สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาโปรแกรมทดสอบหุ่นยนต์รบที่ควบคุมโดยภาษาจาวา

DEVELOP ONLINE ROBOT GAME TESTING FIELD WITH CONTROL BY JAVA

T 058782

พันธุ์ธิดา จันทร์ดา
พิชญ์สิรี ทฤฆชนม์
วิทยา เหมือนเดช

วพ
ว569ก
2547
เลขหมู่ 58782
เลขทะเบียน
วัน,เดือน,ปี 10 ก.พ. 2549

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b. 11458157
i

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

DEVELOP ONLINE ROBOT GAME TESTING FIELD WITH CONTROL BY JAVA

PANTIDA CHANDA
PICHSIREE THARUEKACHON
WITTAYA MUANDET

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ การพัฒนาโปรแกรมทดสอบหุ่นยนต์รบที่ควบคุมโดยภาษาจาวา
DEVELOP ONLINE ROBOT GAME TESTING FIELD WITH CONTROL BY JAVA

ชื่อนักศึกษา นางสาวพันธุ์ธิดา จันทร์ดา 44050437
นางสาวพิชญ์สิรี ทฤฆชนม์ 44050438
นายวิทยา เหมือนเดช 44050459

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา อ.วิสันต์ ตั้งวงษ์เจริญ
อ.ชาญชัย ดีอ่วม

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2547

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	ดร.กรกช ประชุมรักษ์	
กรรมการ	ผศ.ดร.จีรพร ศรีสวัสดิ์	
กรรมการและอาจารย์ที่ปรึกษา	อ.วิสันต์ ตั้งวงษ์เจริญ	
กรรมการและอาจารย์ที่ปรึกษา	อ.ชาญชัย ดีอ่วม	

(รองศาสตราจารย์ ดร.วีระ บุญจริง)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ การพัฒนาโปรแกรมทดสอบหุ่นยนต์รบที่ควบคุมโดยภาษาจาวา

ชื่อนักศึกษา นางสาวพันธุ์ธิดา จันทร์ดา 44050437

นางสาวพิชญ์สิรี ทฤฆชนม์ 44050438

นายวิทยา เหมือนเดช 44050459

ปริญญา วิทยาศาสตรบัณฑิต

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2547

อาจารย์ที่ปรึกษา อ.วิสันต์ ตั้งวงษ์เจริญ

อ.ชาญชัย ดีอ่วม

บทคัดย่อ

ปัจจุบันโปรแกรมภาษาเชิงวัตถุได้รับความนิยมเป็นอย่างมาก เนื่องจากในการพัฒนาโปรแกรม โดยเฉพาะการพัฒนาต่อจากเดิมสามารถกระทำได้โดยง่าย แต่ขั้นตอนในการศึกษาหลักการเขียนโปรแกรมภาษาเชิงวัตถุค่อนข้างซับซ้อนและต้องใช้เวลาในการศึกษานาน ดังนั้นจึงได้ทำการพัฒนาเกมหุ่นยนต์รบที่ควบคุมด้วยภาษาจาวา เพื่อใช้เป็นเครื่องมือแก่ผู้ที่ทำการศึกษาการโปรแกรมเชิงวัตถุ

การการพัฒนาโปรแกรมทดสอบหุ่นยนต์รบที่ควบคุมด้วยภาษาจาวานี้เป็นการพัฒนาต่อจากรุ่นที่แล้ว ซึ่งในการพัฒนาครั้งนี้จะทำการสร้างสนามทดสอบหุ่นยนต์ โดยลักษณะสนามทดสอบที่จะสร้างนั้นเป็นสนามที่มีสิ่งกีดขวางต่าง เป็นอุปสรรค์ในการเดิน และการค้นหาศัตรูของหุ่นยนต์ผู้เล่น เพื่อให้ผู้เล่นได้ฝึกทักษะในการจัดการกับ สถานการต่างๆ ที่เกิดขึ้นภายในสนาม โดยการเล่นในลักษณะนี้ผู้พัฒนาได้ตั้งชื่อว่า Training Mode โดยในส่วน Training Mode นั้น จะทำการสร้างสนามที่มีลักษณะต่างๆ กัน 9 สนาม ในแต่ละสนามจะแบ่งระดับความยากง่ายออกเป็นอีก 3 ระดับ และในแต่สนามจะมีหุ่นยนต์ผู้เล่นเพียง 1 ตัว ต่อสู่กับหุ่นยนต์ของฝ่ายศัตรูตั้งแต่ 1 ตัวขึ้นไป หุ่นยนต์ของผู้เล่นจะตั้งทำลายศัตรูให้หมด ภายในระยะเวลาที่กำหนด จึงจะถือว่าเป็นฝ่ายชนะ นอกจากนั้นยังจะพัฒนากรรมวิธีในการตรวจสอบสิ่งกีดขวางของหุ่นยนต์ให้สามารถทำการรายงานผลการตรวจ ได้ 3 ทิศทาง คือ ด้านซ้าย ด้านหน้า และด้านขวา ของหุ่นยนต์ เพื่อให้หุ่นยนต์สามารถดำเนินจนบรรลุเป้าหมายของเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	DEVELOP ONLINE ROBOT GAME TESTING FIELD WITH CONTROL BY JAVA
Students	Miss.Pantida Chanda 44050437
	Miss.Pichsiree Tharuekachon 44050438
	Mr.Wittaya Muandet 44050459
Degree	Bachelor of Science
Department	Mathematics and Computer Science, Faculty of Science
Programme	Computer Science
Acdemic Year	2004
Special Project Advisor	Mr.Wisan Tangwongcharoen
	Mr.Chanchai Dee-uam

ABSTRACT

Object oriented programming technology now plays an important role in computer programming. Especially because of their benefits such as module developing. To understand the concept of object oriented programming is quite difficult, complicated and take much time. The JavaRobot Game is developed in order to be a intermediary of education.

The JavaRobot project has developed from the last version. In this version has created Training Mode and use 9 JavaRobot's testing fields which have walls to obstruct the way. The users must handle the situations which happen in the field. Each field is seperated into three levels and one player's JavaRobot fighting with one enemy or more. The users have to destroy all of the enemies during the time. Not only abovementioned but also developed the process of checking the wall to report the result in three ways: left, right and front of the JavaRobot.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องการพัฒนาโปรแกรมทดสอบหุ่นยนต์รบที่ควบคุมโดยภาษาจาวา สามารถสำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำขอขอบพระคุณ อาจารย์วิสันต์ ตั้งวงษ์เจริญ และอาจารย์ชาญชัย ดีอ่วม อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำ และเป็นที่ปรึกษาในการแก้ปัญหาต่างๆรวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้และขอขอบพระคุณ อาจารย์ทุกๆท่าน ที่ให้คำแนะนำและให้ข้อคิดดีๆ ในการทำปัญหาพิเศษนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ ในการทำปัญหาพิเศษนี้ให้สำเร็จไปได้ด้วยดี รวมทั้งเพื่อนๆพี่ๆน้องๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษ ไว้ ณ ที่นี้

คณะผู้จัดทำ

มีนาคม 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป..... VIII

บทที่ 1 บทนำ 1

1.1 ความสำคัญและที่มาของปัญหา.....1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตของปัญหา.....2

1.4 ประโยชน์ที่คาดว่าจะได้รับ.....2

1.5 ขั้นตอนในการดำเนินการ.....2

บทที่ 2 ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง 3

2.1 การทำงานแบบเทรด.....3

2.1.1 ความหมายของเทรด.....3

2.1.2 การทำงานของเทรด.....3

2.1.3 เทรดในภาษาจาวา.....5

2.1.4 รายละเอียดการทำงานของคลาสเทรด.....5

2.1.5 รายละเอียดและการทำงานของอินเตอร์เฟสรันเอบิล.....6

2.1.6 เมธตอดที่สำคัญของเทรดในภาษาจาวา.....7

2.1.7 วิธีการใช้ฟังก์ชันเดมอนเทรด.....8

2.1.8 วิธีการใช้ฟังก์ชันเรซิ่งคอนดิชัน.....9

2.1.9 วิธีการใช้ฟังก์ชันซิงโครไนเซชัน.....9

2.1.10 วิธีการใช้ฟังก์ชันเดดล็อค.....9

2.1.11 วัฏจักรของเทรด.....10

2.1.12 ลำดับความสำคัญของเทรดและการจัดลำดับงาน.....11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

2.2 คอมโพเนนต์.....13
2.3 สถาปัตยกรรม 2-Tier.....14
2.3.1 ข้อดีของสถาปัตยกรรม 2-Tier.....14
2.4 หลักในการสร้างเกม.....15
2.4.1 ทฤษฎีเกม.....15
2.4.1.1 ความหมายของทฤษฎีเกม.....16
2.4.1.2 ประโยชน์ของทฤษฎีเกม.....17
2.4.2 โครงสร้างเกม.....17
2.4.3 โครงสร้างโปรแกรม.....18
2.4.4 การประเมินและการออกแบบ.....18
2.4.5 สรุปการออกแบบก่อนทำการเขียนโปรแกรม.....18
2.4.6 การเขียนโปรแกรม.....18
2.4.7 การทดสอบการเล่นเกม.....18

บทที่ 3 การพัฒนาและการออกแบบโปรแกรม.....19
3.1 แนวทางการพัฒนาโปรแกรม.....19
3.2 การออกแบบสนาม.....20
3.2.1 สนามทดสอบกรรมวิธีการเดินแบบปกติ.....20
3.2.2 สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ.....21
3.3 การออกแบบสถาปัตยกรรมที่ใช้ในเกม.....25
3.4 การออกแบบไดอะแกรมของเกม.....26
3.1.1 คลาสไดอะแกรมของสนามทดสอบ.....26
3.1.2 ซีเควนซ์ไดอะแกรมของการส่งหุ่นยนต์เข้าสนามทดสอบ.....29
3.1.3 ซีเควนซ์ไดอะแกรมของการส่งข้อความระหว่าหุ่นยนต์กับแถวคอย.....29
3.5 อัลกอริทึมที่ใช้ในการเขียนเกม.....30
3.5.1 การแบ่งส่วนของสนาม.....30
3.5.2 อัลกอริทึมการสื่อสารระหว่างวัตถุด้วยการส่งผ่านข้อความ.....34
3.5.3 อัลกอริทึมการตรวจสอบสิ่งกีดขวางแบบเรดาร์.....36
3.5.4 อัลกอริทึมการตรวจสอบการชน.....37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

บทที่ 4 การใช้งานโปรแกรม.....38
4.1 โครงสร้างของโปรแกรม.....38
4.1.1 โปรแกรมสำหรับสร้างหุ่นยนต์.....41
4.1.2 เว็บไซต์เกมจาวาโรบอท.....41
4.2 ขั้นตอนในการเล่นเกม.....42
4.2.1 การดาวน์โหลดโปรแกรมและการสร้างหุ่นยนต์.....42
4.2.2 การเข้าสู่ระบบและการนำหุ่นยนต์ที่สร้างไปใช้งาน.....46
4.2.3 การเล่นแบบเทรนนิ่งโหมด.....47
บทที่ 5 สรุปผลการศึกษา ข้อเสนอแนะ และแนวทางการพัฒนาต่อ.....52
5.1 สรุปผลปัญหาพิเศษ.....52
5.2 ปัญหาและข้อจำกัด.....52
5.3 ข้อเสนอแนะและแนวทางพัฒนาต่อ.....52

บรรณานุกรม.....54

ภาคผนวก ก. การติดตั้งโปรแกรม.....55

ภาคผนวก ข. คู่มือการใช้โปรแกรม.....56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่

2.1 แสดงข้อแตกต่างระหว่างเทรดกับการประมวลผล....4
2.2 แสดงคุณสมบัติของดาตาเมมเบอร์....5
2.3 คุณสมบัติเมธตอดของเทรด....7
2.4 คุณสมบัติของเมธตอดที่ inherit จากคลาส java.lang.Object....8
2.5 ตารางแสดงข้อมูลของเทรดที่จะเข้าประมวลผลใน หน่วยประมวลผลกลาง....12
2.6 แสดงการจัดการเข้าประมวลผลของเทรดตามลำดับความสำคัญ....12
3.1 แสดงลำดับของพารามิเตอร์ของรูปแบบที่ 1....35
3.2 แสดงลำดับของพารามิเตอร์ของรูปแบบที่ 2....35
3.3 แสดงลำดับของพารามิเตอร์ของรูปแบบที่ 3....36
4.1 แสดงรายละเอียดของโปรแกรมสำหรับสร้างหุ่นยนต์....41
4.2 แสดงรายละเอียดของเว็บไซต์....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่

2.1 แสดงวัฏจักรของเทรด....10
2.2 แสดงการรูปแบบเข้าถึงข้อมูลของสถาปัตยกรรม 2-Tier....14
3.1 สนามทดสอบกรรมวิธีการเดินแบบปกติ ชนิดก้นหอย....20
3.2 สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ ชนิดที่ 1....21
3.3 สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ ชนิดที่ 2....21
3.4 สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ ชนิดที่ 3....22
3.5 สนามทดสอบแบบที่ 1....22
3.6 สนามทดสอบแบบที่ 4....23
3.7 สนามทดสอบแบบที่ 9....23
3.8 สนามทดสอบแบบที่ 2....23
3.9 สนามทดสอบแบบที่ 3....24
3.10 สนามทดสอบแบบที่ 5....24
3.11 สนามทดสอบแบบที่ 6....24
3.12 สนามทดสอบแบบที่ 7....25
3.13 สนามทดสอบแบบที่ 8....25
3.14 คลาสไดอะแกรมของสนามทดสอบ....26
3.15 ซีเควนซ์ไดอะแกรมของการส่งหุ่นยนต์เข้าสนามทดสอบ....29
3.16 ซีเควนซ์ไดอะแกรมของการส่งข้อความระหว่างหุ่นยนต์กับหน่วยประมวลผลข้อความ....29
3.17 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 1....30
3.18 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 2....30
3.19 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 3....31
3.20 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 4....31
3.21 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 5....32
3.22 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 6....32
3.23 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 7....33
3.24 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 8....33
3.25 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 9....34
3.26 การรายงานผลของการตรวจสอบสิ่งกีดขวางแบบเรดาร์....37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.27 ตำแหน่งที่ใช้ในการตรวจสอบการชน....37
4.1 แสดงแผนผังการทำงานของโปรแกรม....40
4.2 แสดงหน้าแรกของเว็บไซต์ JavaRobot....42
4.3 แสดงหน้าดาวน์โหลดโปรแกรมสร้างหุ่นยนต์และโปรแกรมที่จำเป็น....43
4.4 แสดงหน้าแรกของโปรแกรมที่ใช้ในการสร้างหุ่นยนต์....43
4.5 แสดงหน้าจอการเลือกไฟล์หุ่นยนต์ที่ต้องการแก้ไข....44
4.6 แสดงหน้าที่ใช้เลือกส่วนประกอบ....44
4.7 แสดงหน้าจอที่ทำการเลือกขา....45
4.8 แสดงหน้าที่ใช้สำหรับเขียนโปรแกรมบังคับหุ่นยนต์....45
4.9 แสดง message box หลังจากคอมไพล์โปรแกรมผ่าน....45
4.10 แสดงหน้าที่ให้ผู้เล่นทำการใส่ข้อมูลเพื่อสมัครสมาชิก....46
4.11 แสดงหน้าข้อมูลของสมาชิก....46
4.12 แสดงหน้า Upload....47
4.13 รูปแบบสนาม 9 แบบในเทรนนิ่งโหมด....48
4.14 แสดงหน้าให้ผู้เล่นเลือกสนาม....49
4.15 แสดงหน้าให้ผู้เล่นเลือกระดับการเล่น....49
4.16 แสดงหน้าให้ผู้เล่นเลือกหุ่นยนต์....50
4.17 แสดงตัวอย่างหน้าแสดงผลการต่อสู้ของสนามทดสอบกรรมวิธีการเดินแบบปกติ....50
4.18 แสดงตัวอย่างหน้าแสดงผลการต่อสู้ของสนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ....51
ข-1 แสดงหน้าแรกเพื่อให้ผู้เล่นเลือกการทำงาน....57
ข-2 แสดงหน้าจอในการเลือกไฟล์หุ่นยนต์ที่ต้องการนำมาแก้ไข....57
ข-3 แสดงหน้าจอในการเลือกส่วนประกอบของหุ่นยนต์....58
ข-4 แสดงหน้าจอที่ทำการเลือกลำตัว....58
ข-5 แสดงหน้าจอเมื่อทำการกด HELP....59
ข-6 แสดงหน้าจอที่ใช้สำหรับเขียนโปรแกรมบังคับหุ่นยนต์....59
ข-7 แสดงหน้าจอเมื่อคอมไพล์โปรแกรมที่ผู้เล่นเขียนขึ้นผ่าน....60
ข-8 แสดงหน้าแรกของเว็บ JavaRobot....60
ข-9 แสดงหน้า What's Up....61
ข-10 แสดงหน้า Rule....61
ข-11 แสดงหน้า Service....62
ข-12 แสดงหน้าดาวน์โหลดโปรแกรมที่จำเป็นในการเล่นเกม....62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข-13 แสดงหน้า About.....63
ข-14 แสดงหน้า Sign Up.....63
ข-15 แสดงหน้า User Page.....64
ข-16 แสดงหน้า Training Mode.....64
ข-17 แสดงหน้า Upload.....65
ข-18 แสดงหน้าดาวน์โหลด.....65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ปัจจุบันโปรแกรมภาษาเชิงวัตถุ (Object Oriented) ได้รับความนิยมเป็นอย่างมาก เนื่องจากทางผู้พัฒนาระบบได้มองเห็นข้อดีของการนำแนวคิดเชิงวัตถุมาประยุกต์ใช้ ซึ่งข้อดีของการประยุกต์ใช้แนวคิดเชิงวัตถุมีดังต่อไปนี้

- สนับสนุนการพัฒนาระบบที่ซับซ้อน เนื่องจากนักพัฒนาโปรแกรมจะทำการวิเคราะห์ออกแบบระบบกันในระดับของออบเจ็กต์ ดังนั้นจึงกล่าวได้ว่าการอาศัยแนวคิดเชิงวัตถุจะช่วยให้ผู้พัฒนาสามารถสร้างโปรแกรมได้ง่ายขึ้น
- สนับสนุนการนำกลับมาใช้งานอีกครั้ง เนื่องจากคลาสหรือออบเจ็กต์ที่กำหนดขึ้นจะมีความสมบูรณ์อยู่ในตัวเองบนพื้นฐานแนวคิดของแต่ละออบเจ็กต์เอง ดังนั้นแต่ละคลาสจึงง่ายต่อการนำกลับมาใช้งานปรับปรุงเพิ่มเติม
- ปรับปรุงแก้ไขและบำรุงรักษาง่าย เนื่องจากข้อมูลและฟังก์ชันการทำงานที่เกี่ยวข้องกับออบเจ็กต์หนึ่งๆ จะถูกรวบรวมอยู่ที่เดียวกัน การทำงานภายในของแต่ละออบเจ็กต์จะไม่เกี่ยวข้องพึ่งพิงกับโค้ดที่อยู่ภายนอกออบเจ็กต์ ดังนั้นนักพัฒนาสามารถทำการแก้ไขปรับปรุงรายละเอียดภายในของแต่ละคลาสได้โดยไม่กระทบต่อส่วนที่เรียกใช้งานภายนอกแต่อย่างใด นอกจากนี้ในการขยายระบบก็สามารถทำได้ง่ายๆ โดยการสร้างออบเจ็กต์หรือคลาสเพิ่มเติมลงไปในตัวโปรแกรมนั่นเอง

แต่เนื่องจากหลักการของโปรแกรมภาษาเชิงวัตถุเป็นหลักการที่ค่อนข้างจะมีความซับซ้อน ทำให้มองภาพการทำงานของวัตถุนั้น เป็นสิ่งที่ทำได้ยาก เพื่อเป็นการส่งเสริมให้ผู้ที่มีความสนใจในภาษาเชิงวัตถุเข้าใจหลักการต่างๆ และมองเห็นภาพการทำงานของวัตถุ จึงใช้เกมเป็นสื่อที่ช่วยในการเรียนรู้

1.2 วัตถุประสงค์

1) เพื่อเพิ่มเติมรูปแบบสนามของเกม Java Robot Online ให้มีความหลากหลาย

2) พัฒนาโปรแกรมทดสอบความสามารถของหุ่นยนต์ โดยมีการจำลองสถานการณ์ต่างๆขึ้น ซึ่งหุ่นยนต์ต้องสามารถจัดการกับสถานการณ์จำลองที่เกิดขึ้นได้

3) เพื่อใช้เกมเป็นเครื่องมือในการเรียนรู้ และเสริมสร้างความเข้าใจในหลักการของภาษาเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ใช้เกมเป็นแรงจูงใจ ทำให้ผู้เล่นสนใจศึกษาเนื้อหาภาษาเชิงวัตถุ จากการที่ผู้เล่นได้ทำการเขียนโปรแกรมบังคับหุ่นยนต์

