	5	1	5
5	20	7	1	8	0	4
6	25	5	6	6	2	6
7	17	5	1	5	1	5
8	20	5	5	4	1	5
9	24	7	2	12	1	2
10	25	6	5	9	2	3
11	26	4	7	11	0	4
12	23	4	2	13	0	4
13	42	5	8	20	2	7
14	38	4	11	13	2	8
15	31	4	6	15	0	6
16	34	4	6	13	2	9
17	30	3	14	6	2	5
18	43	1	16	19	5	2

ตารางที่ 4.3 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 3

ฉาก	จำนวนก้อนหิน	ไม่หลบเลย	หลบได้		หลบไม่ได้	
			กระโดด	เคลื่อนที่	กระโดด	เคลื่อนที่
1	24	22	1	1	0	0
2	19	16	2	1	0	0
3	22	13	4	2	0	3
4	22	11	1	7	0	3
5	16	6	2	3	2	3
6	28	6	7	7	2	6

ตารางที่ 4.3 (ต่อ) ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 3

ฉาก	จำนวนก้อนหิน	ไม่หลบเลย	หลบได้		หลบไม่ได้	
			กระโดด	เคลื่อนที่	กระโดด	เคลื่อนที่
7	24	5	3	7	5	4
8	22	3	7	9	0	3
9	25	6	6	9	0	4
10	22	4	2	10	3	3
11	38	6	9	12	1	10
12	25	4	6	10	1	4
13	26	2	5	10	3	6
14	18	1	3	7	0	7
15	18	1	6	4	1	6
16	28	1	5	10	5	7
17	39	3	10	15	0	11
18	45	3	11	19	1	11

ตารางที่ 4.4 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 4

ฉาก	จำนวนก้อนหิน	ไม่หลบเลย	หลบได้		หลบไม่ได้	
			กระโดด	เคลื่อนที่	กระโดด	เคลื่อนที่
1	20	18	0	1	0	1
2	21	16	2	2	1	0
3	23	14	4	3	1	1
4	21	11	1	5	2	2
5	21	10	2	5	1	3
6	16	5	2	6	0	3
7	21	8	3	5	2	3
8	23	6	6	7	0	4
9	24	7	4	10	1	2
10	29	5	6	13	0	5
11	29	6	7	10	3	3
12	29	5	5	13	1	5

ตารางที่ 4.4 (ต่อ) ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 4

ฉาก	จำนวนก้อนหิน	ไม่หลบเลย	หลบได้		หลบไม่ได้	
			กระโดด	เคลื่อนที่	กระโดด	เคลื่อนที่
13	32	4	6	15	1	6
14	36	3	13	12	1	7
15	32	3	9	14	1	5
16	34	1	8	16	2	7
17	33	2	13	10	3	5
18	39	1	15	16	4	3

ตารางที่ 4.5 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 5

ฉาก	จำนวนก้อนหิน	ไม่หลบเลย	หลบได้		หลบไม่ได้	
			กระโดด	เคลื่อนที่	กระโดด	เคลื่อนที่
1	19	18	0	0	0	1
2	23	15	3	1	1	3
3	19	10	3	2	1	3
4	21	9	3	6	0	3
5	23	13	2	5	1	2
6	24	6	6	6	2	4
7	17	5	2	6	1	3
8	24	7	6	7	2	2
9	33	8	4	10	6	5
10	31	5	4	9	5	8
11	28	4	8	11	1	4
12	28	6	4	11	2	5
13	35	4	7	15	3	6
14	30	2	3	14	3	8
15	31	3	6	13	2	7
16	36	1	7	17	2	9
17	35	2	10	12	3	8
18	40	1	10	15	4	10

ตารางที่ 4.6 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 6

ฉาก	จำนวนก้อนหิน	ไม่หลบเลย	หลบได้		หลบไม่ได้	
			กระโดด	เคลื่อนที่	กระโดด	เคลื่อนที่
1	21	20	0	0	0	1
2	20	13	4	0	2	1
3	25	13	7	4	0	1
4	17	10	1	3	3	0
5	25	7	10	7	0	1
6	20	9	3	3	0	5
7	14	5	2	4	1	2
8	18	6	1	7	0	4
9	28	4	6	10	2	6
10	31	6	8	11	1	5
11	28	5	6	14	1	2
12	28	3	8	10	3	4
13	32	4	5	15	2	6
14	31	3	8	12	4	4
15	32	4	11	13	2	2
16	32	2	8	11	5	6
17	39	4	16	15	1	3
18	33	3	10	14	3	3