1.3 ขอบเขตของปัญหา

ทำการเพิ่มรูปแบบสนามในการทดสอบหุ่นยนต์ จากเดิมสนามทดสอบจะเป็นสนามรูปสี่เหลี่ยมผืนผ้าไม่มีสิ่งกีดขวาง โดยทำการเพิ่มเติมสิ่งกีดขวาง ทางแยก และทำการเพิ่มเติมสนามทั้งหมด 9 สนาม

ทำการปรับปรุงการรายงานการตรวจหาขอบสนาม จากเดิมเมื่อตรวจพบของสนามแล้ว จะรายงานทิศทางที่พบขอบสนามว่าตรวจพบขอบสนามทางทิศใด เพียงทิศทางเดียว ให้สามารถรายงานการตรวจสอบการพบขอบสนามได้ 3 ทิศทาง คือ ทางด้านซ้าย ทางด้านขวา และทางด้านหน้า พร้อมกัน พร้อมทั้งบอกระยะห่างระว่างหุ่นยนต์กับขอบสนามในแต่ละทิศทาง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้เกม Java Robot Online ที่มีรูปแบบของสนามทดสอบ 5 แบบ
2) ได้เกมที่เป็นสื่อกลาง ที่ช่วยให้มองเห็นภาพในการศึกษาเรียนรู้การโปรแกรมภาษาเชิงวัตถุ

1.5 ขั้นตอนในการดำเนินการ

1) ทำการศึกษาและทำความเข้าใจในหลักการโปรแกรมภาษาเชิงวัตถุ ภาษาจาวาและเว็บโปรแกรมมิ่ง (Web Programming)
2) เก็บรวบรวมข้อมูลและตัวอย่างที่เกี่ยวข้องกับเกม
3) วิเคราะห์และออกแบบสนามทดสอบ
4) สร้างสนามทดสอบ
5) เขียนโค้ดโปรแกรม สนามทดสอบตามที่ได้ออกแบบไว้
6) ทดสอบการทำงานของของสนามทดสอบ
7) แก้ไขข้อบกพร่องและทำการปรับปรุง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง

2.1 การทำงานแบบเทรด

2.1.1 ความหมายของเทรด

ระบบปฏิบัติการที่แพร่หลายในทุกวันนี้อย่างเช่นยูนิกซ์ (UNIX) และวินโดวส์ (Windows) สามารถรับโปรแกรมเข้าไปทำงานได้มากกว่าหนึ่งโปรแกรมในเวลาเดียวกัน ทั้งที่อาจมีหน่วยประมวลผลเพียงตัวเดียว คุณสมบัติของระบบปฏิบัติอย่างนี้เรียกว่ามัลติทาสกิ้ง (multitasking) เป็นกลไกที่จะช่วยใช้งานระบบปฏิบัติประมวลยังได้ผลอย่างมีประสิทธิภาพมากขึ้น

บางระบบปฏิบัติการอาจกำหนดเวลาที่แต่ละการประมวลผลจะได้ทำงาน หากการประมวลผลใดใช้เวลาทำงานเกินเวลาที่กำหนดให้ ก็จะหยุดและออกมาเข้าคิวรอรอบต่อไป และโปรแกรมที่รอคิวจะเข้าไปทำงานแทน ความเร็วในการประมวลผลของคอมพิวเตอร์ทำให้การทำงานลักษณะนี้ดูคล้ายกับว่าโปรแกรมเหล่านั้นทำงานไปพร้อมๆ กันทั้งที่จริงแล้วยังเป็นการเข้าทำงานทีละโปรแกรม โปรแกรมส่วนใหญ่มักต้องทำงานหลายอย่างไปพร้อมๆ กัน เช่น ในขณะที่ส่วนหนึ่งของโปรแกรมกำลังแสดงภาพเคลื่อนไหวบนจอภาพ ในเวลาเดียวกับที่อีกส่วนหนึ่งกำลังสร้างภาพเคลื่อนไหวที่จะแสดงต่อไป หากงานทั้งสองนี้ถูกแยกกันทำเป็นสองการประมวลผล และทำเป็นเวลานานจะทำให้เห็นการภาพเคลื่อนไหวที่จอภาพสะดุดเป็นช่วงๆ เนื่องจากมีการสลับการทำงานระหว่างการประมวลผลที่สร้างภาพเคลื่อนไหวกับการประมวลผลที่แสดงภาพเคลื่อนไหว ซึ่งการเปลี่ยนการทำงานของการประมวลผลหนึ่งไปสู่อีกการประมวลผลได้ซึ่งเรียกว่า คอนเท็กซ์ สวิตชิ่ง (Context Switching) ในขั้นตอนนี้ต้องมีการเก็บสถานะการคำนวณของการประมวลผลเดิมไว้ เพื่อให้สามารถกลับมาทำงานต่อจากจุดนั้นได้ และต้องโหลดสถานะการคำนวณของการประมวลผล ใหม่เข้าไปทำงานแทนที่ภาระการเปลี่ยนการทำงานของการประมวล ผลหนึ่งไปสู่อีกการประมวลผลนี้เป็นโอเวอร์เฮด (Overhead) ที่ทำให้โปรแกรมงานทำงานช้าลง ผู้ออกแบบระบบปฏิบัติการจึงหาวิธีทำให้โอเวอร์เฮดของการทำคอนเท็กซ์ สวิตชิ่งลดลงด้วยการเสนอกลไกใหม่ที่เรียกว่าเทรด

2.1.2 การทำงานของเทรด

ระบบปฏิบัติการแบบมัลติเทรดดิ้ง (Multi-Threading) จะสามารถแบ่งการประมวลผลออกเป็นหน่วยที่เล็กๆ ซึ่งเรียกว่าเทรด หรือบางทีเรียกว่า light-weight Process ระบบปฏิบัติการแบบนี้จะสามารถสร้างหลายเทรดในหนึ่งการประมวลผล โดยปกติเทรดหนึ่งคือ execution flow ของทำงานอย่างหนึ่ง หากนำหลายๆเทรดมารวมกันในการประมวลผลหนึ่งจะช่วยให้การประมวลผลนั้นสามารถรันได้ในเวลาเดียวกัน นั่นหมายความว่า สามารถทำงานได้มากกว่าหนึ่งอย่างไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พร้อมๆ กันโดยไม่ต้องมีการทำการเปลี่ยนการทำงานของการประมวลผลหนึ่งไปสู่อีกการประมวลผลโปรแกรมจึงทำงานได้เร็ว และเป็นการใช้งานหน่วยประมวลผลได้อย่างมีประสิทธิภาพมากขึ้น

ในระบบมัลติทาสกิ้งโดยปกติหนึ่งการประมวลผลจะมีหนึ่ง shared variable และถ้าเทรดหนึ่งมีตัวแปรของมันเองที่ไม่ถูกใช้โดยเทรดอื่น ก็เรียกตัวแปรนั้นว่า local variable โดยปกติ เทรดหนึ่งจะทำงานในบางส่วนของโปรแกรมเท่านั้น แต่ละเทรดจะมีโปรแกรมของตัวเองมีจุดเริ่มต้นและจุดสิ้นสุดแตกต่างกันไป เทรดแต่ละตัวมักจะทำงานเป็นอิสระต่อกัน ทำให้เทรดหลายๆ ตัวสามารถทำงานพร้อมกันได้ ถ้าเครื่องคอมพิวเตอร์นั้นมีหน่วยประมวลผลหลายตัวก็อาจแบ่งให้หน่วยประมวลผลแต่ละตัวทำงานแต่ละเทรดได้พร้อมกันแบบขนาน(Parallel) ซึ่งจะช่วยให้โปรแกรมทำงานเร็วขึ้น

ระบบปฏิบัติการมีหน้าที่ควบคุมว่าเทรดใดถูกทำงานเมื่อใด และโดยหน่วยประมวลใด ซึ่งผู้เขียนโปรแกรมไม่สามารถควบคุมได้โดยตรง ผู้เขียนโปรแกรมทำได้แต่เพียงสร้างเทรดให้ทำหน้าที่และควบคุมให้เทรดเหล่านั้นเริ่มต้นทำงาน และหยุดการทำงาน ในระหว่างที่เทรดทำงานโปรแกรมยังสามารถออกคำสั่งให้เทรดนั้นหยุดรอ และเริ่มต้นทำงานเมื่อถึงเวลาที่กำหนดได้

ข้อแตกต่างระหว่างเทรดกับการประมวลผล (Process) สรุปได้ดังนี้คือ

ตารางที่ 2.1 แสดงข้อแตกต่างระหว่างเทรดกับการประมวลผล

เทรด	การประมวลผล
แต่ละเทรดของการประมวลผลเดียวกันจะทำงานแตกต่างกันแต่ต้องทำงานอยู่ภายในสภาพแวดล้อมเดียวกัน จึงอาจมีการใช้ตัวแปรบางตัวหรือโปรแกรมบางส่วนร่วมกัน	แต่ละการประมวลผลจะมีสภาพแวดล้อมของตัวเอง
การติดต่อระหว่างเทรดทำได้ง่ายโดยใช้ shared variables	เมื่อเปลี่ยนการทำงานของการประมวลผลหนึ่งไปสู่อีกการประมวล จะเปลี่ยนสภาพแวดล้อมของการประมวลผลด้วย การส่งค่าให้แก่กันทำได้ผ่านทางไฟล์ เรียกว่าไพพ์ (pipe) เท่านั้นซึ่งเป็นวิธีที่สิ้นเปลืองมากและช้ากว่าการอ่านหรือกำหนดค่าตัวแปรอย่างมาก
การประมวลผลที่มีมากกว่าหนึ่งเทรดจะมีการสลับกันทำงานระหว่างเทรดโดยไม่ต้องเปลี่ยนสภาพแวดล้อมและ PCB ของการประมวลผลจึงเป็นภาระต่อเครื่องน้อยกว่า	การเปลี่ยนการทำงานของการประมวลผลหนึ่งไปสู่อีกการประมวลผลจะเปลี่ยนสภาพแวดล้อมของการประมวลผลด้วยจึงเป็นภาระต่อเครื่องมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 เทรดในภาษาจาวา

ภาษาซี (C) และซีพลัสพลัส (C++) สนับสนุนเทรดด้วยไลบรารี (library) ที่เขียนขึ้นต่างหากสำหรับแต่ละคอมไพเลอร์และระบบปฏิบัติการ ส่วนภาษาจาวาสนับสนุนการโปรแกรมมัลติเทรดดิ้งด้วยกลไกที่กำหนดขึ้นภายในภาษา โปรแกรมภาษาจาวาทุกๆ โปรแกรมจะต้องมีเทรดอย่างน้อยหนึ่งเทรดเสมอ คอมไพเลอร์จะเป็นผู้สร้างเทรดให้แก่ main() ของทุกๆ โปรแกรม เรียกว่าเทรดหลัก (main Thread) และจาวาอินเตอร์พรีเตอร์ จะเป็นผู้ควบคุมเทรดหลักที่หลังจากทำงานแล้วอาจจะมีการสร้างเทรดอื่นๆ ได้อีก

การสร้างเทรดใหม่ใช้ในโปรแกรม สามารถสร้างได้ 2 วิธี คือ

1) สร้างคลาสให้สืบทอด (extend) จากคลาสเทรด

2) สร้างคลาสโดยใช้อินเตอร์เฟสรันเอเบิล (Interface Runnable) วิธีนี้คลาสต่างๆ จะแปลงเป็นเทรดโดยไม่มีการเปลี่ยนแปลงภายในคลาส

คลาสเทรดมีดาตาเมมเบอร์(data members)สำหรับเก็บค่าคุณสมบัติของเทรดดังต่อไปนี้

ตารางที่ 2.2 แสดงคุณสมบัติของดาตาเมมเบอร์

data member	**คุณสมบัติ**
private char name [] ;	ชื่อของเทรด เราสามารถตั้งชื่อเทรดโดยส่งสตริงเป็นพารามิเตอร์ให้แก่ คอนสตรักเตอร์(constructor) หากไม่มีชื่อเทรดนั้นจะถูกตั้งชื่อให้โดยอัตโนมัติเป็น “Thread-1”+n โดย n เป็นเลขจำนวนเต็มที่เพิ่มขึ้นตามจำนวนเทรดที่ถูกสร้างขึ้น
private int priority ;	ค่าระดับความสำคัญของเทรดในการถูกระบบปฏิบัติการเลือกไปทำงาน
private boolean daemon = false;	ค่าที่ระบุว่าเทรดนั้นเป็นเดมอนเทรด (Daemon Thread) หรือไม่หากเทรดหนึ่งถูกสร้างขึ้นและถูกเริ่มการทำงานด้วย start() แล้ว จะถือว่ามีชีวิต ซึ่งเมื่อเรียก isAlive() จะเป็น true จนกว่าเมื่อสิ้นสุดการทำงานแล้ว isAlive() จะเป็น false

2.1.4 รายละเอียดและการทำงานของคลาสเทรด

คลาสเทรด (Class Thread) ถูกกำหนดในแพ็คเกจ(package) java.lang ใช้สำหรับสร้างอินสแตนซ์ (instance) ซึ่งเราจะเรียกสั้นๆว่า เทรด คลาสเทรดมีโปรแกรมสำหรับการทำงานของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทรดตัวหนึ่งที่มีการดำเนินของโปรแกรมแยกออกไปจากโปรแกรมที่สร้างมัน เทรดที่สร้างขึ้นจะยังไม่เริ่มต้นทำงานจนกว่า start()ของมันจะถูกเรียก เมื่อ start() ถูกเรียกแล้ว มันจะไปเข้าคิว จนเมื่อมันถูกเลือกไปทำงาน โปรแกรมใน run() ก็จะถูกทำงาน

คลาสเทรดมีเมธตอดสำหรับควบคุมเทรดให้เทรดเข้าสู่สถานต่างๆ เช่น start(), yield(), sleep(), suspend(), resume() และ stop() โดยปกติเราจะไม่โอเวอร์ไรด์ (override) เมธตอดเหล่านี้ นอกจากนี้ยังมี public void run() ซึ่งเป็นเมธตอดที่ว่างเปล่า ที่โดยปกติเราจะต้องโอเวอร์ไรด์เพื่อระบุว่าเมื่อเทรดนั้นอยู่ในสถานะ running จะทำอะไรบ้าง หากไม่ทำการโอเวอร์ไรด์ run() ก็จะได้เทรดที่ไม่ทำอะไรเลย

2.1.5 รายละเอียดและการทำงานของอินเตอร์เฟสรันเอเบิล

วิธีที่สองในการสร้างเทรดคือสร้างคลาสที่อิมพลีเมนต์รันเอเบิลอินเตอร์เฟส (implement Runnable interface) ในอินเตอร์เฟสนี้มีเพียง run() เป็นแอบสทแร็คเมธตอด (abstract method) คลาสที่อิมพลีเมนต์รันเอเบิลอินเตอร์เฟสจะต้องมี run() เพื่อระบุการทำงานของเทรดในสถานะ running

อินเตอร์เฟสรันเอเบิลถูกกำหนดไว้ใน java.lang ดังนี้

```
Public abstract interface Runnable {
Public abstract void run();
}
```

การใช้รันเอเบิลสร้างเทรดอาจทำได้สองวิธี

1) สร้างคลาสที่อิมพลีเมนต์รันเอเบิลโดยมีเทรดตัวหนึ่งเป็นสมาชิก

2)สร้างอินสแตนซ์ของคลาสที่อิมพลีเมนต์รันเอเบิลแล้วส่งให้เป็นพารามิเตอร์ของคอนสตรักเตอร์ของคลาสเทรด

ตัวอย่าง แสดงการสร้างคลาสที่อิมพลีเมนต์รันเอเบิลแล้วส่งให้เป็นพารามิเตอร์ของคอนสตรักเตอร์ของคลาสเทรด

```
// RunnableTest2.java
Class MyThread implements Runnable {
    public void run() {
        For (int i = 0 ; i < 100 ; i++)
            System.out.print(Thread.currentThread( ).getName( )) ;
        }
    }
Class RunnableTest2 {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        public static void main (String args[ ] {
                new Thread (new MyThread ( ) , “A” ) .start ( ) ;
                new Thread (new MyThread ( ) , “B”) .start ( ) ;
        }
}
```

คลาส MyThread ในตัวอย่างนี้อิมพลีเมนต์รันเอเบิล แต่ไม่มีการประกาศเทรดเป็นสมาชิกอยู่ภายใน ดังนั้นเมื่อจะเรียก getName() ต้องเรียก Thread.currentThread() ก่อนเพื่อได้รีเฟอร์เรนซ์ (reference) ของเทรดที่กำลังทำงานอยู่ ส่วนใน main() มีการสร้างอินสแตนซ์ของคลาสเทรดขึ้นโดยส่งผ่านไปในลักษณะเป็นพารามิเตอร์เท่านั้น โดยคอนสตรักเตอร์มีพารามิเตอร์ตัวแรกเป็นรันเอเบิลซึ่งมี run() ที่จะถูกทำงานและตัวที่สองเป็นสตริงที่จะเป็นชื่อของเทรดนั้น

2.1.6 เมธตอดที่สำคัญของเทรดในภาษาจาวา

เมธตอดของ java.lang.Thread

ตารางที่ 2.3 คุณสมบัติเมธตอดของเทรด

เมธตอด	คุณสมบัติ
currentThread()	คืนค่ารีเฟอร์เรนซ์ของเทรดที่กำลังเอ็กซีคิวต์ (execute)
destroy()	ทำลายเทรดโดยไม่คลีนอัพ (clean up)
getName()	คืนชื่อของเทรด
getPriority()	คืนค่าความสำคัญของเทรด
getThreadGroup()	คืนค่า ThreadGroup
isAlive()	เทรดอยู่หรือไม่
isDaemon()	เทรดเป็นเดมอนเทรดหรือไม่
join(long millis)	คอยให้เทรดตาย
resume()	เรียกเทรดที่ถูก suspend() ให้กลับมาทำงาน
run()	จะถูกเรียกทำงานเมื่อเทรดเข้าสู่สถานะ ready
setDaemon()	กำหนดให้เทรดเป็นเดมอนเทรด
setName(String name)	กำหนดชื่อเทรด
setPriority()	กำหนดความสำคัญของเทรด
sleep(long millis)	สั่งให้เทรดหลับ
start()	สั่งให้เทรดเริ่มต้นทำงาน JVM จะเรียก run() มาทำงานให้เอง
stop()	สั่งให้เทรดหยุดทำงาน อาจจะก่อให้เกิดเอกเซฟชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 (ต่อ)

suspend()	สั่งให้เทรดหยุดการทำงานชั่วคราว
yield()	สั่งให้เทรดเปลี่ยนสถานะจาก running เป็น ready

เมธตอดที่ inherit จากคลาส java.lang.Object

ตารางที่ 2.4 คุณสมบัติของเมธตอดที่ inherit จากคลาส java.lang.Object

wait()	สั่งให้เทรดเข้าสู่สถานะบล็อก(blocked state)
notify()	ใช้เรียกให้เทรดอื่นที่ถูก wait() กลับสู่สถานะ ready
notifyAll()	สั่งให้ทุกเทรดที่ถูก wait มีโอกาสถูกเลือกเข้าทำงานเท่าๆกัน

2.1.7 **วิธีการใช้ฟังก์ชันเดมอนเทรด**

มีหลายกรณี ที่เทรดหนึ่งต้องสร้างเทรดลูกขึ้นมาเพื่อคอยสนับสนุนการทำงานของมันโดย ปกติเทรดลูกพวกนี้จะถูกสร้างขึ้นมาในระหว่างเทรดแม่ทำงาน และจะทำงานร่วมกับเทรดแม่ไปจน เมื่อเทรดแม่จะสิ้นสุดการทำงานก็สั่ง stop() เทรดเหล่านั้น แต่ในบางกรณีอาจยุ่งยากที่จะเก็บ รี เฟอร์เรนซ์ของเทรดลูกทั้งหมดที่สร้างขึ้น และการตรวจสอบว่าเทรดใดตายไปแล้วหรือยังหรือจะมี ปัญหาถ้าเทรดแม่จบการทำงานอย่างไม่ปกติโดยไม่มีโอกาสได้เรียก stop() ของเทรดลูกที่มันสร้าง ขึ้น เราเรียกเทรดลูกที่ยังทำงานอยู่ แต่เทรดแม่ตายของมันไปแล้วว่า orphan Threads ซึ่งเทรด แบบนี้มักจะใช้ทรัพยากรเครื่องอย่างไม่ก่อให้เกิดประโยชน์

จาวาได้ขอยืมเครื่องหมายต่างๆ ของเดมอนเทรดจากยูนิกซ์ เดมอนเทรดเป็นเทรดที่มีการ ประมวลผลในแบ็คกราวน์และคอยให้บริการเทรดปกติ และทำการประมวลผลไปไปเรื่อยๆ จนกว่า จะมีการร้องขอให้บริการ และจะรอคอยที่จะให้บริการต่อไป จนกระทั่งไม่มีเทรดมารับบริการ เมื่อ งานทั้งหมดได้สิ้นสุดลงแล้วเดมอนเทรดก็จะตาย โดยปกติ หากยังมีเทรดใดเทรดหนึ่งยังทำงาน อยู่ JVM ก็จะปล่อยให้ทำงานต่อไปเรื่อยแต่ ภาษาจาวาเสนอเทรดพิเศษขึ้นเรียกว่าเดมอนเทรด หาก JVM พบว่าเทรดที่ยังทำงานอยู่นั้นมีแต่ เดมอนเทรดเท่านั้น ไม่มีเทรดธรรมดาทำงานอยู่เลย มันก็จะหยุดตัวเองลง ส่งผลให้เดมอนเทรดเหล่านั้นตายโดยอัตโนมัติ ดังนั้นหากเทรดแม่ตัวหนึ่งมี เทรดลูกที่ทำงานสนับสนุนเป็นเดมอนเทรดทั้งหมด สมมติว่ามีเพียงเทรดแม่ตัวเดียวที่เป็นเทรด ปกติเมื่อ เทรดแม่ตายเทรดลูกทั้งหมดก็จะตายลงเองด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.8 วิธีการใช้ฟังก์ชันเรซิ่งคอนดิชัน