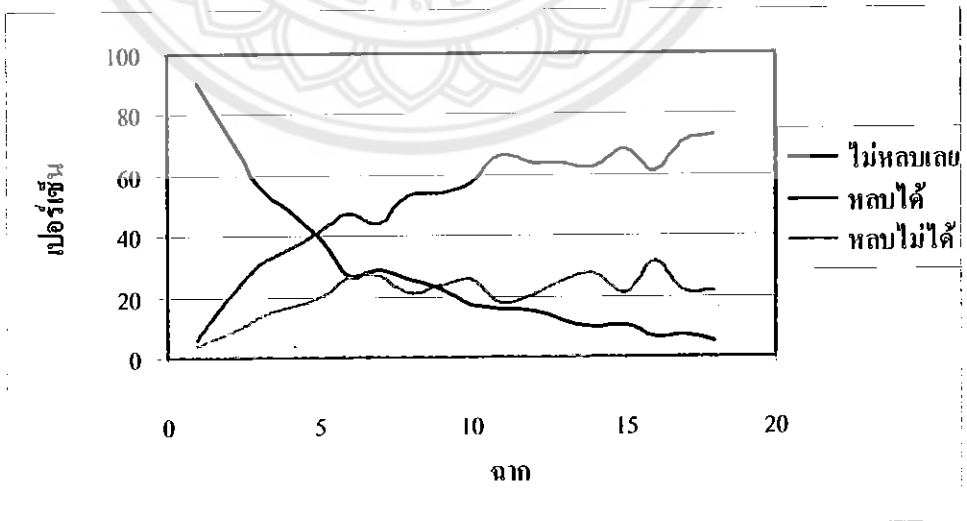
ตารางที่ 4.7 ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 7

ฉาก	จำนวนก้อนหิน	ไม่หลบเลย	หลบได้		หลบไม่ได้	
			กระโดด	เคลื่อนที่	กระโดด	เคลื่อนที่
1	18	14	1	1	1	1
2	14	11	1	1	1	0
3	22	11	5	2	1	3
4	23	8	7	4	0	4
5	26	9	9	3	1	4
6	23	4	8	5	2	4

ตารางที่ 4.7 (ต่อ) ตารางบันทึกผลความสามารถของตัวละคร 18 ฉาก ครั้งที่ 7

ฉาก	จำนวนก้อนหิน	ไม่หลบเลย	หลบได้		หลบไม่ได้	
			กระโดด	เคลื่อนที่	กระโดด	เคลื่อนที่
7	19	4	2	8	0	5
8	30	7	9	4	0	10
9	29	5	3	9	3	9
10	25	3	3	11	3	5
11	24	3	7	12	0	2
12	36	4	13	12	1	6
13	32	4	11	9	0	8
14	25	4	7	7	1	6
15	31	4	4	15	2	6
16	25	2	6	9	3	5
17	45	3	11	22	3	6
18	32	2	9	15	2	4

จากผลการทดลอง 7 ครั้ง สามารถสรุปออกมาเป็นกราฟแสดงค่าการหลบได้ หลบไม่ได้ และไม่หลบเลยเป็นเปอร์เซ็นต์ได้ดังนี้



รูปที่ 4.2 กราฟแสดงค่าการหลบได้ หลบไม่ได้ และไม่หลบเลยเป็นเปอร์เซ็นต์

จากกราฟแสดงให้เห็นถึงการพัฒนาทักษะของตัวละครซึ่งการพัฒนาแปรผันตรงตามฉาก

บทที่ 5
บทสรุป

โครงงานนี้สร้างเกมเพื่อจำลองระบบทางปัญญาประดิษฐ์ ด้วย J2ME ในรูปแบบของแนวเกมแอคชั่น ซึ่งเป็นการแข่งขันวิ่งวิบาก ต้องพบเจอกับอุปสรรคและการโจมตีจากคู่ต่อสู้ในเกม ตัวละครในเกมล้วนมีความสามารถแตกต่างกันออกไปขึ้นอยู่กับสถานการณ์ที่พบ