หากมีเทรดตั้งแต่สองตัวขึ้นไป แย่งกันใช้ทรัพยากรหนึ่งอย่างไม่ ถูกควบคุมก็อาจจะเกิดปัญหาขึ้นได้ เนื่องจากลำดับการใช้งานทรัพยากรนั้นของแต่ละเทรดอาจมีความสำคัญต่อผลลัพธ์ของโปรแกรมในบางกรณีอาจทำให้ โปรแกรมเดียวกันให้ผลลัพธ์แตกต่างกันในการทำงานแต่ละครั้งปรากฏการณ์นี้เรียกว่าเรซิ่งคอนดิชัน (Racing Condition) ของเทรดในโปรแกรมนั้น

2.1.9 วิธีการใช้ฟังก์ชันซิงโครไนเซชัน

ภาษาจาวามีกลไกสำหรับการควบคุมให้ที่เวลาหนึ่ง คลาสของอินสแตนซ์หรือทรัพยากรนั้นถูกใช้งานได้โดยเทรดเดียวเท่านั้น โดยนำกลไกมูทัวลี่เอกซ์คลูซีฟ (mutually exclusive) ของระบบปฏิบัติการมาใช้ครอบทรัพยากรไว้นั่นคือ หากคลาสนั้นมีเมธตอดหนึ่งที่ถูกระบุด้วย คีย์เวิร์ด (keyword) “synchronized” เมื่อเมธตอดนั้นถูกเรียกจากเทรดใดๆก็ตาม จะจำกัดให้มีเพียงเทรดเดียวเท่านั้นที่สามารถเข้าครอบครองและใช้งาน ระหว่างที่ทรัพยากรนั้นถูกเทรดที่ใช้ฟังก์ชันซิงโครไนเซชัน (Synchronization) ครอบครองอยู่ หากมีเทรดอื่นต้องการใช้งานทรัพยากรนั้นก็จะหยุดรอจนกว่าเทรดที่ใช้งานอยู่จะยอมปล่อยทรัพยากรนั้นหรือประมวลผลเสร็จ แต่หากในระหว่างนั้นก็มีเทรดอื่นที่กำลังทำงานเมธตอดนั้นอยู่ เทรดที่พยายามเรียกก็จะต้องรอจนกว่าเทรดที่ทำงานอยู่จะสิ้นสุดการทำงานในเมธตอดนั้น

สังเกตว่า เฉพาะสมาชิกที่เป็นเมธตอดเท่านั้น จึงจะมีคีย์เวิร์ด “synchronized” ได้ ไม่อาจใช้กับสมาชิกที่เป็นดาตาเมมเบอร์

2.1.10 วิธีการใช้ฟังก์ชันเดดล็อค

หากมีเทรดหนึ่งต้องรอให้อีกเทรดหนึ่งทำงานสิ้นสุดลงเสียก่อนจึงจะทำงานต่อไปได้ เราเรียกว่าเทรดนั้นขึ้นกับ (depend on) อีกเทรดหนึ่ง โปรแกรมมัลติเทรดดิ้งที่มีการขึ้นต่อกันระหว่างเทรดอาจมีปัญหาเกิดขึ้นในระหว่างทำงานได้ เช่นหากเทรดหนึ่งได้ครอบครองทรัพยากรที่เป็นมูทัวลี่เอกซ์คลูซีฟ และถ้าเทรดนั้นไม่สามารถจบการทำงานลงได้เทรดอื่นๆที่รอใช้ทรัพยากรนั้นก็จะไม่สามารถทำงานได้ ปัญหาแบบนี้ถือเป็นความผิดพลาดของการสร้างโปรแกรมที่พอตรวจสอบและแก้ไขได้ แต่ก็มีปัญหาอีกแบบหนึ่งที่ตรวจสอบและแก้ไขได้ยากกว่าคือ ปัญหาเดดล็อค (Deadlock) ซึ่งเกิดขึ้นกับเทรดมากกว่าหนึ่งตัวขึ้นไปที่ไม่สามารถทำงานต่อไปได้ โดยจะอยู่ในสถานะบล็อกไปอย่างถาวร

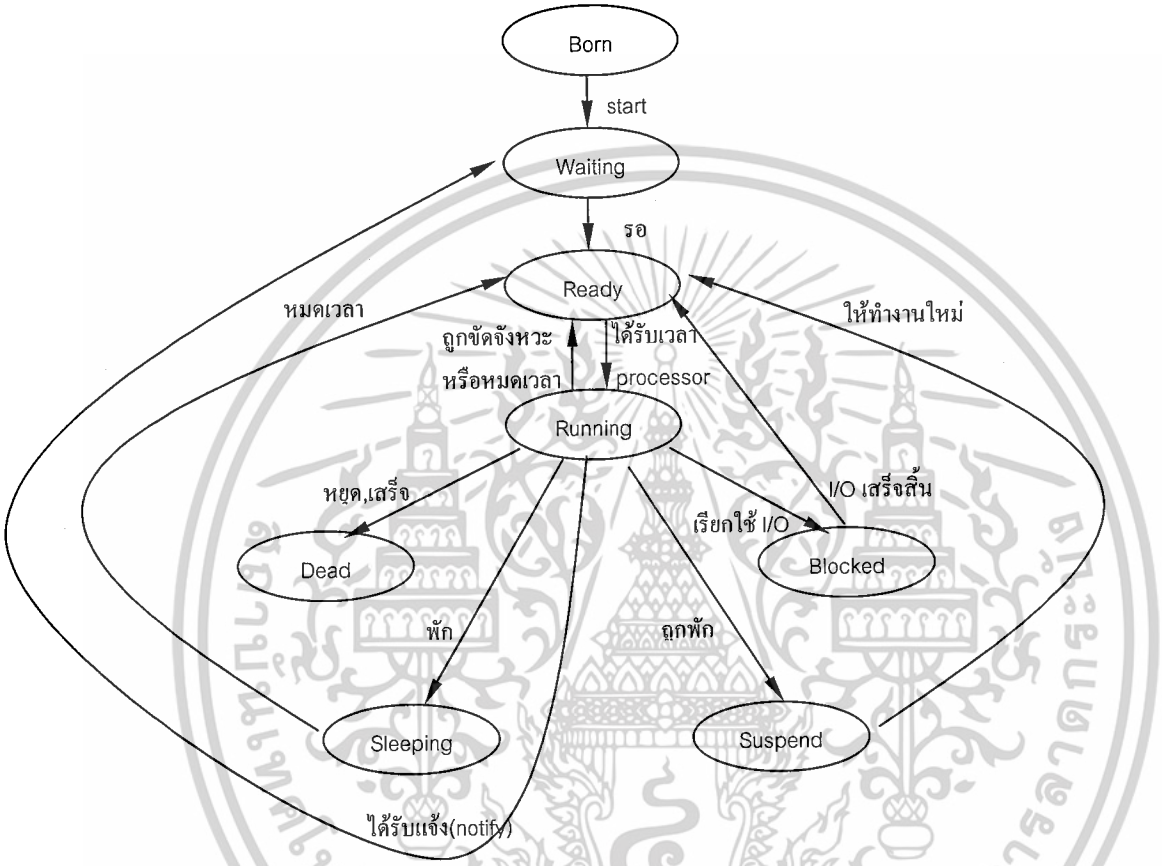
เดดล็อคจะเกิดขึ้นได้เมื่อเทรดอย่างน้อยสองเทรดสมมติว่าเป็น Aและ B โดยที่ เทรด Aได้ครอบครองทรัพยากร X ที่เป็นมูทัวลี่เอกซ์คลูซีฟและ เทรด B ได้ครอบครองทรัพยากร Y ที่เป็นมูทัวลี่เอกซ์คลูซีฟ เช่นกัน แต่ใจขณะนั้น เทรด A ต้องการใช้ทรัพยากร Y และ เทรด B ก็ต้องการใช้ทรัพยากร X ทำให้เทรดทั้งสองต้องเข้าสู่สถานะบล็อกเพื่อรอให้ทรัพยากรที่รอใช้งานถูกปล่อยออกมาจึงจะทำงานต่อไปได้ แต่ทั้งเทรด A และ B จะไม่ปล่อยทรัพยากรที่ตัวครอบครองอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากยังทำงานไม่เสร็จ นั่นคือเทรดทั้งสองกำลังรอคอยสิ่งที่จะไม่เกิดขึ้นจึงต้องอยู่ในสถานะบล็อกตลอดไป

2.1.11 วัฏจักรของเทรด

วัฏจักร (lifetime) ของเทรดแสดงได้เป็นรูปดังรูปนี้



รูปที่ 2.1 แสดงวัฏจักรของเทรด

ในเวลาต่างๆเทรดจะอยู่ในสภาวะต่างๆ (Thread States) เริ่มจากสภาวะเริ่มต้น (Born State) เทรดจะอยู่ในสภาวะดังกล่าวจนกว่าเมธตอด Start ของเทรดดังกล่าวจะถูกเรียก เมื่อเรียกเมธตอด Start แล้วเทรดจะอยู่ในสภาวะพร้อม (Ready State) เทรดซึ่งมีความสำคัญสูงสุดจะเข้าไปอยู่ในสภาวะทำงาน (Running State) เมื่อได้รับเวลาจากหน่วยประมวลผล (Processor) เทรดจะหมดสภาพหรือตาย (Dead State) เมื่อมันทำงานเสร็จสมบูรณ์ หรือถูกเรียกด้วยเมธตอด stop เทรดที่ตายแล้วจะถูกกำจัดโดยระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บางครั้งเทรดอาจอยู่ในสภาวะบล็อกเมื่อเทรดมีการเรียกใช้ Input/Output Block เทรดจะกลายสภาพเป็นสภาวะพร้อมเมื่อการทำงานของ I/O ที่รออยู่สิ้นสุดลงเทรดที่อยู่ในสภาวะบล็อกไม่สามารถใช้เวลาของหน่วยประมวลผล ถึงแม้หน่วยประมวลผลจะว่างอยู่ก็ตาม

เทรดจะอยู่ในสภาวะพักผ่อนเมื่อถูกเรียกด้วยเมธตอด sleep เทรดที่พักผ่อนจะกลับมาอยู่ในสภาวะพร้อมอีกครั้งเมื่อหมดเวลาพักผ่อนเทรดที่อยู่ในสภาวะพักผ่อนจะไม่สามารถใช้เวลาของหน่วยประมวลผลถึงแม้หน่วยประมวลผลจะว่างอยู่ก็ตาม

เมื่อเทรดที่ทำงานอยู่ถูกเรียกด้วยเมธตอด suspend จะกลายสภาพเป็นถูกพัก (Suspend State) เทรดที่ถูกพักจะกลายสภาพมาเป็นสภาวะพร้อมอีกครั้ง เมื่อถูกเรียกด้วยเมธตอด notify จากเทรดอื่นที่ทำงานกับวัตถุนั้นเทรดทุกอันจะเปลี่ยนสภาพเป็นพร้อมทำการประมวลผลตามปกติ เมื่อถูกเรียกด้วยเมธตอด notifyAll

2.1.12 **ลำดับความสำคัญของเทรดและการจัดลำดับงาน**

ทุกจาวาแอปพลิเคชั่นและจาวาแอปเพล็ตเป็นมัลติเทรด เทรดต่างๆ ทั้งหมดอาจกำหนดให้มีค่าความสำคัญ (Priority) เท่ากัน ซึ่งถ้าไม่มีการกำหนดค่าให้จาวาจะกำหนดให้มีความสำคัญเป็น Thread.NORM_PRIORITY แต่บางครั้งจำเป็นต้องสร้างบางเทรดให้สำคัญกว่าเทรดอื่น ทุกเทรดในจาวามีค่าความสำคัญอยู่ระหว่างช่วง Thread.MIN_PRIORITY และ Thread.MAX_PRIORITY

จาวาในบางระบบปฏิบัติการใช้แนวคิดไทม์สไลซิงค์ (Timeslicing) แต่บางระบบก็ไม่ใช้ กรณีที่ไม่ใช้ไทม์สไลซิงค์แต่ละเทรดในชุดของเทรดที่มีลำดับความสำคัญ (Schedulling) เท่ากันจะทำงานจนเสร็จสมบูรณ์เทรดอื่นจึงจะมีโอกาสทำงาน แต่ในไทม์สไลซิงค์แต่ละเทรดจะได้รับเวลาควอนตัม (TimeQuantum) จำนวนหนึ่งเมื่อใช้เวลาดังกล่าวหมดแม้งานจะยังไม่เสร็จหน่วยประมวลผลจะถูกเรียกกลับไปทำงานให้กับเทรดอื่นซึ่งมีความสำคัญเท่ากันต่อไป

งานของตัวจัดลำดับงานของจาวา ก็คือ ให้เทรดที่มีความสำคัญทำงานก่อน และถ้าเทรดที่มีความสำคัญเท่ากันมีหลายตัว ก็จัดให้เทรดเหล่านั้นทำงานในลักษณะวนรอบ

ลำดับความสำคัญของเทรดสามารถเปลี่ยนแปลงได้โดยใช้เมธตอด setPriority ซึ่งรับค่าจำนวนเต็มที่บอกถึงลำดับความสำคัญของเทรด ค่าจำนวนเต็มดังกล่าวต้องอยู่ในช่วง 1-10 ถ้าค่าดังกล่าวไม่อยู่ในช่วงเมธตอด setPriority จะ Throw IllegalArgumentException เมธตอด getPriority ส่งคืนค่าจำนวนเต็มแทนความสำคัญของเทรดนั้นๆ

เทรดสามารถเรียกเมธตอด yield เพื่อให้โอกาสเทรดอื่นได้ทำงาน ปกติเมื่อเทรดที่มีความสำคัญสูงกว่าพร้อมจะแย่งเวลาหน่วยประมวลผลไปจากเทรดปัจจุบันเมธตอด yield ให้โอกาสแก่ เทรดที่มีความสำคัญเท่ากันในการทำงาน ในกรณีที่เป็นระบบไทม์สไลซิงค์เราไม่จำเป็นต้องใช้ เมธตอด yield เนื่องจากเทรดแต่ละเทรดมีความสำคัญเท่ากันเวียนกันทำงานด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาควอนตัมของตนเอง

เทรดจะทำงานจนกระทั่งมันตาย ถูกบล็อกรอ I/O sleep, wait, yield หรือถูกแย่งจากเทรดที่มีความสำคัญสูงกว่า

ตารางที่ 2.5 ตารางแสดงข้อมูลของเทรดที่จะเข้าประมวลผลในหน่วยประมวลผลกลาง

เวลาการมาถึงของเทรด	ชื่อเทรด	ค่าระดับความสำคัญ	เวลาในการประมวลผล
0	A	5	2
1	B	5	5
4	C	7	3
5	D	10	1
7	E	5	3

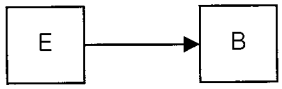
จากตารางที่ 2.5 สามารถนำข้อมูลมาเขียนแสดงการจัดการเทรดเข้าทำงานตามระดับความสำคัญของหน่วยประมวลผลกลางได้ดังตารางที่ 2.6

ตารางที่ 2.6 แสดงการจัดการเข้าประมวลผลของเทรดตามลำดับความสำคัญ

เวลา	แถวคอยพร้อม(Ready Queue)	เทรดที่เข้าประมวลผล
0		A
1	B	A
2	B	A
3		B
4	B	C
5	B → C	D
6	B	C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 (ต่อ)

7		

2.2 คอมโพเนนต์

ในระบบคอมพิวเตอร์ซึ่งประกอบไปด้วยซอฟต์แวร์ (software) ฮาร์ดแวร์ (hardware) ที่เชื่อมต่อกันจะมีการออกแบบในส่วนประกอบเหล่านั้น เพื่อเป็นการกำหนดรายละเอียดเชิงเทคนิคของระบบ เช่น แต่ละโปรแกรมประกอบด้วยไฟล์อะไรบ้าง ซึ่งเป็นการมองระบบตามความเป็นจริง

คอมโพเนนต์ (component) คือ ซอฟต์แวร์ต่างๆ ที่ทำงานภายในระบบ เช่น ไฟล์ซอร์สโค้ด (source code) ไฟล์ไบนารี (binary code) และไฟล์เอ็กซีคิวต์ (executable code) ในภาษาจาวา แต่ละคอมโพเนนต์อาจเปรียบได้กับหนึ่งจาวาไฟล์ (.java) โดยอาจบรรจุมากกว่าหนึ่งคลาส ถ้าเป็นระบบงานธุรกิจก็อาจเป็นองค์ประกอบหรือเอกสารต่างๆ ที่ใช้ในการประกอบธุรกิจ โดยการออกแบบคอมโพเนนต์สามารถนำไปประยุกต์ใช้กับการเขียนโปรแกรมภาษาหรือการทำงานบนแพลตฟอร์ม (platform) ใดก็ได้

การออกแบบคอมโพเนนต์นั้นไม่ได้มีหลักการตายตัว แต่อย่างไรก็ตามสามารถยึดหลักการต่อไปนี้ได้

1) พิจารณาว่าผู้ใช้งานมีกี่กลุ่ม แต่ละกลุ่มมีลักษณะการใช้งานเหมือนหรือแตกต่างกันอย่างไร จะทำให้ได้คอมโพเนนต์ที่ควรจะมีในพรีเซนเทชั่นลอจิกซับซิสเทม (Presentation Logic Subsystem) เช่น หน้าจอต่างๆ และอาจจะได้บางส่วนของคอมโพเนนต์ในเวอร์คกิ้งลอจิกซับซิสเทม (Working Logic Subsystem)

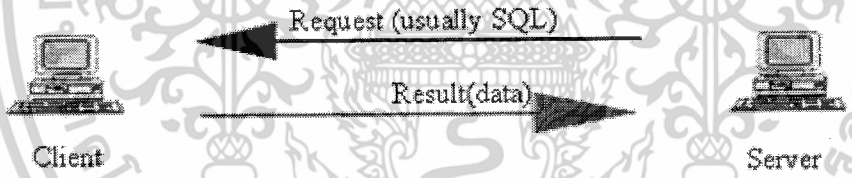
2) มีข้อมูลใดหรือไม่ที่จำเป็นต้องบันทึกเพื่อใช้ได้ในเวลานาน ผู้ใช้งานแต่ละกลุ่มใช้ข้อมูลใด และมีข้อมูลส่วนใดที่มีผู้ใช้งานหลายกลุ่มใช้ร่วมกัน การพิจารณาในส่วนนี้จะช่วยในการหาและแบ่งแยกคอมโพเนนต์ในดาตาเบสลอจิกซับซิสเทม (Database Logic Subsystem) นั่นคือช่วยให้พิจารณาได้ว่าจะใช้ฐานข้อมูลเป็นแบบศูนย์กลางหรือแบบกระจาย หรือควรจะมีเทเบิล (table) ใดบ้างในฐานข้อมูลหนึ่งๆ

การพิจารณาว่าในระบบมีคอมโพเนนต์ใดบ้างนั้นเป็นหน้าที่โดยตรงของผู้พัฒนาในอันที่จะสัมภาษณ์และสอบถามผู้ใช้งานถึงความต้องการ ลักษณะการใช้งานรวมถึงฮาร์ดแวร์แพลตฟอร์มที่ผู้ใช้งานต้องการด้วย โดยการสัมภาษณ์นั้นจะแยกทีละซับซิสเทม (Subsystem) หรือรวมกันเลยก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 สถาปัตยกรรม 2-Tier

ส่วนประกอบ 3 ส่วน ของโปรแกรมอันได้แก่ ส่วนแสดงผลส่วนประมวลผลและส่วนของข้อมูล ซึ่งสามารถแบ่งออกได้เป็น 2 จำพวก คือ ส่วนของรหัสของไคล์เอนท์และส่วนของฐานข้อมูลของเซิร์ฟเวอร์ โปรแกรมของไคล์เอนท์ที่ได้รับการพัฒนาแล้วนั้น จะต้องเป็นตัวหลักที่ทำงานได้อย่างคล่องแคล่วในการการส่งการร้องขอของไคล์เอนท์ ไปยังเครื่องเซิร์ฟเวอร์ซึ่งเป็นสิ่งสำคัญมากสำหรับการวางโครงสร้างแบบ 2-Tier ส่วนของการแสดงผลถูกควบคุมโดยไคล์เอนท์ ส่วนการประมวลผลนั้นจะแบ่งระหว่างตัวไคล์เอนท์และตัวเซิร์ฟเวอร์และส่วนของข้อมูลนั้นจะถูกเก็บ และจัดการผ่านตัวเซิร์ฟเวอร์ ในการร้องขอข้อมูลจะอยู่ในรูปฟอร์มของSQL โดยการส่ง SQLจากไคล์เอนท์ถึงเซิร์ฟเวอร์ต้องการการเชื่อมต่อที่ค่อนข้างแน่นอนระหว่างทั้งสองชั้นไคล์เอนท์จะต้องรู้ถึงวากยสัมพันธ์ของเซิร์ฟเวอร์หรือมีการแปลงรูปแบบผ่าน API (Application Program Interface) และมันจะต้องรู้ว่าเซิร์ฟเวอร์ว่าตั้งอยู่ที่ใดและข้อมูลจะถูกจัดการอย่างไรและข้อมูลถูกกำหนดอย่างไรการร้องขอจะถูกเก็บและประมวลผลบนเครื่องเซิร์ฟเวอร์ซึ่งเป็นศูนย์รวมของงานทั้งหมด เช่นการเช็คความถูกต้องของข้อมูลการรวบรวมข้อมูลและการรักษาความปลอดภัยของข้อมูล ข้อมูลจะถูกส่งคืนไปยังตัวไคล์เอนท์ และถูกจัดการในระดับของไคล์เอนท์ แล้วแสดงผลออกมา ดังรูปที่ 2.2



รูปที่ 2.2 แสดงการรูปแบบเข้าถึงข้อมูลของสถาปัตยกรรม 2-Tier

2.3.1 **ข้อดีของสถาปัตยกรรม** 2-Tier

1. โปรแกรมที่พัฒนาขึ้นมานั้นเร็วมาก เพราะระบบ two tier สามารถพัฒนาเป็นส่วนย่อยเล็กๆ ได้ หลักจากนั้นมันจะนำรหัสมาเปรียบเทียบกัน แต่เป็นระบบที่ค่อนข้างจะยืดหยุ่นน้อย

2. เครื่องมือของ two tier สามารถใช้ลักษณะโครงสร้างข้อมูล รวมทั้งการสร้างในโพรซิเยอร์ และฟังก์ชั่นได้หลายรูปแบบ อีกทั้งยังป้องกัน สิ่งที่เกิดขึ้นจากการโปรแกรม เช่นการจัดการหน่วยความจำ เครื่องมือเหล่านี้จะใช้เทคนิคการเรียกซ้ำ และใช้เทคนิค rapid application develpment(RAD) ซึ่งทำให้แน่ใจได้ว่า ความต้องการของผู้ใช้จะสามารถทำได้อย่างรวดเร็ว และสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. IS organizations สามารถติดต่อส่วนที่เหลืออยู่ โดยใช้คำสั่งของผู้ใช้ ผ่านเครื่องมือสำหรับการพัฒนาระบบ two tier ไคล์เอนท์/เซิร์ฟเวอร์ ซึ่งพัฒนาได้อย่างรวดเร็ว และแปลงรูปแบบได้มาก

4. โครงสร้างแบบ two tier สามารถทำงานได้ดีในสภาวะแวดล้อมที่ไม่มีการเปลี่ยนแปลงโครงสร้างนี้ จะไม่เหมาะสมกับ สภาวะแวดล้อมแบบกระจาย หรือแตกต่างกับกฎที่เปลี่ยนแปลงไปอย่างรวดเร็ว เพราะว่าโปรแกรมขนาดใหญ่ที่มีอยู่บนเครื่องไคล์เอนท์ PC ทำให้โครงสร้างแบบ two tier จึงเจอกับปัญหาของการควบคุม และ ปัญหาของการ re-distribution การเปลี่ยนกฎเกณฑ์ จะต้องเปลี่ยนที่ตัวไคล์เอนท์ แต่ละตัว แต่ละโปรแกรมประยุกต์ การเปลี่ยนแปลงไคล์เอนท์ผ่านเครือข่ายเป็นงานที่ยาก เนื่องจากขาดการควบคุมการปรับปรุงรุ่นของโปรแกรมในเครื่อง ดังนั้น การปรับเปลี่ยน หรือปรับปรุงโปรแกรมจึงต้องทำกับไคล์เอนท์ทุกเครื่อง

5. ระบบรักษาความปลอดภัยในระบบ two tier มีความซับซ้อนมาก เพราะว่า ผู้ใช้แต่ละคนต้องใช้รหัสผ่านที่แตกต่างกัน สำหรับการประมวล SQL แต่ละครั้ง โปรแกรมไคล์เอนท์/เซิร์ฟเวอร์ที่ได้รับการพัฒนา ส่วนมากจะออกแบบโดยปราศจากการตบตา ซึ่งทำให้เพิ่มประสิทธิภาพความปลอดภัยมากขึ้น แต่โอกาสที่รหัสผ่านจะซ้ำกัน ซึ่งทำให้ผู้ใช้ที่ไม่มีในการประมวลข้อมูล หรือตารางที่ซ่อนไว้ เข้าไปใช้ข้อมูล ทำให้ข้อมูลอาจเปลี่ยนแปลง หรือสูญหายไปได้

2.4 หลักในการสร้างเกม

2.4.1 ทฤษฎีเกม

ทฤษฎีเกม เป็นเครื่องมือชนิดหนึ่งที่อธิบายและกล่าวถึงปัญหาทางสังคม เกมทั่วๆไปมักจะสะท้อนหรือบอกให้ทราบถึงลักษณะที่เกิดขึ้นในสถานการณ์จริง โดยเฉพาะอย่างยิ่งในสถานการณ์การแข่งขันหรือการร่วมมือกัน เกมเหล่านี้จะแนะนำกลยุทธ์ที่ใช้ในการแก้ไขสถานการณ์ที่เกิดขึ้นซึ่งเมื่อทำความเข้าใจกลยุทธ์ของผู้เล่นเกมในแต่ละเกม อาจสามารถทำนายได้ว่าผู้เล่นเหล่านั้นจะวางตัวอย่างไรในสถานการณ์ที่กำหนดให้

โดยทั่วไปขณะที่คนมุ่งจะชนะในเกมการแข่งขันมักจะพยายามชนะหรือบรรลุผลประโยชน์หรือเป้าหมายในการแข่งขัน อย่างไรก็ตามทั้งในการเล่นเกมและในความเป็นจริงจำเป็นต้องปฏิบัติตามกฎกติกาที่วางไว้เพื่อให้บรรลุสิ่งเหล่านี้ ในบางเกมก็คล้ายกับสถานการณ์จริง คือผู้ชนะจะได้ทุกอย่างที่ต้องการ โดยปกติเกมเหล่านี้จะมีลักษณะของการแข่งขันสูง คือ มีผู้ชนะเพียงคนเดียว เช่นเกมหมากรุก เกมอื่นๆ อาจต้องใช้การร่วมมือกันเพื่อให้ได้มาซึ่งชัยชนะ มีเกมจำนวนมากที่ต้องการกลยุทธ์ความร่วมมือจากผู้เล่นหลายคนเพื่อให้ผู้เล่นคนใดคนหนึ่งได้รับชัยชนะ ในโลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แห่งความจริงแม้ในช่วงของการต่อสู้ ส่วนมากคู่ต่อสู้จะมีผลประโยชน์ร่วมกันและต้องร่วมมือกันในระดับหนึ่ง

2.4.1.1 ความหมายของทฤษฎีเกม

ทฤษฎีเกมให้เครื่องมือที่ช่วยวิเคราะห์ตรวจสอบกลยุทธ์ที่เกิดจากความสัมพันธ์ของผู้เล่นเกมสองฝ่ายหรือมากกว่าสองฝ่าย โดยการใช้สถานการณ์จำลองทางคณิตศาสตร์แบบง่ายๆ ในการศึกษาความเกี่ยวข้องทางสังคมที่ยุ่งยากซับซ้อนทฤษฎีเกมจะอธิบายให้ทราบถึงศักยภาพ และความเสี่ยงที่ควบคู่มากับพฤติกรรมที่ต้องร่วมมือกันในระหว่างคู่แข่งขันที่ไม่ไว้วางใจกันและกัน แม้ว่าผู้เล่นจะไม่คุ้นเคยเหมือนเกมที่เล่นบนแผ่นกระดานแบบอื่นๆ หรือวีดีโอเกม แต่บทเรียนที่ได้รับจากทฤษฎีเกมมีลักษณะเป็นนามธรรม หรือเป็นสมมติฐานมากกว่าสามารถนำไปใช้ในการแก้ไขสถานการณ์ทางสังคมได้อย่างกว้างขวางมากกว่า

เกมที่เล่นจำลองเหตุการณ์ในชีวิตจริง มีองค์ประกอบ 5 อย่าง ได้แก่

1. ผู้เล่นหรือผู้ตัดสินใจ
2. กลยุทธ์ของผู้เล่นแต่ละคน
3. กฎกติกาที่ใช้ในการควบคุมการตัดสินใจและแสดงพฤติกรรมของผู้เล่นแต่ละคน
4. ผลคะแนน คะแนนที่ผู้เล่นแต่ละคนได้รับ เนื่องจากตัดสินใจเลือกทางเดิน
5. การมีส่วนได้ส่วนเสียที่สะสมขึ้นเรื่อยๆของผู้เล่นแต่ละคน เนื่องมาจากคะแนนสะสม

เกมมักจะนี้ตั้งสมมติฐานไว้ว่า ผู้เล่นแต่ละคนจะใช้กลยุทธ์ที่ช่วยให้เขาประสบความสำเร็จได้คะแนนมากที่สุดในทุกๆสถานการณ์ ในชีวิตจริงที่เต็มไปด้วยสถานการณ์ที่เกิดขึ้นโดยตั้งใจหรือไม่ตั้งใจก็ดี ผู้คนต่างแสวงหาผลประโยชน์โดยให้คนอื่นเป็นฝ่ายเสียเปรียบ นี่เป็นการกระทำที่นำไปสู่ความขัดแย้งหรือแข่งขันกัน มีการใช้เกมอธิบายความสัมพันธ์เหล่านี้ โดยการกำหนดผลประโยชน์ของผู้เล่นทั้งสองฝ่ายให้ขัดแย้งกัน เมื่อผู้เล่นคนหนึ่งเสียผลประโยชน์มาก ผู้เล่นอีกคนหนึ่งก็เสียผลประโยชน์น้อย เพื่อจะบรรลุผลประโยชน์ร่วมกัน ผู้เล่นหลายคนจำเป็นต้องใช้กลยุทธ์ร่วมมือกัน เพราะว่าถ้าหากผู้เล่นแต่ละคนต่างเสี่ยงทุ่มเทให้กับการได้และเสียมากที่สุด คะแนนที่ได้รับจะไม่มาก อย่างไรก็ตาม เกมเหล่านี้จะเน้นให้เห็นถึงความยากลำบากของการได้รับความร่วมมือจากคู่แข่งขันที่ไม่น่าไว้วางใจ เพราะว่าผู้เล่นแต่ละคนต่างก็อยากจะได้ผลประโยชน์มากที่สุด การร่วมมือกันต้องการให้ผู้เล่นทั้งสองฝ่ายประนีประนอมกัน ห้ามใจตนเองมิให้มุ่งหวังจะได้รับประโยชน์สูงสุด และในการประนีประนอมกันผู้เล่นแต่ละคนต้องเสี่ยงกับการสูญเสีย ถ้าหากคู่ต่อสู้ตัดสินใจอยากได้ผลประโยชน์สูงสุด ผู้เล่นมีแนวโน้มยอมเสียน้อยมากกว่าสูญเสียจนหมด

2.4.1.2 ประโยชน์ของการมีทฤษฎีเกม

สถานการณ์จำลองเหล่านี้จะทำให้ผู้เล่นสามารถหยั่งรู้ในกลยุทธ์ที่เป็นผู้เล่นเกมคนอื่นใช้เป็นทางเลือกและคะแนนที่จะได้รับจากสถานการณ์นั้นๆ การหยั่งรู้นี้ ผู้ตัดสินใจจะสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเมินผลที่จะเกิดขึ้นจากการกระทำของตนเองได้ดี และสามารถตัดสินใจเพื่อบรรลุเป้าหมายที่ต้องการโดยหลีกเลี่ยงความขัดแย้ง

ยกตัวอย่างทฤษฎีการยับยั้งฝ่ายตรงกันข้าม ทำให้เกิดยุทธศาสตร์การป้องกันตัวของสหรัฐอเมริกา ตั้งแต่สิ้นสงครามโลกครั้งที่ 2 มีสมมติฐานว่าการตอบโต้อย่างรุนแรง แบบตาต่อตาฟันต่อฟันสามารถป้องกันยับยั้งพฤติกรรมก้าวร้าวของผู้บุกรุกได้ ถ้าหากปัจเจกชนเชื่อว่าพฤติกรรมก้าวร้าวรุนแรงอาจก่อให้เกิดการตอบโต้อย่างรุนแรงจากผู้อื่น ปัจเจกชนผู้นั้นย่อมจะไม่ประพฤติก้าวร้าวรุนแรงต่อผู้อื่น การข่มขู่ว่าจะตอบโต้โจมตีไม่สามารถลดความรุนแรงที่จะเกิดขึ้นโดยตรง แต่การรับรู้ถึงผลลัพธ์ที่จะเกิดขึ้นเนื่องจากความก้าวร้าวจะช่วยลดความรุนแรงในสถานการณ์ที่อาจขึ้นเกิดจากการโต้ตอบโจมตีกันลงมา หากปัจเจกชนทั้ง 2 ฝ่ายยอมรับว่าผลประโยชน์สูงสุดที่จะได้รับเกิดจากการหลีกเลี่ยงไม่ใช้ความรุนแรงโต้ตอบกัน ไม่มีการอาฆาตมาดร้ายกันและกัน นี่คือแนวทางหลักเบื้องหลังสงครามเย็นระหว่างสหรัฐอเมริกากับรัสเซีย

แนวความคิดเกี่ยวกับผลประโยชน์รวมที่เกิดจากการยับยั้งฝ่ายตรงกันข้ามทำให้เกิดมาตรการควบคุมอาวุธและการร่วมมือกัน การเน้นความสนใจเรื่องยุทธศาสตร์ทางเลือกและผลตอบแทนที่จะได้รับ ทฤษฎีเกมอธิบายให้เราเห็นภาพวาความสัมพันธ์ที่มีศักยภาพในการทำลายล้างเราสามารถควบคุมจัดการและเปลี่ยนแปลงให้เกิดผลประโยชน์ร่วมกันได้ รวมทั้งหลีกเลี่ยงการแข่งขันการสร้างอาวุธ และการทำสงครามนิวเคลียร์

2.4.2 **โครงสร้างของเกม**

ปัญหาสำคัญในการออกแบบโครงสร้างเกมคือการจินตนาการตอนจบของเกมและวิธีที่จะทำให้ระบบทำงานได้ โครงสร้าง I/O จะทำให้ทราบถึงข้อจำกัดที่มีในโครงสร้างเกม ผู้ออกแบบมีอิสรภาพในการออกแบบโครงสร้างภายใน เพราะผู้เล่นจะติดต่อตรงส่วนนั้นโดยตรงไม่ได้หรือไม่จำเป็นต้องเข้าใจการทำงานภายในระบบ ผู้ออกแบบเกมต้องระบุคีย์เอลิเมนต์ (key element) จำนวนหนึ่งจากหัวข้อสิ่งแวดล้อมต่างๆ และเริ่มสร้างเกมขึ้นรอบๆ คีย์นั้น คีย์นี้เป็นสัญลักษณ์ของคำสั่งที่ระบุอยู่ภายใน โดยคีย์เอลิเมนต์จะต้องยอมให้ผู้เล่นทำสิ่งที่ต้องการทำได้เพื่อให้จินตนาการไปกับเกม เป็นต้นว่าในเกมต่อสู้ การยิงถือเป็นคีย์เอลิเมนต์ ถ้าการยิงของผู้เล่นถูกจำกัด ผู้เล่นก็จะไม่สามารถจินตนาการไปตามเกมได้ ซึ่งเกมส่วนมากจะใช้คีย์เอลิเมนต์หลายคีย์

ความผิดพลาดทั่วๆ ไปของผู้ออกแบบคือการใส่รายละเอียดต่างๆ ในโครงสร้างเกมมากเกินไปทำให้กลายเป็นเกมที่ประณีตมากเกินไปหรือผู้เล่นสามารถใช้วิธีโกงในการดำเนินเกมได้ จึงควรปรับเปลี่ยนโครงสร้าง I/O ให้เหมาะสม

2.4.3 **โครงสร้างโปรแกรม**

โครงสร้างโปรแกรมเสมือนเป็นวิธีที่จะเปลี่ยนโครงสร้าง I/O และโครงสร้างเกมให้ออกมาเป็นเกมจริงๆ เอลิเมนต์หนึ่งที่สำคัญที่สุดของโครงสร้างโปรแกรมคือแผนที่หน่วยความจำ ต้องมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
58782

การจัดสรรพื้นที่ของหน่วยความจำให้ดีเพราะอาจมีการใช้หน่วยความจำมากเกินไป ทำให้เหลือหน่วยความจำไม่เพียงพอสำหรับงานสำคัญๆ

2.4.4 การประเมินการออกแบบ

การประเมินการออกแบบคือการประเมินจุดบกพร่องทั้งหมดของการออกแบบที่จะทำให้เกมมีข้อเสีย ต้องมีการตรวจดูความมั่นคงของโครงสร้างเกม เนื่องจากเกมเป็นกระบวนการไดนามิค (dynamic process) ดังนั้นอาจมีเหตุการณ์ใดๆ ที่ทำให้เกมหลุดพ้นจากขอบเขตที่ควบคุมไว้ ตัวอย่างเช่น หากในเกมมีเงินให้ผู้เล่น เราควรจะเขียนคำสั่งควบคุมเอาไว้ด้วย และควรทดสอบหาวิธีลัดที่ผู้เล่นอาจใช้เพื่อให้ดำเนินไปถึงตอนจบเร็วๆ ได้ เพราะผู้เล่นจะไม่ได้รับประสบการณ์ที่ต้องการให้ผู้เล่นได้รับ

2.4.5 สรุปการออกแบบก่อนทำการเขียนโปรแกรม

กำหนดโครงสร้าง I/O และโครงสร้างภายในเกม แนวของเอกสารอ้างอิงนี้ควรจะเน้นด้านประสบการณ์ที่ผู้เล่นจะได้รับมากกว่าการเทคนิคที่ใช้ เปรียบเทียบเอกสารส่วนแรกนี้กับหมายเหตุโครงสร้างโปรแกรมเบื้องต้น ปรับเอกสารโครงสร้างโปรแกรมถ้าจำเป็น

2.4.6 การเขียนโปรแกรม

การเขียนโปรแกรมต้องการรายละเอียดมากกว่าสิ่งอื่นๆ บางกรณีที่เกมสร้างไม่สำเร็จเป็นเพราะโปรแกรมเมอร์ไม่มีทักษะในการเขียนเกมหรือไม่มีความพยายาม

2.4.7 การทดสอบการเล่นเกม

ในทางทฤษฎีแล้ว การทดสอบระบบเกมคือกระบวนการที่จะให้ข้อมูลเพื่อใช้ปรับปรุงการออกแบบเกม ส่วนทางปฏิบัติ การทดสอบระบบเกมแสดงให้เห็นแนวการออกแบบพื้นฐานและปัญหาในการแก้ไขโปรแกรม บางครั้งการทดสอบระบบเกมเผยให้เห็นส่วนประกอบต่างๆ หรือการกระทำที่มีน้อยหรือมากเกินไปซึ่งไม่ได้สังเกตก่อนหน้านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
การออกแบบและพัฒนาโปรแกรม

3.1 แนวทางการพัฒนาโปรแกรม

รูปแบบการเล่นของเกมจาวาโรบอทเวอร์ชันเดิม จะเริ่มจากการที่ผู้เล่นจะต้องทำการสร้างตัวหุ่นยนต์ของตัวเอง โดยใช้โปรแกรม Java Robot Creator ซึ่งการสร้างหุ่นยนต์ขึ้นมานั้นผู้เล่นต้องทำการเลือกส่วนประกอบของหุ่นยนต์และทำการเขียนโปรแกรมลงไป เพื่อใช้ในการควบคุมหุ่นยนต์ และใช้สำหรับตอบสนองเหตุการณ์ต่างๆ ที่เกิดขึ้นกับหุ่นยนต์ ในขณะที่นำไปประมวลผลบนสนามแข่งขัน ผู้เล่นสามารถส่งหุ่นยนต์รบเข้าสู่สนามแข่งขันได้โดยการอัพโหลดหุ่นยนต์ของตัวผู้เล่นเองไปยัง เซิฟร์เวอร์ แล้วทำการเลือกรูปแบบการแข่งขัน และเลือกหุ่นยนต์ที่ต้องการ จากนั้นหุ่นยนต์ก็จะทำงานตามที่ผู้เล่นได้โปรแกรมไว้ในตอนสร้าง โดยสามารถดูการแข่งขันได้จากบราวเซอร์ รูปแบบการแข่งขันแบ่งออกเป็น 2 โหมดคือสตอรี่โหมดและไฟต์ติ้งโหมด ลักษณะของสนามทั้ง 2 โหมดนี้จะเป็นสนามโล่ง