5.1 วิเคราะห์ผลการทดลอง

จากผลการทดลองในบทที่ 4 ข้อมูลการวัดความสามารถของตัวละครจากการกระโดดและเคลื่อนที่เพื่อหลบสิ่งกีดขวาง ตัวละครมีการพัฒนาอย่างไร ซึ่งการพัฒนาของตัวละครมี 3 ด้านคือ

- ด้านการหลบอุปสรรคต่าง ๆ
- ด้านการไล่โจมตี
- ด้านการวิ่งขึ้นหน้าเพื่อเข้าเส้นชัย

ทั้ง 3 ด้านนี้ตัวละครแต่ละตัวจะพัฒนาความสามารถตามเหตุการณ์ที่พบแตกต่างกันออกไป ทำให้การเล่นแต่ละครั้งเกิดความแปลกใหม่ สร้างความเพลิดเพลินแก่ผู้เล่นได้ ซึ่งการวัดการไล่โจมตีและการวิ่งขึ้นหน้านั้นเป็นไปได้ยาก ฉะนั้นจึงนำเอาข้อมูลส่วนการหลบอุปสรรคต่าง ๆ มาบันทึกและวิเคราะห์ผล

5.2 สรุปผลการทดลอง

จากผลการทดสอบเกมพบว่าตัวละครสามารถพัฒนาทักษะการหลบสิ่งกีดขวางได้ตามจำนวนอุปสรรคที่พบและแตกต่างกันออกไปในแต่ละครั้งที่เล่น ดังตารางบันทึกผลตารางที่ 4.1 ถึงตารางที่ 4.7 และรูปที่ 4.2 ซึ่งจากกราฟแสดงให้เห็นว่าตัวละครมีการพัฒนาการไม่หลบเลยมากที่สุดและสามารถหลบได้มากเพิ่มขึ้น แต่การหลบไม่ได้นั้นมีความไม่แน่นอนเกิดขึ้นเนื่องจากขึ้นอยู่กับจำนวนอุปสรรคที่ตัวละครได้พบ

5.3 ข้อเสนอแนะ

เกมวิ่งแข่งบนมือถือสามารถนำไปพัฒนาต่อกับการเล่นหลาย ๆ คนผ่าน Bluetooth Socket หรือนำไปทำเป็นเกมบนคอมพิวเตอร์ก็ได้ เพราะเป็นเกมที่เล่นในระยะเวลาที่สั้น ๆ เหมาะสำหรับผู้ที่ต้องการความเพลิดเพลินแต่เวลามีจำกัด

เอกสารอ้างอิง

[1] สุทัศน์ จิตรวัชรโกมล.ศิระวุฒิ จักร์ชูศรี. **"สร้างโปรแกรมและเกมส์ด้วย J2ME ดุณก็ทำได้"**. กรุงเทพมหานคร: บริษัท โปรลายน์ มีเดีย จำกัด

[2] พ.อ. เจนวิทย์ เหลืองอร่าม.ปิยวิทย์ เหลืองอร่าม. **"การเขียนโปรแกรมสำหรับ Wireless Applications ด้วย J2ME"**. กรุงเทพมหานคร: บริษัท ธรรมสาร จำกัด.2546

[3] กาญจนา ตันวิสุทธิ์.มณีโชติ สมานไทย. **"เขียนเกมและโปรแกรมบนมือถือJ2ME"**. นนทบุรี: ไอดีซีฯ. 2547

[4] John F. Canny. Douglas D. Edwards. Jitendra M. Malik. Sebastian Thrun. **"Artificial Intelligence A Modern Approach"**. United States of America: Pearson Education, Inc. 1997, 2003

[5] Wikimedia Foundation, Inc. "ปัญญาประดิษฐ์". [Online]. Available: http://th.wikipedia.org/wiki/AI. 2008

ภาคผนวก ก.

ตัวอย่างโปรแกรม

ตัวอย่างโค้ดบางส่วนเพื่อไว้ศึกษาเป็นตัวอย่างและทำการพัฒนาต่อ

```
/**
* Running.java
* ----------
* Class หลักเมื่อเปิดตัวเกมขึ้นมาซึ่งจะจัดการทางด้านการเลือกโหมดการเล่น
* การเริ่มต้นของโปรแกรมรวมถึงการหยุดการทำงานและทำลายการทำงานด้วย
*
* @author
* @date 2007 January 15
* @note JDK 1.5.0_05 (Windows XP SP2)
* @note J2ME Wireless toolkit
*/
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
import java.util.*;
import java.io.*;

public class Running extends MIDlet implements CommandListener
{
        private Display display;
        private RunningCanvas screen;
        private RunningClient screenClient;
        private static Running theApp;

        private final Command EXIT_CMD = new Command("Exit", Command.EXIT, 2);
        private final Command OK_CMD = new Command("Ok", Command.SCREEN, 1);

        private final Command exitCommand = new Command("Exit", Command.EXIT, 2);
        private final Command newCommand = new Command("Exit", Command.OK, 2);
```

```
private static final String[] elements = {"Story mode", "VS mode"};

private final List menu = new List("Running Bot Contest", List.IMPLICIT, elements,
                                   null);

/**
* Constructor
* @param
* @contract Running : ->
* @example new Running()
*/
public Running()
{
        display = Display.getDisplay(this);
        menu.addCommand(EXIT_CMD);
        menu.addCommand(OK_CMD);
        menu.setCommandListener(this);
}

/**
* เริ่มการทำงาน
* @param
* @contract startApp : -> void
* @example  new Running().startApp()
*/
public void startApp() throws MIDletStateChangeException
{
        show();
}
```

```
/**
* หยุดการทำงานชั่วคราว
* @param
* @contract pauseApp : -> void
* @example new Running().pauseApp()
*/
public void pauseApp()
{
}

/**
* ทำลายการทำงานของ Class
* @param unconditional the initial boolean for destroy
* @contract destroyApp : boolean -> void
* @example new Running().destroyApp(true)
* @example new Running().destroyApp(false)
*/
public void destroyApp(boolean unconditional)
{
        if(screen != null)
        {
                screen.destroy();
        }
        if(screenClient != null)
        {
                screenClient.destroy();
        }
}

/**
* function เกี่ยวกับเงื่อนไขเมื่อกด commandและเลือกระบบที่จะเล่น
* @param c the initial command that is choosed
```

```
* @param s the initial display
* @contract commandAction : Command Displayable -> void
* @example new Running().commandAction(exitCommand, Displayable)
*/
public void commandAction(Command c, Displayable s)
{
        if(c == exitCommand || c == newCommand)
        {
                destroyApp(false);
                notifyDestroyed();
        }
        if(c == EXIT_CMD)
        {
                destroyApp(false);
                notifyDestroyed();
                return;
        }
        switch(menu.getSelectedIndex())
        {
                case 0:
                        screen = new RunningCanvas(this);
                        screen.addCommand(exitCommand);
                        screen.addCommand(newCommand);
                        screen.setCommandListener(this);
                        display.setCurrent(screen);
                        screen.StartThread();
                        break;
                case 1:
                        screenClient = new RunningClient(this);
                        screenClient.addCommand(exitCommand);
                        screenClient.addCommand(newCommand);
                        screenClient.setCommandListener(this);
```

```
                    display.setCurrent(screenClient);
                    screenClient.StartThread();
                    break;
                default:
                    System.err.println("Unexpected choice...");
                }
            }

        /**
        * แสดงผลหน้าต่าง menu
        * @param
        * @contract show : -> void
        * @example new Running().show()
        */
        public void show()
        {
            Display.getDisplay(this).setCurrent(menu);
        }
}
```

```
/**
* Sprite.java
* ---------------
* Class เกี่ยวกับการคำนวณค่าต่างๆ
*
* @author
* @date 2007 January 17
* @note JDK 1.5.0_05 (Windows XP SP2)
* @note J2ME Wireless toolkit
*/
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import java.io.*;

public class Sprite
{
        public int[][] temp = new int[4][4];

        /**
        * Constructor
        * @param
        * @contract Sprite : ->
        * @example new Sprite()
        */
        public Sprite()
        {
        }