ในการพัฒนาเกมจาวาโรบอทนั้น ยังคงรูปแบบการสร้างหุ่นยนต์โดยใช้โปรแกรม Java Robot Creator ไว้เหมือนเดิม ผู้พัฒนาได้ทำการพัฒนาในส่วนของรูปแบบสนาม โดยเพิ่มเติมสิ่งกีดขวางเข้าไปในสนามเพื่อเป็นอุปสรรคในการเดินและการค้นหาหุ่นยนต์รบของฝ่ายตรงข้าม โดยส่วนที่เพิ่มเติมเข้ามานั้นชื่อว่าเทรนนิ่งโหมด ทางผู้พัฒนาได้แบ่งประเภทของสนามออกเป็น 2 ประเภทใหญ่ โดยจะอธิบายในหัวข้อถัดไป การที่เพิ่มเติมสิ่งกีดขวางลงไปในสนามทำให้ต้องสร้างคลาสต่างๆ ขึ้นมาเพื่อรองรับการพัฒนาดังกล่าว ได้แก่คลาส JBulete JHole JWall JBorder และ JavaRobot ซึ่งคลาสเหล่านี้จะเป็นคลาสที่จะนำไปสร้างเป็นวัตถุที่เป็นองค์ประกอบของสนาม นอกจากนี้ยังมีคลาส JLayout ซึ่งเป็นคลาสแม่ที่ต้องนำมาทำการสร้างคลาสลูกเพื่อกำหนดคุณลักษณะและรูปแบบของสนามลงไป

นอกจากนั้นทางผู้พัฒนาได้สร้างกรรมวิธีในการตรวจสอบสิ่งกีดขวางและการรายงานสิ่งกีดขวาง โดยมี 2 รูปแบบคือ

1) การตรวจสอบในรูปแบบการสแกน คือจะตรวจสอบสิ่งกีดขวางที่อยู่ภายในระยะการสแกน และรายงานผลการสแกนทิศทางที่พบวัตถุ พร้อมทั้งระบุชนิดของวัตถุที่สแกนพบ

2) การตรวจสอบในรูปแบบเรดาร์ กรรมวิธีการตรวจสอบรูปแบบนี้จะถูกเรียกใช้โดยอัตโนมัติเมื่อการสแกนของหุ่นยนต์ทั้ง 3 ทิศทางสแกนพบสิ่งกีดขวาง โดยเรดาร์จะทำการบอกให้หุ่นยนต์ทราบว่าทางด้านซ้ายและทางด้านขวาของสิ่งที่สแกนพบนั้นเป็นวัตถุชนิดใด สามารถเดินผ่านไปได้หรือไม่ พร้อมทั้งบอกค่าระยะทาง องศาระหว่างหุ่นยนต์กับวัตถุ และทิศทางให้กับหุ่นยนต์ทราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

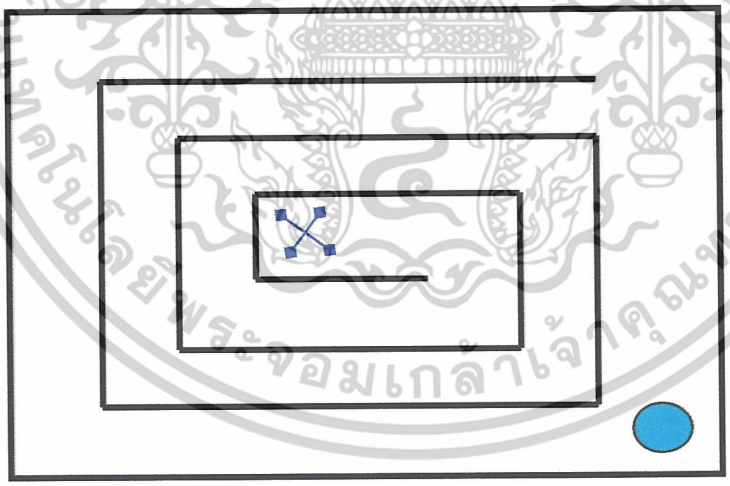
การปรับปรุงการสื่อสารของวัตถุต่างๆ ที่ประมวลผลอยู่บนสนาม ก็เป็นอีกสิ่งหนึ่งที่ต้องทำการปรับปรุง เพื่อให้รองรับกับการปรับปรุงรูปแบบสนาม โดยจะต้องทำการปรับปรุงคลาส JProcess ซึ่งใช้รองรับข้อความและคำร้องขอของวัตถุต่างๆ แล้วทำการตัดสินใจว่าคำร้องขอนั้นมีความเป็นไปได้ในทางปฏิบัติหรือไม่ ในการปรับปรุง JProcess ได้ทำการแยกส่วนของการจัดเก็บข้อความออกมาเอาไว้ในคลาส JQueue

3.2 การออกแบบสนาม

สนามในการทดสอบหุ่นยนต์ก่อนที่จะนำหุ่นยนต์ไปแข่งจริงนั้นสามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ๆ คือ

3.2.1 สนามทดสอบกรรมวิธีการเดินแบบปกติ

สนามในรูปแบบนี้ ผู้พัฒนาจะทำการสร้างทางเดินไว้ให้ จากจุดเริ่มต้นไปยังจุดหมายปลายทาง ซึ่งที่จุดหมายปลายทางนั้นจะมีหุ่นยนต์ที่ผู้พัฒนาเตรียมเอาไว้ให้ผู้เล่นต่อสู้ จนกว่าจะรู้ผลการต่อสู้ถึงจะจบการประมวลผลของสนามนี้ ผู้เล่นจะต้องพัฒนากรรมวิธีการเดินของหุ่นยนต์ให้เดินไปตามทางที่กำหนดเอาไว้ จากจุดเริ่มไปจนถึงปลายทางที่กำหนด ตัวอย่างของสนามทดสอบกรรมวิธีการเดินแบบปกติ เช่นสนามทดสอบกรรมวิธีการเดินแบบปกติชนิดก้นหอย

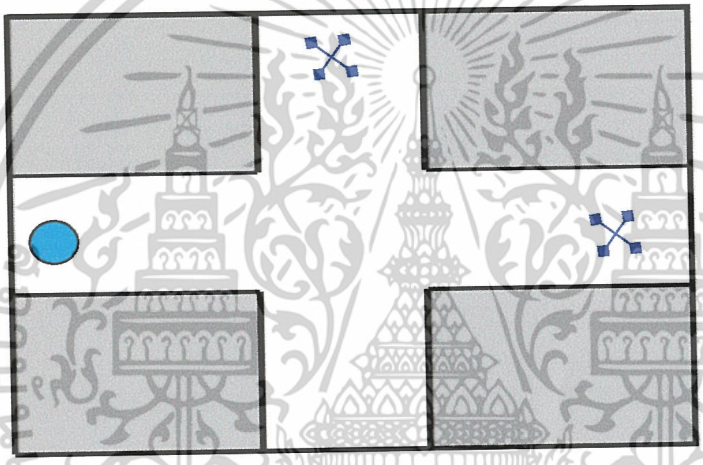


รูปที่ 3.1 สนามทดสอบกรรมวิธีการเดินแบบปกติ ชนิดก้นหอย

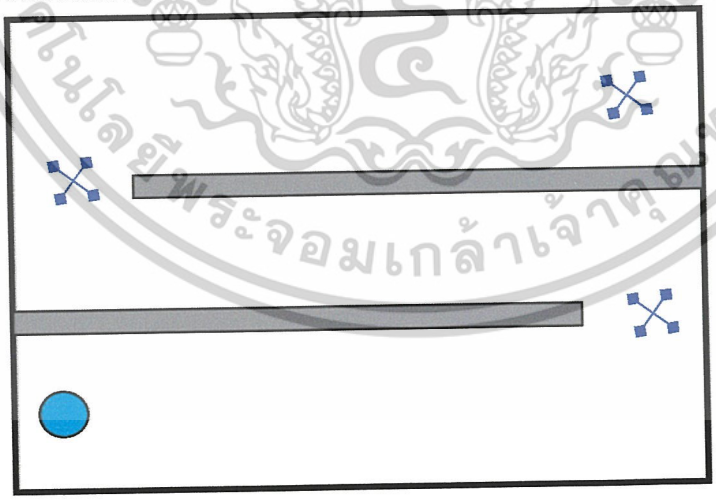
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ

สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ เป็นสนามที่มีลักษณะเป็นทางแยกหรือมีสิ่งกีดขวาง วางขวางกั้นทางเดินเอาไว้ หุ่นยนต์ที่นำมาทดสอบในสนามรูปแบบนี้จะต้องมีกรรมวิธีในการตัดสินในว่าจะเดินไปในลักษณะใดเมื่อเจอทางแยกหรือสิ่งกีดขวางที่วางขวางกั้นทางเดิน หุ่นยนต์ของผู้เล่นจะตัดสินใจว่าจะทำอย่างไร เพื่อสามารถเดินต่อไปได้ สิ่งกีดขวางที่นำมาใช้ในสนามรูปแบบนี้คือหุ่นยนต์ที่ทางผู้พัฒนาได้สร้างขึ้นเอาไว้ โดยเมื่อหุ่นยนต์ที่ผู้เล่นสร้างขึ้นเดินมาพบจะต้องทำลายเพื่อที่จะได้เดินไปสู้กับหุ่นยนต์ตัวอื่นๆ ในสนามจนครบทุกตัว สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ แบ่งออกเป็น 3 สนามย่อยด้วยกันดังนี้

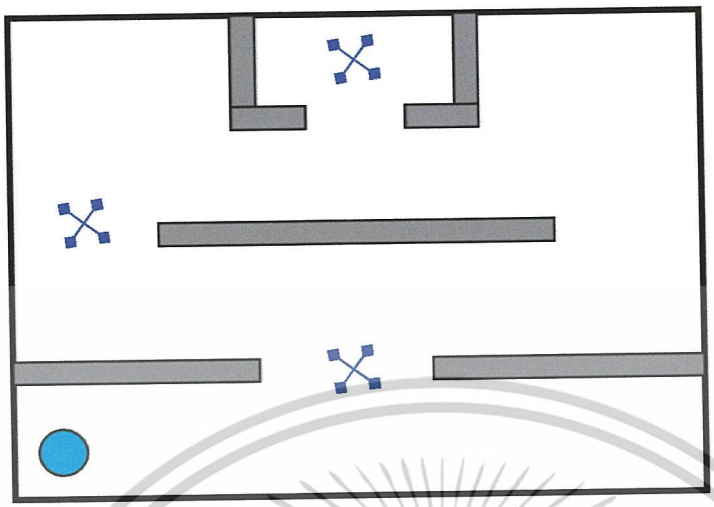


รูปที่ 3.2 สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ ชนิดที่ 1



รูปที่ 3.3 สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ ชนิดที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ ชนิดที่ 3

หมายเหตุ

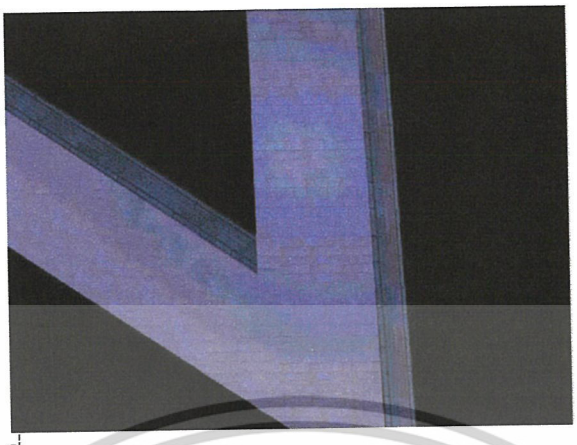
- สัญลักษณ์ แทนหุ่นยนต์ของผู้เล่น
- สัญลักษณ์ แทนหุ่นยนต์ทดสอบที่ผู้พัฒนาสร้างขึ้น
- เส้นทางที่หุ่นยนต์สามารถเดินได้ แทนด้วยสีขาว
- สิ่งกีดขวาง แทนด้วยสีเทา และเส้นสีดำ

ลักษณะสนามทดสอบที่ออกแบบมาทั้งหมดนั้น ทางผู้พัฒนาได้ทำการเลือกในแต่ละแบบมาสร้าง โดยได้สร้างออกมาเป็นสนามทดสอบทั้งหมด 9 สนาม ตามการแบ่งประเภทสนามดังนี้

สนามทดสอบกรรมวิธีการเดินแบบปกติ

รูปที่ 3.5 สนามทดสอบแบบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 สนามทดสอบแบบที่ 4

รูปที่ 3.7 สนามทดสอบแบบที่ 9

สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ

รูปที่ 3.8 สนามทดสอบแบบที่ 2

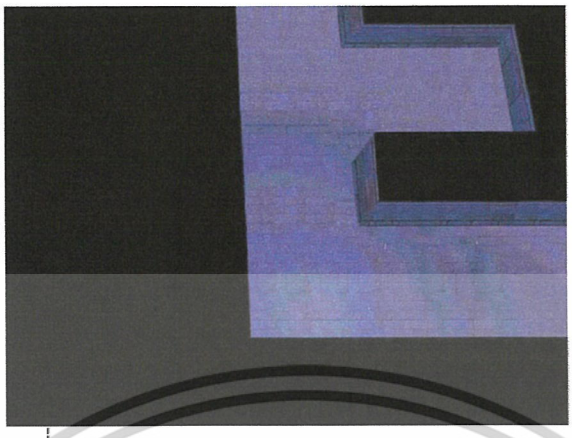
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 สนามทดสอบแบบที่ 3

รูปที่ 3.10 สนามทดสอบแบบที่ 5

รูปที่ 3.11 สนามทดสอบแบบที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 สนามทดสอบแบบที่ 7

รูปที่ 3.13 สนามทดสอบแบบที่ 8

หมายเหตุ รูปภาพสนามได้ทำการดัดแปลงมาจากรูปภาพของเกม Ragnarok Online

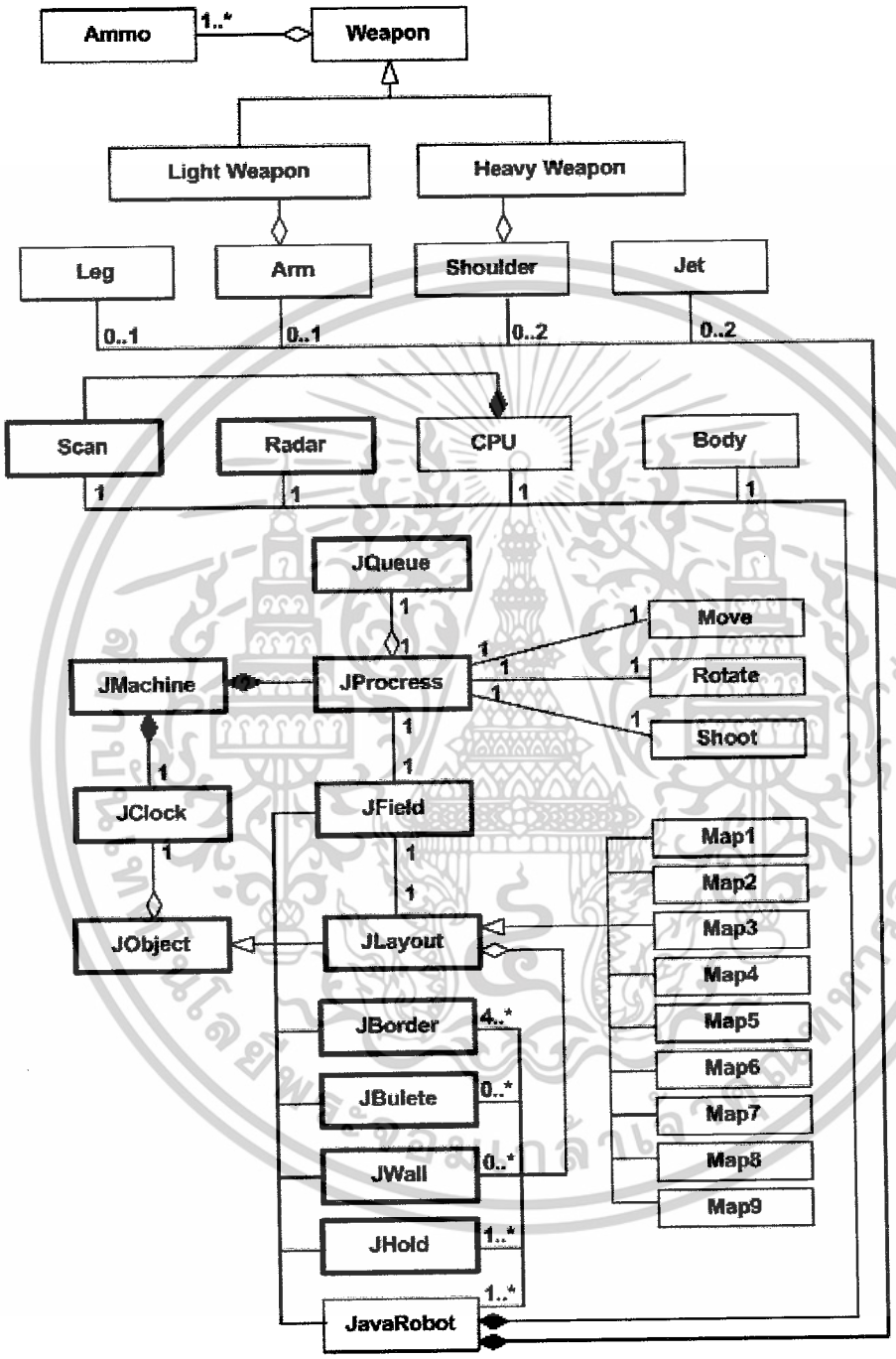
3.3 การออกแบบสถาปัตยกรรมที่ใช้ในเกม

ใช้สถาปัตยกรรมที่เป็นลักษณะ 2-Tier Client/Server โดยทางฝั่งไคล์เอนท์จะติดต่อกับเซิร์ฟเวอร์ ผ่านโปรโตคอล (protocol) HTTP โดยผู้เล่นจะต้องทำการลงทะเบียนผ่านเว็บเพื่อเข้าสู่หน้าข้อมูลของผู้ใช้ สำหรับการติดต่อกับ Server จะใช้ ภาษา PHP เป็นสคริปต์ (Script) เพื่อโต้ตอบกับเซิร์ฟเวอร์ เช่น การเลือกหุ่นยนต์เพื่อเข้าทำการต่อสู้ และผู้เล่นสามารถที่จะชมการแข่งขันของหุ่นยนต์ได้จากการเรียกแอปเพล็ตผ่านทางเว็บบราวเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบไดอะแกรมของเกม

3.4.1 คลาสไดอะแกรมของสนามทดสอบ



รูปที่ 3.14 คลาสไดอะแกรมของสนามทดสอบ

หมายเหตุ - บริเวณสีเข้มคือส่วนที่พัฒนาขึ้นในการทำโครงงานครั้งนี้

- บริเวณสีอ่อนเป็นส่วนที่พัฒนาขึ้นในโครงงานครั้งที่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Class Scan เป็น class ที่เก็บ method ในการคำนวณจุดของ scan โดยจะคำนวณจากค่าองศาการ scan, ระยะทางและตำแหน่งปัจจุบันของหุ่นยนต์ นอกจากนี้ยังทำหน้าที่เก็บผลของการ scan ที่ได้จากการทำงานของ Class Jfield เอาไว้ให้หุ่นยนต์ใช้ในการตัดสินใน จุดในการ scan มี 3 จุดคือ จุดทางด้านซ้าย จุดตรงกลาง และจุดทางด้านขวา ทำให้การรายงานผลการ scan ก็จะรายงานผล 3 จุดเช่นเดียวกัน Class scan มี class CPU เป็นองค์ประกอบเพื่อใช้สำหรับการนำค่าตำแหน่งปัจจุบันของหุ่นยนต์, องศา และ ระยะการแสกน มาคำนวณจุด scan ของหุ่นยนต์

Class Radar เป็น class ที่ใช้ในการเก็บผลการ scan โดยใช้ radar ซึ่ง Jfield จะเรียกใช้ radar โดยอัตโนมัติทันทีเมื่อผลการ scan ของหุ่นยนต์ ทั้ง 3 จุดพบว่าเป็นเป็นผนัง

Class scan กับ Class radar เป็น class องค์ประกอบของ class JavaRobot เพื่อให้หุ่นยนต์สามารถนำผลของการ scan มาใช้ในการตัดสินใจ

Class JObject เป็นคลาสแม่ของทุกคลาสที่ทางผู้พัฒนาได้สร้างขึ้น ทำหน้าที่เก็บรวบรวมคุณสมบัติพื้นฐานของคลาสลูกที่มีเหมือนกันเอาไว้ โดย class JObject จะมี class JClock เป็นองค์ประกอบ ดังนั้นคลาสลูกที่ขยายมาจาก class JObject ก็จะมีองค์ประกอบเป็น class JClock ด้วย เหตุที่ทุกวัตถุต้องมีองค์ประกอบ เป็น class JClock ก็เพื่อจะได้ดึงค่าเวลาจาก class JClock มาใช้ในการสร้างข้อความเพื่อส่งไปทำการประมวลผล นอกจากนั้น JObjcet ยังมี JQueue เป็นองค์ประกอบ การสร้างความสัมพันธ์ระหว่าง JObject กับ JQueue จะผ่านทาง class JProcress

Class JClock เป็นคลาสที่ทำหน้าที่ในการเวลาของระบบ ทุกๆ วัตถุจะต้องดึงค่าเวลา จาก JClock มาใช้ในการสร้างข้อความเพื่อระบุว่าข้อความใดถูกสร้างก่อนและหลัง

Class JBullet เป็นคลาสซึ่งทำหน้าที่เก็บรวบรวมคุณสมบัติของกระสุนที่หุ่นยนต์ยิง

Class JHold, Class JWall และ Class JBorder ทั้ง 3 คลาสนี้จะเก็บรวบรวมคุณสมบัติของพื้นที่ว่าง ผนัง และ ขอบสนามตามลำดับ ซึ่งคุณสมบัติที่เก็บ ได้แก่ ค่าจุดบนซ้าย และค่าจุดล่างขวาของวัตถุ เพื่อใช้ในการบอกตำแหน่งและพื้นที่ของวัตถุนั้น

Class JLayout เป็นคลาสต้นแบบที่มีไว้สำหรับนำไปสร้างสนามทดสอบในรูปแบบต่างๆ โดยองค์ประกอบของ Class JLayout ได้แก่ วัตถุที่สร้างจาก Class JHold, Class JWall, Class Jborder Class JavaRobot และ Class JBullet โดยคุณสมบัติของวัตถุแต่ละก้อนจะขึ้นอยู่กับรูปแบบของสนาม เนื่องจากวัตถุองค์ประกอบของ Class JLayout มีหลายชนิดและหลายวัตถุ ผู้พัฒนาได้ใช้โครงสร้างข้อมูลแบบ Vector ของ ภาษา Java ในการเก็บวัตถุเหล่านั้น ดั้งนั้นกรรมวิธีของ Class JLayout ส่วนใหญ่จะเกี่ยวข้องกับการจัดการโครงสร้างข้อมูล Vector เช่น การเพิ่มวัตถุ การค้นหาวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นต้น ผู้พัฒนา ได้นำ Class JLayout ที่เป็นคลาสต้นแบบมาสร้างเป็นคลาสสนามในรูปแบบต่างๆ 9 รูปแบบ ได้แก่ Class Map1 ไปจนถึง Class Map9

Class JField เป็นคลาสที่รวบรวมกรรมวิธีในการตรวจสอบวัตถุต่างๆ ที่อยู่ในรัศมีการ scan และเหตุการณ์ต่างๆ ที่เกิดขึ้นกับตัวหุ่นยนต์ โดยกรรมวิธีการตรวจสอบของ Class JField จะพิจารณาจากกลุ่มของวัตถุองค์ประกอบ ของ Class JLayout ดังนั้น Class JField และ Class JLayout จึงมีความสัมพันธ์กันซึ่งกันและกันในแบบหนึ่งต่อหนึ่ง

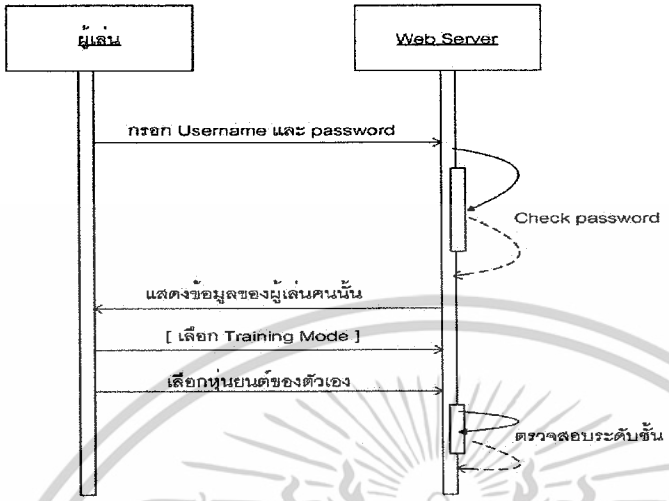
Class JProcress เป็นคลาสที่รวบรวมกรรมวิธีในการประมวลผลและจัดการข้อความที่วัตถุจากคลาสต่างๆ อาทิเช่น วัตถุจาก class JFiled class JavaRobot class JBullet เป็นต้น วัตถุเหล่านี้จะส่งข้อความมาเก็บไว้ในแถวคอย ของ class JQueue ซึ่งเป็นองค์ประกอบของ class JProcress การประมวลผลข้อความของ JProcress จะเริ่มจากการพิจารณาว่าข้อความที่วัตถุต่างๆ ส่งมานั้นมีความเป็นไปได้ในทางปฏิบัติได้หรือไม่ ถ้าเป็นไปได้ก็จะปฏิบัติตาม Class JProcress มีองค์ประกอบ เป็น class Move class Rotate และ class Shoot โดย class Move ทำหน้าที่ในการประมวลผล ข้อความในเรื่องการเคลื่อนที่ของหุ่นยนต์ class Rotate ทำหน้าที่ในการประมวลผลข้อความในการหมุนของหุ่นยนต์ โดยจะทำการคำนวณองศาหน้าของหุ่นยนต์และ class Shoot ทำหน้าที่ในการประมวลผลการยิงของหุ่นยนต์โดยการสร้างวัตถุจาก class JBullet พร้อมทั้งกำหนดค่าเริ่มต้นให้กับวัตถุที่สร้างนั้น และจักการการเคลื่อนที่ของกระสุนที่ยิงออกไป นอกจากนั้น JProcress ยังมีองค์ประกอบ เป็น class JQueue ซึ่งถือว่าเป็นองค์ประกอบที่สำคัญ เพื่อใช้ในการเก็บข้อความที่วัตถุต่างๆ ส่งมาเก็บไว้เพื่อรอการประมวลผล class JProcress ต้องมีความสัมพันธ์กับ class JField ในแบบหนึ่งต่อหนึ่ง เพื่อเรียกใช้กรรมวิธีต่างๆ ของ class JField ในการประมวลผลข้อความ ในเรื่องการ scan ของหุ่นยนต์ และกรรมวิธีในการตรวจสอบเหตุการณ์ที่เกิดขึ้นกับหุ่นยนต์

Class JQueue เป็นคลาสที่ทำหน้าที่ในการเก็บรวบรวมข้อความจากวัตถุต่างๆ เอาไว้ภายในโครงสร้าง Vector ภายใน class ดังนั้นกรรมวิธีต่างๆ ของ class JQueue จะเกี่ยวข้องกับการจัดการข้อความ ได้แก่การเก็บข้อความที่วัตถุต่างๆ ส่งมาลงในโครงสร้าง Vector การลบข้อความออกจากโครงสร้าง Vector การค้นหาข้อความภายในโครงสร้าง Vector เป็นต้น

Class JMachine เป็นคลาสที่ทำหน้าที่ในการสั่งงานให้ JProcress ทำการประมวลผลข้อความที่วัตถุต่างๆ ส่งมา นอกจากนั้นยังทำหน้าที่ในการจัดจังหวะเวลาของ JClock ให้สอดคล้องกับการทำงานของวัตถุแต่ละก้อนในระบบ class JMachine มีองค์ประกอบเป็น class JProcress เพื่อควบคุมและสั่งการให้ JProcress ประมวลผลข้อความและมี class JClock เป็นองค์ประกอบเพื่อสั่งให้ JClock เพิ่มเวลาของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

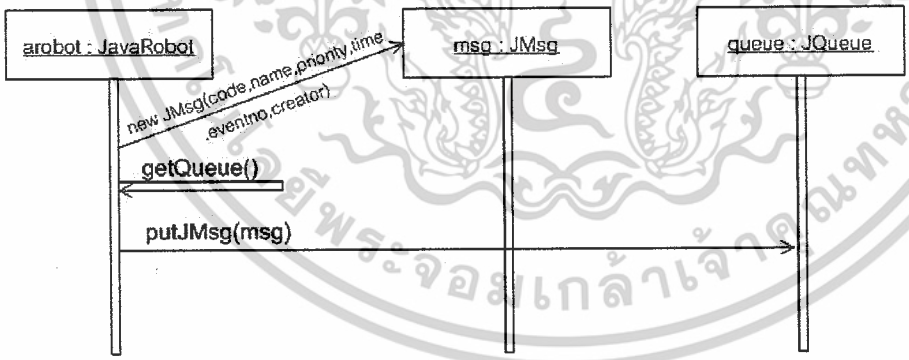
3.4.2 ซีเควนซ์ไดอะแกรมของการส่งหุ่นยนต์เข้าสนามทดสอบ



รูปที่ 3.15 ซีเควนซ์ไดอะแกรมของการส่งหุ่นยนต์เข้าสนามทดสอบ

จากรูปที่ 3.15 การส่งหุ่นยนต์เข้าสนามทดสอบของผู้เล่น ผู้เล่นจะต้องกรอกข้อมูลสมาชิก ได้แก่ username และpassword ผ่านทางหน้าเว็บ จากนั้นระบบจะทำการตรวจสอบสิทธิ์ของผู้ใช้ และนำข้อมูลของผู้เล่นขึ้นมาแสดง ผู้เล่นเลือกรูปแบบการเล่น แบบ Training Mode และทำการเลือกหุ่นยนต์ที่ต้องการนำไปใช้ในสนามทดสอบ

3.4.3 ซีเควนซ์ไดอะแกรมของการส่งข้อความระหว่างหุ่นยนต์กับแถวคอย



รูปที่ 3.16 ซีเควนซ์ไดอะแกรมของการส่งข้อความระหว่างหุ่นยนต์กับหน่วยประมวลผลข้อความ

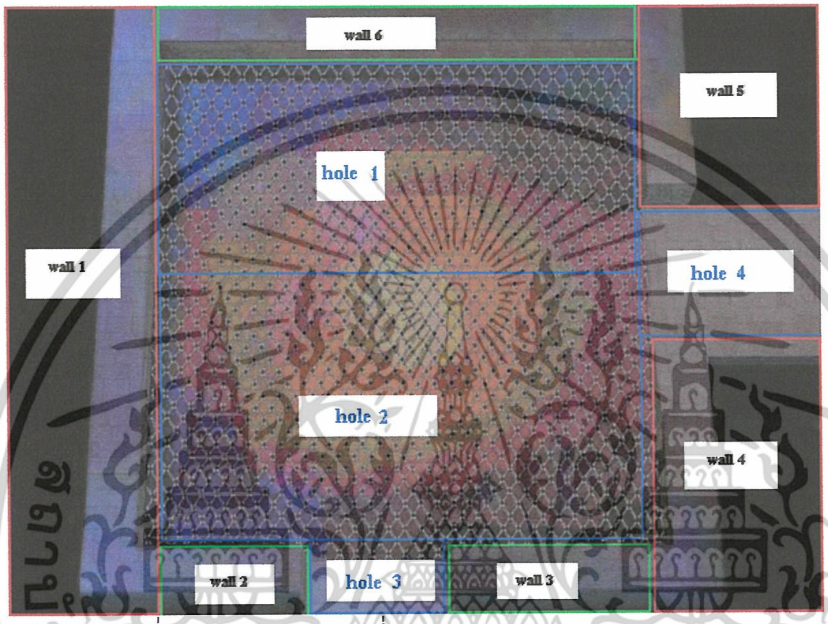
จากรูปที่ 3.16 การส่งข้อความระหว่างหุ่นยนต์กับหน่ายประมวลผลข้อความจะเริ่มต้นจากที่หุ่นยนต์ทำการสร้าง Object จาก class JMsg พร้องทั้งกำหนดค่าต่างๆ ลงไป จากนั้นหุ่นยนต์จะทำการส่งข้อความนั้นไปเก็บไว้ใน JQueue เพื่อรอให้ JProcress นำไปประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

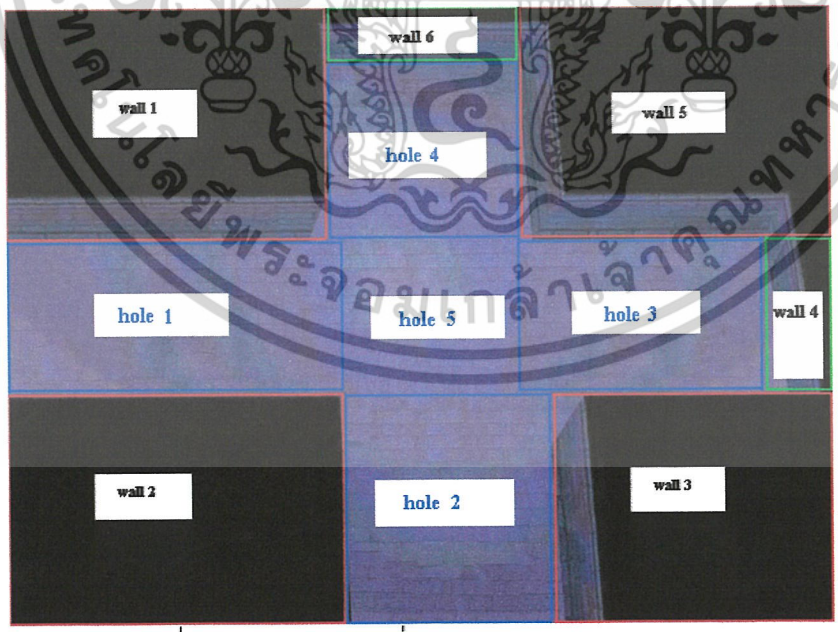
3.5 อัลกอริทึมที่ใช้ในการเขียนเกม

3.5.1 การแบ่งส่วนของสนาม

ในแต่ละสนามนั้นทางผู้พัฒนาได้ทำการแบ่งส่วนของสนามออกเป็นหลายๆส่วนๆ ประกอบด้วยส่วนที่เป็นกำแพง ช่องว่างสำหรับทางเดิน และขอบสนาม ซึ่งกำแพงจะแทนด้วยสัญลักษณ์ wall และช่องว่างสำหรับทางเดินแทนด้วยสัญลักษณ์ hole แสดงให้เห็นดังรูปต่อไปนี้