        /**
        * function เกี่ยวกับการ input sound มาเก็บไว้ในตัวแปร
        * @param string the initial name's music
        * @contract getMusic : String -> InputStream
```

```
* @example new Sprite().getMusic("run_race03") -> InputStream
* @example new Sprite().getMusic("run_race05") -> InputStream
* @example new Sprite().getMusic("run_event02") -> InputStream
*/
public InputStream getMusic(String string)
{
        InputStream is = getClass().getResourceAsStream("" + string);
        return is;
}

/**
* function คำนวณการวาดรูปว่าควรจะวาดตัวละครตัวไหนก่อน
* ซึ่งตัวละครที่อยู่ด้านบนจะต้องวาดก่อนแล้วตัวละครที่อยู่ด้านล่างต้องวาดทีหลัง
* เนื่องจากการซ้อนทับกันของภาพ
* @param point the initial point of character now
* @contract calGraphics : int[] -> void
* @example new Sprite().calGraphics({100, 48, 57, 80})
* @example new Sprite().calGraphics({50, 70, 90, 2})
* @example new Sprite().calGraphics({17, 140, 42, 66})
*/
public void calGraphics(int[] point)
{
        int keep;
        for(int i=0; i<4; i++)
        {
                temp[i][0] = i;             //เก็บลำดับของตัวละคร
                temp[i][1] = point[i];      //เก็บตำแหน่งของตัวละคร
        }
        //ทำการ sort
        for(int i=0; i<3; i++)
                for(int j=0; j<3; j++)
        {
```

```
                    //เมื่อเปลี่ยนตำแหน่งของตัวละคร ลำดับของตัวละครจะต้องเปลี่ยนด้วย
                    if(temp[j][1] > temp[j+1][1])
                    {
                            keep = temp[j][0];
                            temp[j][0] = temp[j+1][0];
                            temp[j+1][0] = keep;

                            keep = temp[j][1];
                            temp[j][1] = temp[j+1][1];
                            temp[j+1][1] = keep;
                    }
                }
            }
}
```

```
/**
* ImageSet.java
* ---------------
* Class เกี่ยวกับการตัดรูปใหญ่ให้เหลือขนาดที่ต้องการภายในรูปนั้น
*
* @author
* @date 2007 January 15
* @note JDK 1.5.0_05 (Windows XP SP2)
* @note J2ME Wireless toolkit
*/
import javax.microedition.lcdui.Image;
import javax.microedition.lcdui.Graphics;
import java.io.IOException;
```

```
public class ImageSet
{
        /**
        * Constructor
        * @param
        * @contract ImageSet : ->
        * @example new ImageSet()
        */
        public ImageSet()
        {
        }

        /**
        * ตัดรูปให้เหลือรูปที่ต้องการ
        * @param source the initial Image that want to cut
        * @param x the initial point width that want to cut
        * @param y the initial point height that want to cut
        * @param width the initial dimension width of new Image from point x,y
        * @param height the initial dimension height of new Image from point x,y
        * @contract extractFrames : Image int int int int -> Image
        * @example new ImageSet().extractFrames(Image, 16, 32, 16, 16) -> new Image
        * @example new ImageSet().extractFrames(Image, 27, 10, 35, 15) -> new Image
        * @example new ImageSet().extractFrames(Image, 10, 10, 32, 40) -> new Image
        */
        public final static Image extractFrames(Image source, int x, int y, int width, int height)
        {
                Image frames = Image.createImage(width, height);
                if(x + width > source.getWidth() || y + height > source.getHeight())
                {
                        System.out.println("Warning : attempting extract using(" + x + "," + y +
                        "," + width + "," + height + ") when image is " + "(" + source.getWidth()
                        + "," + source.getHeight() + ")");
```

```
                }
                frames.getGraphics().drawImage(source, -x, -y,
Tools.GRAPHICS_TOP_LEFT);
                return frames;
        }
}
```

```
/**
* RunningCanvas.java
* ----------
* Class ดำเนินการการเล่นของเกม
*
* @author
* @date 2007 January 15
* @note JDK 1.5.0_05 (Windows XP SP2)
* @note J2ME Wireless toolkit
*/
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import java.util.*;
import java.io.*;

class RunningCanvas extends Canvas implements Runnable
{
        private Running midlet;
        private Thread progress;
        private Graphics offGraphic;
        private InputStream music1, music2, music3;
        private Player play1, play2, play3;
        private Random rdm = new Random();
        private String str;
```

```
/**
* ตัวละครในเกมคำนวนการโจมตีคู่ต่อสู้
* @param i the initial number of bot 1-3
* @param rdo the initial other player
* @param rand the initial value for calculate percent hit