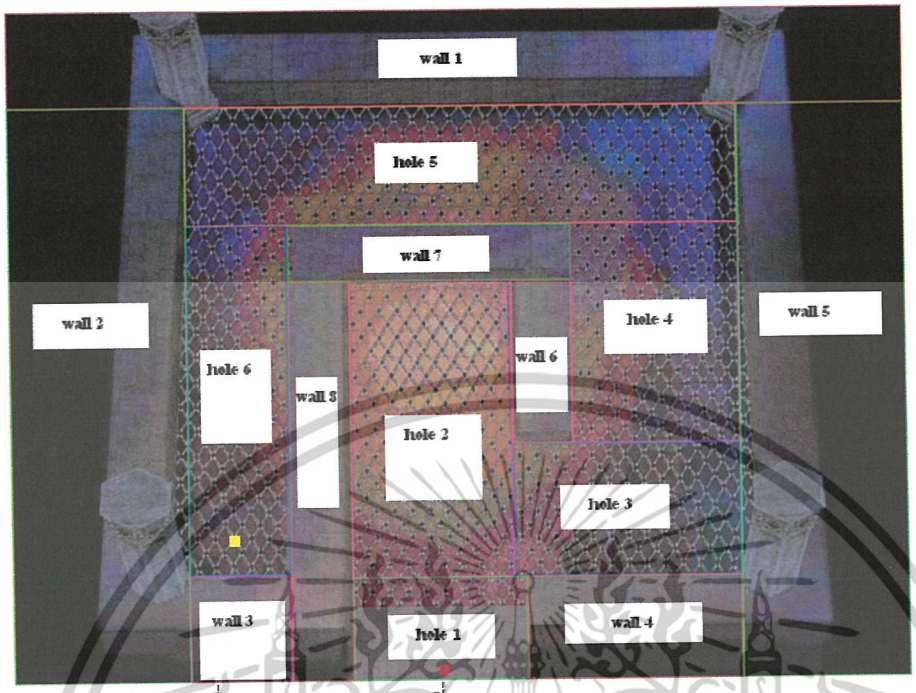


รูปที่ 3.17 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 1

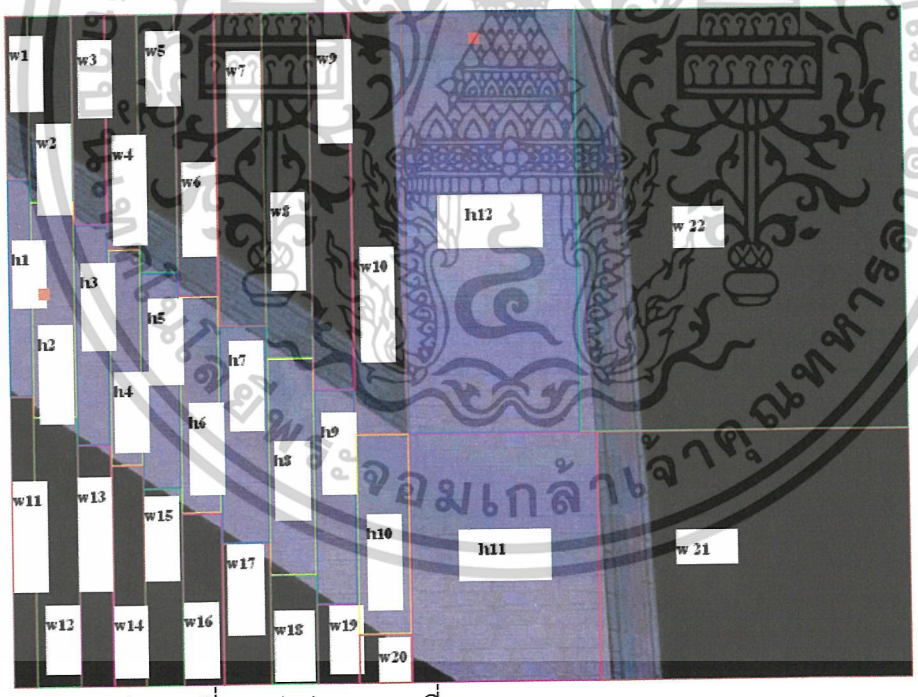


รูปที่ 3.18 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

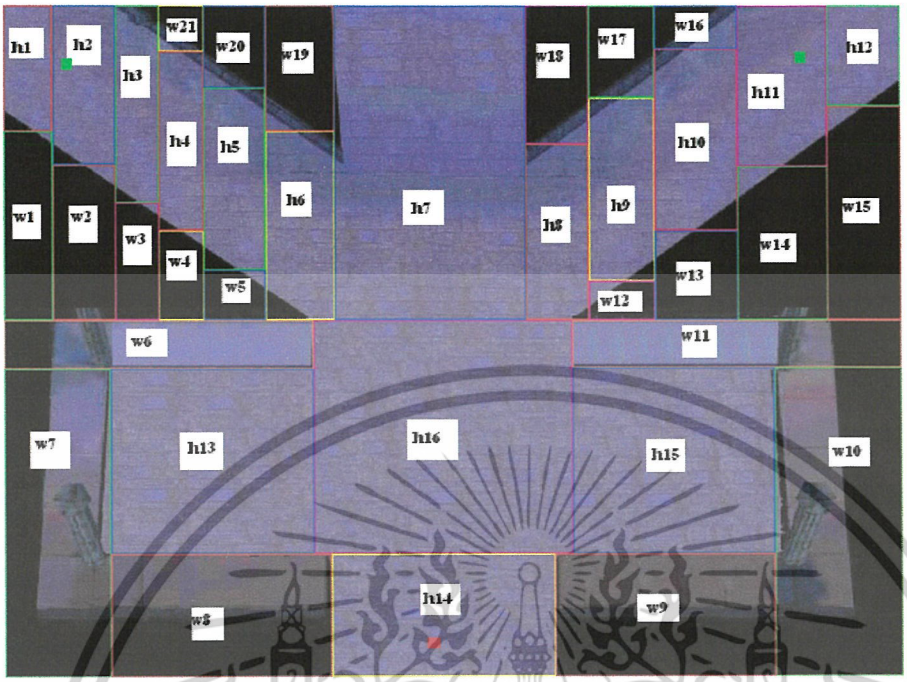


รูปที่ 3.19 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 3

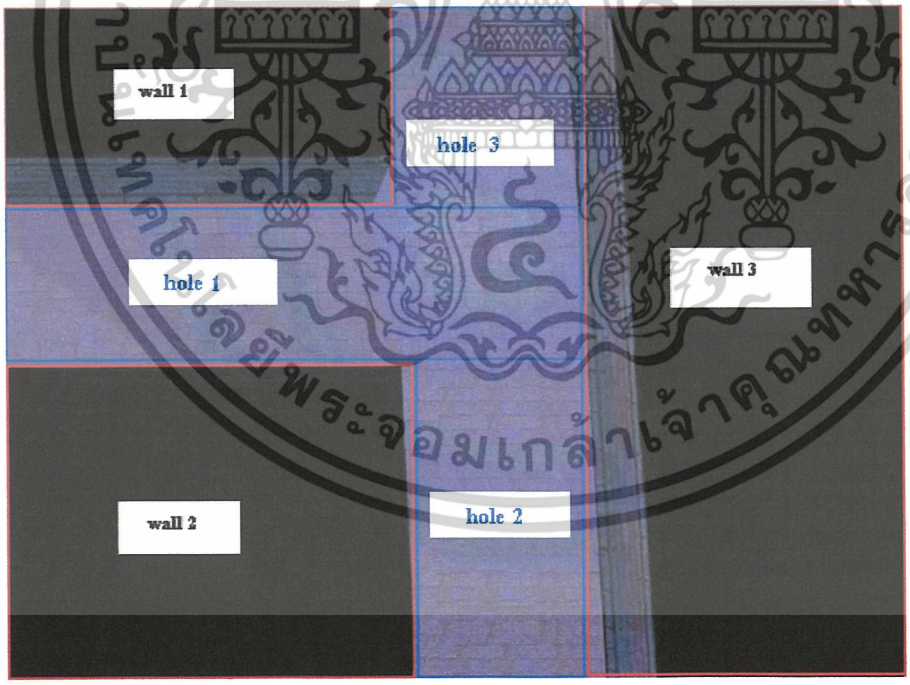


รูปที่ 3.20 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

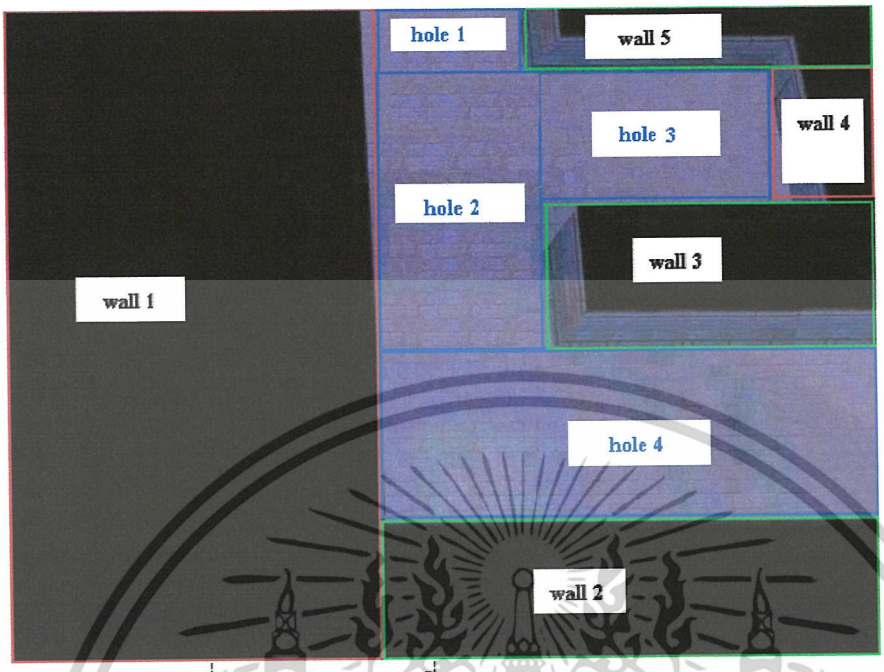


รูปที่ 3.21 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 5

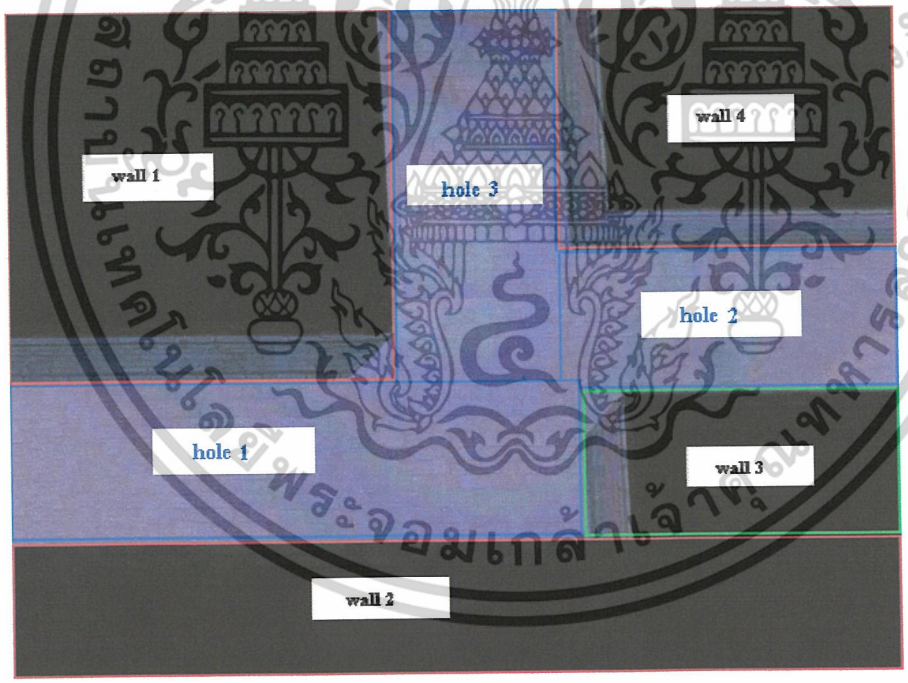


รูปที่ 3.22 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

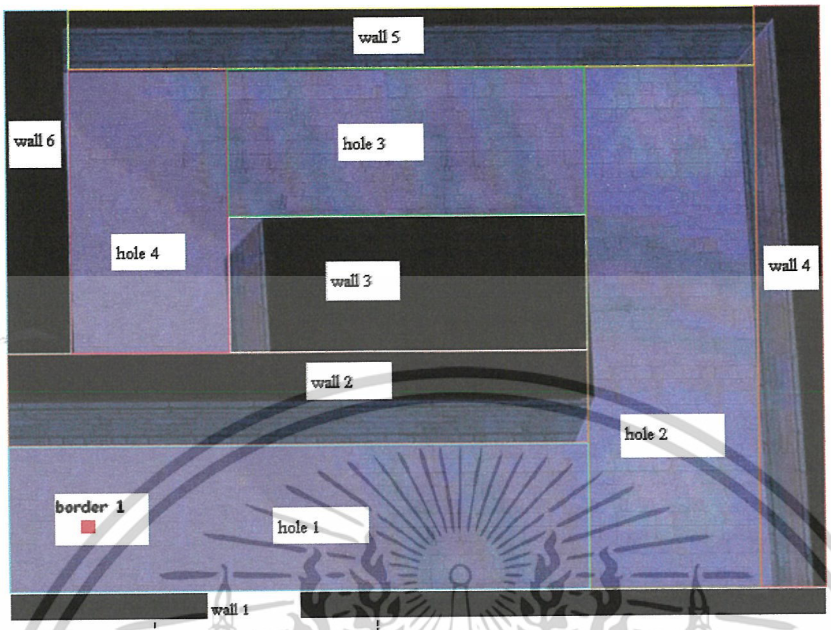


รูปที่ 3.23 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 7



รูปที่ 3.24 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 แสดงรูปสนามที่ถูกแบ่งส่วน สนามที่ 9

การแบ่งสนามออกเป็นส่วนต่างๆ ดังที่ได้กล่าวมาข้างต้นนั้น เพื่อแสดงให้เห็นว่าสนามต่างๆ นั้นมีองค์ประกอบ เป็น วัตถุใดบ้าง และแต่ละวัตถุมีพื้นที่ครอบคลุมบริเวณใดบ้าง ตำแหน่งอยู่ที่ใด

3.5.2 อัลกอริทึมการสื่อสารระหว่างวัตถุ ด้วย การส่งผ่านข้อความ

วัตถุหนึ่งๆ สามารถสื่อสารกับวัตถุอื่นๆในระบบ เพื่อรายงานเหตุการณ์ที่เกิดขึ้น และส่งข้อมูลให้วัตถุต่างๆ ได้โดยทำการสร้าง วัตถุจาก class JMessage แล้วใส่ข้อมูลที่ต้องการส่งลงไปเป็นค่าคุณสมบัติของวัตถุนั้น ต่อไปนี้จะขอเรียก วัตถุที่สร้างจาก class JMessage ว่า “ข้อความ”

เมื่อ วัตถุหนึ่งๆ สร้างข้อความขึ้นมาแล้ว ก็จะทำการส่งข้อความที่สร้างขึ้นไปเก็บใน โครงสร้างแบบ Vector ที่อยู่ภายใน class JQueue เพื่อรอให้ JProcress นำเอาข้อความเหล่านั้นไปประมวลผล หลักในการจัดเรียงข้อความเพื่อเก็บข้อความลงในโครงสร้างแบบ Vector ของ class JQueue จะพิจารณาจาก เวลาที่ข้อความถูกสร้างขึ้น ข้อความที่ถูกสร้างขึ้นก่อนจะเก็บอยู่ด้านบนของ Vector ถ้าในกรณีข้อความถูกสร้างขึ้น ณ เวลาเดียวกัน จะพิจารณา ที่ค่าความสำคัญของข้อความนั้น ข้อความที่มีความสำคัญมากจะมีค่าความสำคัญน้อยและจะถูกจัดเก็บอยู่ด้านบนของโครงสร้าง Vector ส่วนข้อความที่มีความสำคัญน้อยจะมีค่าความสำคัญมากและจะถูกจัดเก็บไว้ด้านล่างของโครงสร้าง Vector ในกรณีที่ข้อความถูกสร้าง ณ เวลาเดียวกัน และมีค่าความสำคัญเท่ากัน ก็จะพิจารณาการมาถึง ของข้อความ ข้อความที่มาถึงก่อนจะถูกเก็บอยู่ด้านบน ข้อความที่มาถึงทีหลังจะถูกเก็บด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำข้อความไปประมวลผลของ JProcress จะดึงเอาเฉพาะข้อความที่อยู่ด้านบนสุดของโครงสร้าง Vector ภายใน JQueue ไปทำการประมวลผลก่อน โดยถ้าเป็นข้อความชนิดคำร้อง JProcress จะพิจารณาถึงความเป็นไปได้ของคำร้องนั้น ว่าสามารถทำตามคำร้องข้อนั้นได้หรือไม่ ก่อนที่จะปฏิบัติตาม แต่ถ้าเป็นข้อความประเภทที่ต้องการส่งข้อมูลไปถึง วัตถุอื่นๆ JProcress ก็จะส่งข้อความไปยังวัตถุนั้นๆ

ผู้พัฒนาได้สร้างคลาส JMessage ขึ้นมาสำหรับให้บริการคลาสอื่นๆ ในการสร้างข้อความเพื่อใช้ในการสื่อสารระหว่างกัน โดยผู้พัฒนาได้ทำการโอเวอร์โหลด (overload) คอนสตรักเตอร์เมธตอดของคลาส JMessage ขึ้นมา 4 รูปแบบ ต่อไปนี้

รูปแบบที่ 1 ใช้สำหรับการบอกเหตุการณ์ต่างๆ ที่เกิดขึ้นกับออบเจ็กต์หนึ่งๆ โดยมีลำดับของพารามิเตอร์ ดังนี้

ตารางที่ 3.1 แสดงลำดับของพารามิเตอร์ของรูปแบบที่ 1

ลำดับที่	ชนิดตัวแปร	ชื่อตัวแปร	หมายเหตุ
1	int	code	รหัสข้อความ
2	String	name	ชื่อข้อความ
3	int	priority	ค่าแสดงความสำคัญของข้อความ
4	long	time	เวลาที่สร้างข้อความ
5	JObject	creator	ผู้สร้างข้อความหรือผู้ส่ง
6	JObject	responder	ผู้ตอบสนองหรือผู้รับปลายทาง

รูปแบบที่ 2 ใช้สำหรับให้หุ่นยนต์สร้างข้อความร้องขอเพื่อทำการเคลื่อนที่และยิงกระสุนใส่ศัตรู

ตารางที่ 3.2 แสดงลำดับของพารามิเตอร์ของรูปแบบที่ 2

ลำดับที่	ชนิดตัวแปร	ชื่อตัวแปร	หมายเหตุ
1	int	code	รหัสข้อความ
2	String	name	ชื่อข้อความ
3	int	priority	ค่าแสดงความสำคัญของข้อความ
4	long	time	เวลาที่สร้างข้อความ
5	JObject	creator	ผู้สร้างข้อความหรือผู้ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 (ต่อ)

6	JObject	responder	ผู้ตอบสนองหรือผู้รับปลายทาง
7	int	evenno	หมายเลขอ้างอิงเหตุการณ์
8	int	degree	อาศาในการยิงและการเคลื่อนที่ของหุ่นยนต์
9	int	distant	ระยะทางในการเคลื่อนที่ของหุ่นยนต์หรือกระสุน
10	int	direction	ทิศทางในการเคลื่อนที่ของหุ่นยนต์หรือกระสุน

รูปแบบที่ 3 ใช้สำหรับให้หุ่นยนต์สร้างข้อความสำหรับยกเลิกกิจกรรมที่ตอบสนองกับเหตุการณ์ที่เกิดขึ้น

ตารางที่ 3.3 แสดงลำดับของพารามิเตอร์ของรูปแบบที่ 3

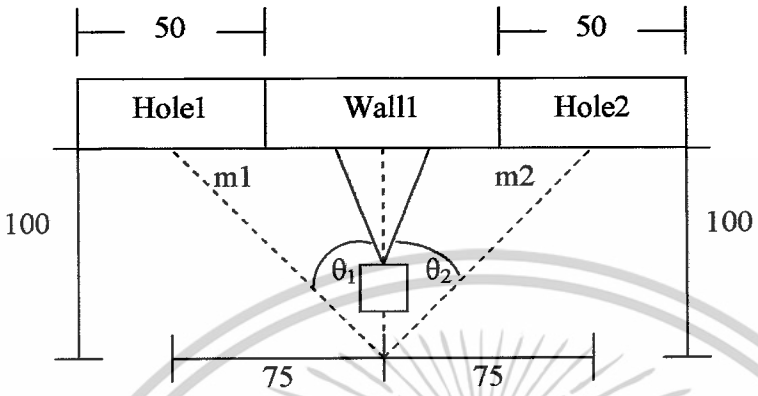
ลำดับที่	ชนิดตัวแปร	ชื่อตัวแปร	หมายเหตุ
1	int	code	รหัสข้อความ
2	String	name	ชื่อข้อความ
3	int	priority	ค่าแสดงความสำคัญของข้อความ
4	long	time	เวลาที่สร้างข้อความ
5	JObject	creator	ผู้สร้างข้อความหรือผู้ส่ง
6	JObject	responder	ผู้ตอบสนองหรือผู้รับปลายทาง
7	int	evenno	หมายเลขอ้างอิงเหตุการณ์
8	int	ignoreno	หมายเลขอ้างอิงเหตุการณ์ที่ต้องการให้ยกเลิกกิจกรรม

3.5.3. อัลกอริทึมการตรวจสอบสิ่งกีดขวางแบบเรดาร์

เมื่อหุ่นยนต์ทำการสแกนแล้วพบสิ่งกีดขวางทั้ง 3 ด้าน คือ ด้านซ้าย ตรงกลางและด้านขวา JField จะทำการเรียกใช้กรรมวิธีการตรวจสอบสิ่งกีดขวางแบบ Radar ซึ่งกรรมวิธีนี้จะทำการตรวจสอบว่าวัตถุที่อยู่ด้านข้างสิ่งกีดขวางนั้นเป็นวัตถุใด พร้อมทั้งรายงานค่าต่างๆ ได้แก่ระยะห่างจากวัตถุในแนวราบ ระยะห่างจากวัตถุในแนวดิ่ง ความกว้างของวัตถุ ทิศทางและค่าของมุมระหว่างหุ่นยนต์กับวัตถุนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างต่อไปนี้จะทำให้เข้าใจหลักการของกรรมวิธีการตรวจสอบสิ่งกีดขวางแบบ Radar มากยิ่งขึ้น



รูปที่ 3.26 การรายงานผลของการตรวจสอบสิ่งกีดขวางแบบเรดาร์

<u>ทางด้านซ้าย</u>

วัตถุที่ตรวจพบ คือ Hole1

ระยะห่างในแนวดิ่ง คือ 100

ระยะห่างในแนวราบ คือ 75

จุดกึ่งกลางด้านของวัตถุที่พบ คือ m1

ความกว้างของวัตถุที่พบ คือ 50

ค่ามุมระหว่างหุ่นยนต์กับวัตถุ คือ θ_1

<u>ทางด้านขวา</u>

วัตถุที่ตรวจพบ คือ Hole2

ระยะห่างในแนวดิ่ง คือ 100

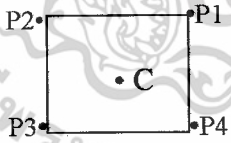
ระยะห่างในแนวราบ คือ 75

จุดกึ่งกลางด้านของวัตถุที่พบ คือ m2

ความกว้างของวัตถุที่พบ คือ 50

ค่ามุมระหว่างหุ่นยนต์กับวัตถุ คือ θ_2

3.5.4. อัลกอริทึมในการตรวจสอบการชน



รูปที่ 3.27 ตำแหน่งที่ใช้ในการตรวจสอบการชน

กรรมวิธีการตรวจสอบการชนจะใช้หลักการโดยนำเอาจุดที่อยู่รอบๆ ตัวหุ่นยนต์ไปทำการตรวจสอบว่าตกอยู่ภายในบริเวณพื้นที่ของวัตถุใดในสนามบ้าง ถ้าไปตกในวัตถุ Hole จะถือว่าไม่ชน แต่ถ้าตกในวัตถุที่ไม่ใช่ Hole จะถือว่าหุ่นยนต์ชน

เนื่องจากจุดที่อยู่รอบตัวหุ่นยนต์รบมีหลายจุด การที่จะนำมาพิจารณาทุกจุดจะทำให้การประมวลผลช้า ผู้พัฒนาจึงเลือกจุดมาใช้ในการพิจารณาการชนเพียง 4 จุดดังที่แสดงให้เห็นในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การใช้งานโปรแกรม

เกม JavaRobot เป็นเกมชนิดออนไลน์ (online) ที่มีลักษณะการเล่นเป็นแบบการต่อสู้ โดยมีสถาปัตยกรรมของระบบเครือข่ายเป็นแบบ 2-Tier Client/Server โดยใช้โปรโตคอล HTTP การทำงานจะแบ่งออกเป็น 2 ฝ่าย คือเซิร์ฟเวอร์และไคล์เอนท์ ซึ่งโปรแกรมของทางฝั่งเซิร์ฟเวอร์จะแบ่งออกเป็น 2 โปรแกรมย่อย ได้แก่ โปรแกรมของผู้พัฒนารุ่นก่อนและโปรแกรมของผู้พัฒนาปัจจุบันซึ่งได้เพิ่มโหมดการเล่นเพิ่มอีก 1 โหมดคือเทรนนิ่งโหมด ส่วนโปรแกรมฝั่งไคล์เอนท์จะมีเพียงโปรแกรมที่ใช้สร้างหุ่นยนต์ การประมวลผลทั้งหมดจะดำเนินการที่เซิร์ฟเวอร์ และส่งผลลัพธ์กลับมาแสดงยังไคล์เอนท์

การเล่นเกม JavaRobot ผู้เล่นจะต้องทำการต่ออินเตอร์เนตและเข้าไปยังเว็บไซต์ http://161.246.60.12/project/Java Robot1.html โดยผู้เล่นที่ต้องการเล่นเป็นครั้งแรกจะต้องทำการสมัครสมาชิกเพื่อให้สามารถสร้างหุ่นยนต์จากโปรแกรม JavaRobot Creater และนำหุ่นยนต์เข้ามาเล่นในเกมได้ สำหรับผู้เล่นที่เคยสมัครสมาชิกแล้วสามารถใส่ชื่อและรหัสผ่านเพื่อเข้าสู่ระบบได้เลย ผู้เล่นต้องเข้าไปยังหน้าอัพโหลด (Upload) เพื่อทำการอัพโหลดหุ่นยนต์ที่สร้างขึ้นจากโปรแกรม JavaRobot Creater ไปเก็บยังเซิร์ฟเวอร์ หรือผู้เล่นสามารถดาวน์โหลดหุ่นยนต์มาแก้ไขและอัพโหลดขึ้นไปเก็บที่เซิร์ฟเวอร์ได้เช่นเดียวกัน เมื่ออัพโหลดหุ่นยนต์เสร็จเรียบร้อยแล้วก็ทำการเลือกเทรนนิ่งโหมด โดยเป็นการต่อสู้ระหว่างหุ่นยนต์ของผู้เล่นกับหุ่นยนต์ที่ถูกกำหนดไว้เพื่อทดสอบการต่อสู้กับหุ่นยนต์ และการหลบหลีกกำแพงและสิ่งกีดขวางต่างๆ ในสนาม โดยรูปแบบสนามจะแตกต่างจากสตอรี่โหมดและไฟต์ติ้งโหมดของผู้พัฒนารุ่นก่อนตรงที่มีกำแพงและสิ่งกีดขวางกั้นอยู่ และมีระดับความยากของการเล่นให้ผู้เล่นเลือกได้ถึง 3 ระดับ ได้แก่ ระดับง่าย (easy) ระดับปานกลาง (normal) และระดับยาก (hard) ในเทรนนิ่งโหมดมีรูปแบบสนามให้เลือกถึง 9 สนาม โดยสนามทั้ง 9 แบบสามารถแบ่งลักษณะออกได้เป็น 2 ประเภทใหญ่ๆ คือสนามทดสอบกรรมวิธีการเดินแบบปกติและสนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ ซึ่งผู้เล่นสามารถเลือกรูปแบบสนามใดก็ได้ จากนั้นก็เลือกระดับที่ต้องการเล่น และสุดท้ายก็ทำการเลือกหุ่นยนต์เพื่อเข้าต่อสู้