* @contract botCalHit : int int int -> boolean
* @example new RunningCanvas().botCalHit(1, 2, 100) == true
* @example new RunningCanvas().botCalHit(2, 2, 130) == false
* @example new RunningCanvas().botCalHit(3, 1, 245) == true
*/
public boolean botCalHit(int i, int rdo, int rand)
{
        //hitActing คือถ้ามีการโจมตีเกิดขึ้นก็จะส่งค่าเป็น true
        //status เป็นค่าสถานะ ในที่นี้ status=3 คือการโจมตี
        boolean hitActing = false;
        if( i != rdo && (status[rdo] == 0 || status[rdo] == 3) && sp[i] >= 12 &&
                charWidth[rdo] >= charWidth[i] && charWidth[rdo] <=
                                                        charWidth[i]+20 &&
                charHeight[rdo] >= charHeight[i]-4 && charHeight[rdo] <=
                                                        charHeight[i]+4 )
        {
                if(rand < perHit[i] && sp[i] > 8)
                {
                        //ถ้าหากได้โจมตีก็จะมีการพัฒนา
                        perHit[i] += Math.abs(rdm.nextInt())%7 + 4;
                        xxHit[i]++;
                        sp[i] -= 5;
                        status[i] = 3;
                        delay[i] = 1;
                        hitActing = true;
                }
        }
```

```
        return hitActing;
}

/**
 * bot calculate for escape danger
 * if move return true
 * else   return false
 * @param i the initial nmber bot 1-3
 * @param j the initial danger that will calculate
 * @param rand the initial percent random
* @param wall the initial point width of wall(rock)
* @contract botCalEscape : int int int int[] -> boolean
* @example new RunningCanvas().botCalEscape(1, 2, 250, {30, 200, 105, 200, 200, 4})
                                        == true
* @example new RunningCanvas().botCalEscape(2, 4, 370, {180, -2, 45, 7, 200, 200})
                                        == true
 * @example new RunningCanvas().botCalEscape(3, 5, 115, {115, 20, 106, 89, -40,
                                -33}) == false
*/
public boolean botCalEscape(int i, int j, int rand, int[] wall)
{
        int moveActing = 0;

        //การคำนวนการหลบหลีกจะแบ่งออกเป็น ลู่การวิ่งซึ่งมีทั้งหมด 4 ลู่
        //โดยจะวัดจากตำแหน่งของตัวละครนั้นๆว่าอยู่ในลู่ไหน
        if(charHeight[i]>=40 && charHeight[i]<=56)
        {
                for(int wd=0; wd<6; wd++)
                {
                        if(widthDanger1[wd] >= charWidth[i]-15 &&
                                widthDanger1[wd] <= wall[j]) break;
                        moveActing++;
```

```
                }
                if(moveActing == 6)
                {
                        //ทำการเคลื่อนที่ลง
                        botMove(i, "down");
                        if(perEscapeChoose[i] < 49)
                                perEscapeChoose[i] += Math.abs(rdm.nextInt())%2;
                }
        }
        else if(charHeight[i]>=57 && charHeight[i]<=72)
        {
                if(charHeight[i] <= 64+(rand%2))
                {
                        for(int wd=0; wd<6; wd++)
                        {
                                if(widthDanger0[wd] >= charWidth[i]-15 &&
                                        widthDanger0[wd] <= wall[j])
                                break;
                                moveActing++;
                        }
                        if(moveActing == 6)
                        {
                                botMove(i, "up");
                                if(perEscapeChoose[i] < 49)
                                perEscapeChoose[i] += Math.abs(rdm.nextInt())%2;
                        }
                }
                else
                {
                        for(int wd=0; wd<6; wd++)
                        {
                                if(widthDanger2[wd] >= charWidth[i]-15 &&
```

```
                                widthDanger2[wd] <= wall[j])
                            break;
                            moveActing++;
                        }
                        if(moveActing == 6)
                        {
                            botMove(i, "down");
                            if(perEscapeChoose[i] < 49)
                            perEscapeChoose[i] += Math.abs(rdm.nextInt())%2;
                        }
                    }
                }
                else if(charHeight[i]>=73 && charHeight[i]<=88)
                {
                    if(charHeight[i] <= 80+(rand%2))
                    {
                        for(int wd=0; wd<6; wd++)
                        {
                            if(widthDanger1[wd] >= charWidth[i]-15 &&
                                widthDanger1[wd] <= wall[j])
                            break;
                            moveActing++;
                        }
                        if(moveActing == 6)
                        {
                            botMove(i, "up");
                            if(perEscapeChoose[i] < 49)
                            perEscapeChoose[i] += Math.abs(rdm.nextInt())%2;
                        }
                    }
                    else
                    {
```

```
                for(int wd=0; wd<6; wd++)
                {
                        if(widthDanger3[wd] >= charWidth[i]-15 &&
                                widthDanger3[wd] <= wall[j])
                        break;
                        moveActing++;
                }
                if(moveActing == 6)
                {
                        botMove(i, "down");
                        if(perEscapeChoose[i] < 49)
                        perEscapeChoose[i] += Math.abs(rdm.nextInt())%2;
                }
            }
        }
        else if(charHeight[i]>=89 && charHeight[i]<=125)
        {
            for(int wd=0; wd<6; wd++)
            {
                if(widthDanger2[wd] >= charWidth[i]-15 &&
                    widthDanger2[wd] <= wall[j])
                break;
                moveActing++;
            }
            if(moveActing == 6)
            {
                botMove(i, "up");
                if(perEscapeChoose[i] < 49)
                perEscapeChoose[i] += Math.abs(rdm.nextInt())%2;
            }
        }
```

```
//เพิ่มทักษะความสามารถเมื่อมีการหลบหลีกหินได้
if(moveActing == 6)
{
        if(perEscape[i] < 180)
                perEscape[i] += Math.abs(rdm.nextInt())%5 + 1;
        if(perEscape[i] < 330)
                perEscape[i] += Math.abs(rdm.nextInt())%3 + 1;
        else
                perEscape[i] += Math.abs(rdm.nextInt())%2 + 1;
        return true;
}
else return false;
}

/**
* เมื่อมีการกดปุ่มจะ set ค่าให้เป็น true
* @param keyCode the initial key button
* @contract keyPressed : int -> void
* @example new RunningCanvas().keyPressed(4)
* @example new RunningCanvas().keyPressed(5)
* @example new RunningCanvas().keyPressed(9)
*/
public synchronized void keyPressed(int keyCode)
{
        int action = getGameAction(keyCode);
        switch(keyCode)
        {
                case KEY_NUM4:left = true;break;
                case KEY_NUM5:right = true;break;
                case KEY_NUM6:right = true;break;
                case KEY_NUM1:up = true;break;
                case KEY_NUM2:up = true;break;
```

```
                case KEY_NUM7:down = true;break;
                case KEY_NUM8:down = true;break;
                case KEY_NUM3:jump = true;break;
                case KEY_NUM9:hit = true;break;
                case KEY_NUM0:zero = true;break;
                case KEY_STAR:star = true;break;
                case KEY_POUND:pound = true;break;
        }
        switch(action)
        {
                case LEFT:left = true;break;
                case RIGHT:right = true;break;
                case UP:up = true;break;
                case DOWN:down = true;break;
        }
}

/**
* เมื่อมีการปล่อยปุ่มจะ set ค่าให้เป็น false
* @param keyCode the initial key button
* @contract keyReleased : int -> void
* @example new RunningCanvas().keyReleased(4)
* @example new RunningCanvas().keyReleased(5)
* @example new RunningCanvas().keyReleased(9)
*/
public synchronized void keyReleased(int keyCode)
{
        int action = getGameAction(keyCode);
        switch(keyCode)
        {
                case KEY_NUM4:left = false;break;
                case KEY_NUM5:right = false;break;
```

```
                case KEY_NUM6:right = false;break;
                case KEY_NUM1:up = false;break;
                case KEY_NUM2:up = false;break;
                case KEY_NUM7:down = false;break;
                case KEY_NUM8:down = false;break;
                case KEY_NUM3:jump = false;break;
                case KEY_NUM9:hit = false;break;
                case KEY_NUM0:zero = false;break;
                case KEY_STAR:star = false;break;
                case KEY_POUND:pound = false;break;
            }
            switch(action)
            {
                case LEFT:left = false;break;
                case RIGHT:right = false;break;
                case UP:up = false;break;
                case DOWN:down = false;break;
            }
        }
}
```

สำหรับการพัฒนาต่อ

```
/**
* RunningClient.java
* ----------
* Class เกี่ยวกับ Client ที่จะติดต่อเข้ากับ Server
* สำหรับการเล่นแบบ MultiPlayer
*
* @author
* @date 2007 September 13
* @note JDK 1.5.0_05 (Windows XP SP2)
* @note J2ME Wireless toolkit
*/
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import javax.microedition.io.*;
import java.util.*;
import java.io.*;

class RunningClient extends Canvas implements Runnable
{
        private StreamConnection streamConnection = null;
        private OutputStream outputStream = null;
        private DataOutputStream dataOutputStream = null;
        private InputStream inputStream = null;
        private DataInputStream dataInputStream = null;
        private String connectString;
        private StringBuffer results;
        private StringItem resultField;

        /**
        * Constructor
        * @param midlet the initial class Running
```

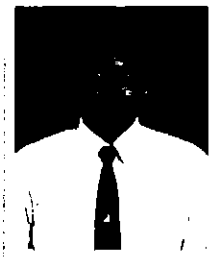
```
* @contract RunningClient : Running ->
* @example new RunningClient(new Running())
*/
public RunningClient(Running midlet)
{
        this.midlet = midlet;

        setClass();
        newGame();
        progress = new Thread(this);
}

/**
* ส่งค่าและรับค่า Server-Client
* @param act the initial message that will send to server
* @contract sendAndRecieve : String -> void
* @example new RunningClient().sendAndRecieve("13")
 * @example new RunningClient().sendAndRecieve("24")
 * @example new RunningClient().sendAndRecieve("2E")
*/
public void sendAndRecieve(String act)
{
        try{
                streamConnection = (StreamConnection)
                Connector.open(connectString);

                outputStream = streamConnection.openOutputStream();
                dataOutputStream = new DataOutputStream(outputStream);

                dataOutputStream.writeChars(act);
                dataOutputStream.flush();
```

```
inputStream = streamConnection.openInputStream();
dataInputStream = new DataInputStream(inputStream);

int input;
int[] sms = new int[80];
int i = 0;
while( (input = dataInputStream.read()) != 13)
{
        sms[i] = input;
        i++;
}
if(i != 0) caseRecieve(sms);

}catch(IOException e){
    System.err.println("Exception caught:" + e);
}finally{
    try{
        if(dataInputStream != null)
            dataInputStream.close();
    }catch(Exception ignored){}
    try{
        if(dataOutputStream != null)
            dataOutputStream.close();
    }catch(Exception ignored){}
    try{
        if(outputStream != null)
            outputStream.close();
    }catch(Exception ignored){}
    try{
        if(inputStream != null)
            inputStream.close();
    }catch(Exception ignored){}
```

```
                    try{
                    if(streamConnection != null)
                        streamConnection.close();
                    }catch(Exception ignored){}
                }
            }
        }
```

```
}
```

ประวัติผู้เขียนโครงงาน



ชื่อ นายณัฐพล เกงขุนทด

ภูมิลำเนา 801 หมู่ 8 ทัณฑสถานหญิงพิษณุโลก ต.วังทอง อ.วังทอง จ.พิษณุโลก 65130

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจาก โรงเรียนขาณุวิทยา
- ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail thaiiluminati@hotmail.com

ชื่อ นางสาววิมลรัตน์ ไพโรจน์

ภูมิลำเนา 112 หมู่ 12 ต.ทุ่งหลวง อ.คีรีมาศ จ.สุโขทัย 64160

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจาก โรงเรียนอุดมดรุณี
- ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail kai_ko_l_@hotmail.com

ชื่อ นายกิตติพัฒน์ บุญพร้อม

ภูมิลำเนา 68/3 หมู่ 8 ต.หย่วน อ.เชียงคำ จ.พะเยา 56110

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจาก โรงเรียนเชียงคำวิทยาคม
- ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail phaiphai000@hotmail.com