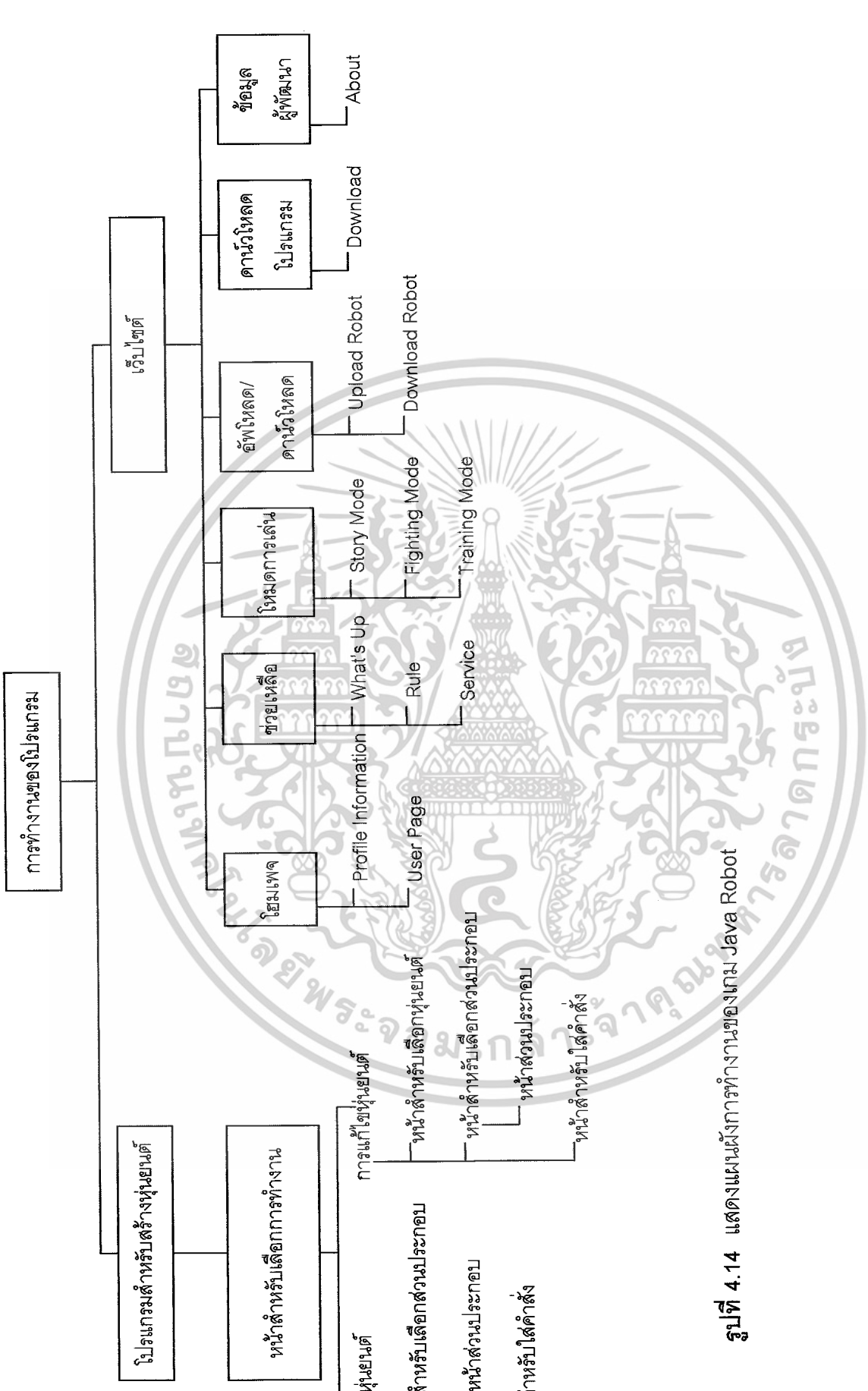
4.1 โครงสร้างของโปรแกรม

เกม JavaRobot ได้แบ่งโปรแกรมออกเป็น 2 แบบตามลักษณะหน้าที่การทำงาน คือ โปรแกรมสำหรับสร้างหุ่นยนต์และเว็บไซต์ โดยโปรแกรมสำหรับสร้างหุ่นยนต์จะเป็นส่วนที่ทำงานบนฝั่งไคล์เอนท์ คือทำงานบนเครื่องของผู้เล่นโดยไม่ต้องทำการต่อกับอินเตอร์เนตก่อนก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับให้ผู้เล่นทำการเลือกส่วนประกอบต่างๆ มาประกอบกันขึ้นเป็นหุ่นยนต์ และให้ผู้เล่นเขียนโปรแกรมเพิ่มเติมเพื่อสั่งให้หุ่นยนต์ทำงานในเหตุการณ์ต่างๆ หรือแก้ไขโปรแกรมเดิมที่มีอยู่ของหุ่นยนต์ และเว็บไซต์ซึ่งเป็นส่วนที่ประมวลผลบนฝั่งเซิร์ฟเวอร์ ซึ่งใช้สำหรับแสดงรายละเอียดต่างๆ ของเกม JavaRobot เช่น การสมัครสมาชิกก่อนที่จะเล่นเกมสำหรับผู้เล่นครั้งแรก การให้ผู้เล่น Login เพื่อเข้าเล่นเกม การดาวน์โหลดโปรแกรมที่จำเป็นที่จะต้องใช้ในการเล่นเกม เช่น โปรแกรม JavaRobot Creater การ Upload Robot เพื่อให้ผู้เล่นส่งหุ่นยนต์ของตนเองที่ต้องการใช้ในการเล่นขึ้นมายังเว็บไซต์ การ Download Robot เพื่อให้ผู้เล่นดาวน์โหลดหุ่นยนต์ไปทำการเปลี่ยนแปลงแก้ไข การทำงานของโปรแกรมสามารถแสดงเป็นแผนผังได้ดังรูปที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงแผนผังการทำงานของเกม Java Robot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1. โปรแกรมสำหรับสร้างหุ่นยนต์ ประกอบด้วย

ตารางที่ 4.1 แสดงรายละเอียดของโปรแกรมสำหรับสร้างหุ่นยนต์

ส่วนประกอบ	**รายละเอียด**
หน้าสำหรับเลือกการทำงาน	เป็นหน้าแรกที่ใช้สำหรับเลือกว่าจะทำการสร้างหุ่นยนต์ใหม่ ทำการปรับปรุงแก้ไขหุ่นยนต์ตัวเดิมที่มีอยู่ หรือทำการออกจากโปรแกรม
หน้าสำหรับเลือกส่วนประกอบ	เป็นหน้าที่ใช้สำหรับเลือกชิ้นส่วนต่างๆ เพื่อมาประกอบเป็นหุ่นยนต์
หน้าสำหรับใส่คำสั่ง	เป็นหน้าที่ใช้สำหรับให้ผู้เล่นเขียนโปรแกรมบังคับหุ่นยนต์เพิ่มเติม
หน้าสำหรับเลือกหุ่นยนต์	เป็นหน้าที่ใช้สำหรับเลือกไฟล์หุ่นยนต์ที่ต้องการนำมาปรับปรุงแก้ไข
หน้าส่วนประกอบ	ชิ้นส่วนแต่ละชนิดจะมีหลายชิ้นเพื่อให้ผู้เล่นทำการเลือกตามความต้องการ

4.1.2 เว็บไซต์เกมจาวาโรบอท ประกอบด้วย

ตารางที่ 4.2 แสดงรายละเอียดของเว็บไซต์

ส่วนประกอบ	**รายละเอียด**
Homepage	เป็นหน้าสำหรับทำการ Login และ Sign Up
Profile Information	เป็นหน้าสำหรับให้ผู้เล่นทำการสมัครสมาชิก
User Page	เป็นหน้าสำหรับบอกรายละเอียดเกี่ยวกับผู้เล่นว่ามีหุ่นยนต์ทั้งหมดกี่ตัว สถิติการแข่งขันเป็นอย่างไร เป็นต้น
What's Up	เป็นหน้าสำหรับอธิบายลักษณะของเกม JavaRobot
Rule	เป็นหน้าสำหรับอธิบายขั้นตอนและกฎในการเล่นเกม Java Robot
Service	เป็นหน้าสำหรับอธิบายว่าเว็บไซต์นี้ให้บริการดาวน์โหลดโปรแกรมใดบ้าง และอธิบายว่า แต่ละโปรแกรมมีความจำเป็นในการเล่นเกม JavaRobot อย่างไร
Story Mode	เป็นหน้าสำหรับให้ผู้เล่นเลือกหุ่นยนต์ของตัวเองเข้าแข่งขัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

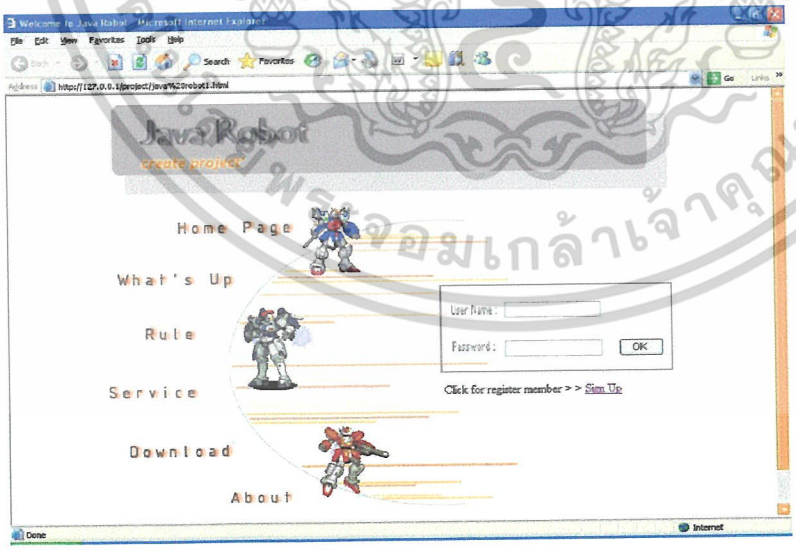
ตารางที่ 4.2(ต่อ)

Fighting Mode	เป็นหน้าสำหรับให้ผู้เล่นเลือกหุ่นยนต์ของตัวเองและหุ่นยนต์ที่ผู้เล่นต้องการสู้ด้วย
Training Mode	เป็นหน้าสำหรับให้ผู้เล่นเลือกสนามแข่งขันโดยมีรูปแบบสนามให้เลือก 9 แบบ ระดับความยากในการเล่นและเลือกหุ่นยนต์ของตัวเองเข้าทำการแข่งขัน
Upload Robot	เป็นหน้าสำหรับให้ผู้เล่นทำการ Upload หุ่นยนต์ของตัวเองไปเก็บที่เซิร์ฟเวอร์และยังให้ผู้เล่นกำหนดหุ่นยนต์นั้นๆ ว่าอนุญาตให้ผู้เล่นคนอื่นเลือกทำการแข่งขันด้วยหรือไม่
Download Robot	เป็นหน้าสำหรับให้ผู้เล่นดาวน์โหลดหุ่นยนต์มาเป็นของตัวเอง
Download	เป็นหน้าสำหรับให้บริการดาวน์โหลดโปรแกรมที่จำเป็นในการเล่นเกม Java Robot
About	เป็นหน้าสำหรับบอกข้อมูลของผู้สร้าง

4.2 ขั้นตอนในการเล่นเกม

4.2.1 การดาวน์โหลดโปรแกรมและการสร้างหุ่นยนต์

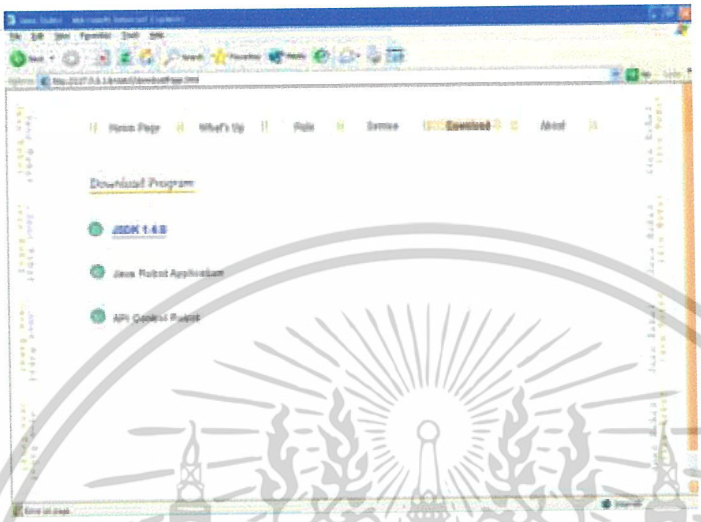
เป็นขั้นตอนแรกสุดของการเล่นเกม JavaRobot โดยเริ่มต้นจากการออนไลน์เข้าสู่เว็บไซต์ของ JavaRobot ดังรูปที่ 4.2



รูปที่ 4.2 แสดงหน้าแรกของเว็บไซต์ JavaRobot

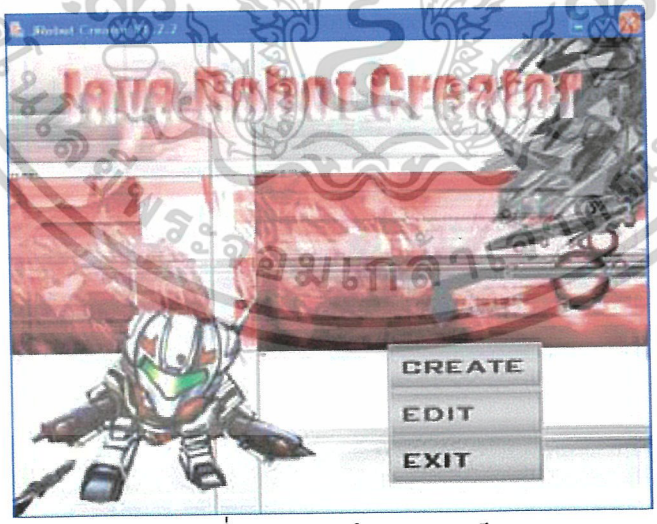
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นคลิก Download เพื่อเข้าสู่หน้าดาวน์โหลด ดังรูปที่ 4.2 ในหน้านี้จะมีโปรแกรมให้ผู้เล่นทำการดาวน์โหลดโปรแกรมที่จำเป็นสำหรับการเล่นเกม ดังนี้ โปรแกรม JDK เป็นคอมไพเลอร์ที่ใช้ในการสร้างหุ่นยนต์ โปรแกรม XML Parser ใช้ในการติดต่อกับไฟล์ XML โปรแกรมในการสร้างหุ่นยนต์ และ API Control Robot เป็นชุดคำสั่งที่ใช้ในการควบคุมหุ่นยนต์



รูปที่ 4.3 แสดงหน้าดาวน์โหลดโปรแกรมสร้างหุ่นยนต์และโปรแกรมที่จำเป็น

เมื่อทำการติดตั้งโปรแกรมที่ดาวน์โหลดมาเรียบร้อยแล้ว ขั้นต้นสำหรับการเล่นเกม ผู้เล่นต้องทำการสร้างหุ่นยนต์ของตัวเองก่อน โดยเปิดโปรแกรม JavaRobot Creater ที่ใช้ในการสร้างหุ่นยนต์ เมื่อเปิดโปรแกรมขึ้นมาจะเป็นดังรูปที่ 4.4



รูปที่ 4.4 แสดงหน้าแรกของโปรแกรมที่ใช้ในการสร้างหุ่นยนต์

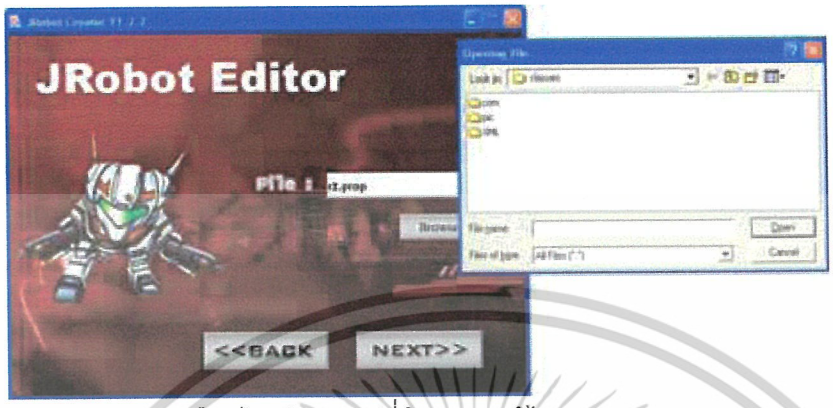
- ผู้เล่นที่ทำการเล่นเป็นครั้งแรกหรือต้องการสร้างหุ่นยนต์ตัวใหม่

ทำการคลิกปุ่ม CREATE เพื่อทำการสร้างหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ผู้เล่นที่เคยสร้างหุ่นยนต์เอาไว้แล้วและต้องการปรับปรุงแก้ไข

ทำการคลิกปุ่ม EDIT เพื่อที่จะทำการปรับปรุงแก้ไขหุ่นยนต์ที่ผู้เล่นได้เคยสร้างไว้แล้ว โดยทำการเลือกไฟล์หุ่นยนต์ที่ต้องการแก้ไข ดังรูปที่ 4.5



รูปที่ 4.5 แสดงหน้าจอการเลือกไฟล์หุ่นยนต์ที่ต้องการแก้ไข

- ผู้เล่นที่ต้องการออกจากโปรแกรม

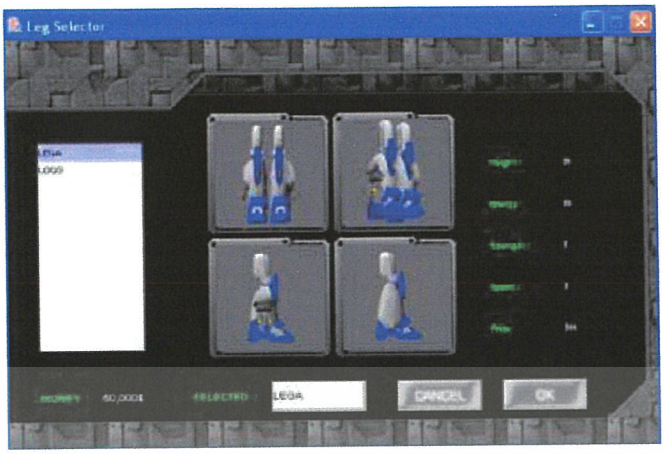
ทำการคลิกปุ่ม EXIT เป็นการออกจากโปรแกรม

จากนั้นทำการตั้งชื่อหุ่นยนต์และเลือกส่วนประกอบดังรูปที่ 4.5

รูปที่ 4.6 แสดงหน้าที่ใช้เลือกส่วนประกอบ

โดยเมื่อทำการเลือกส่วนประกอบจะแสดงหน้าจอที่แสดงถึงส่วนประกอบประเภทนั้นๆ เช่น เมื่อเลือกขาจะแสดงหน้าจอของขาประเภทต่างๆ ให้ผู้เล่นเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

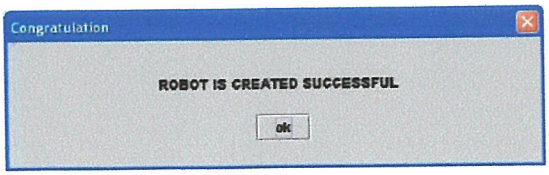


รูปที่ 4.7 แสดงหน้าจอที่ทำการเลือกขา

เมื่อเราทำการเลือกส่วนประกอบแล้ว(กด OK) เงินที่เรามีอยู่จะลดลงโดยอัตโนมัติแต่ถ้าเราไม่ต้องการเลือก(กด CANALE) โปรแกรมจะไม่ทำการเลือกส่วนประกอบนั้น เมื่อผู้เล่นเลือกส่วนประกอบเสร็จแล้วให้คลิกปุ่ม NEXT เพื่อเข้าสู่หน้าที่ใช้เขียนโปรแกรมบังคับหุ่นยนต์เพิ่มเติม ดังรูปที่ 4.8

รูปที่ 4.8 แสดงหน้าที่ใช้สำหรับเขียนโปรแกรมบังคับหุ่นยนต์

เมื่อเขียนโปรแกรมในการบังคับหุ่นยนต์เรียบร้อยแล้วคลิกปุ่ม Generate เพื่อทำการคอมไพล์โปรแกรมที่ผู้เล่นเขียนขึ้น เมื่อคอมไพล์ผ่านแล้วจะปรากฏข้อความขึ้น ดังรูปที่ 4.9



รูปที่ 4.9 แสดง message box หลังจากคอมไพล์โปรแกรมผ่าน

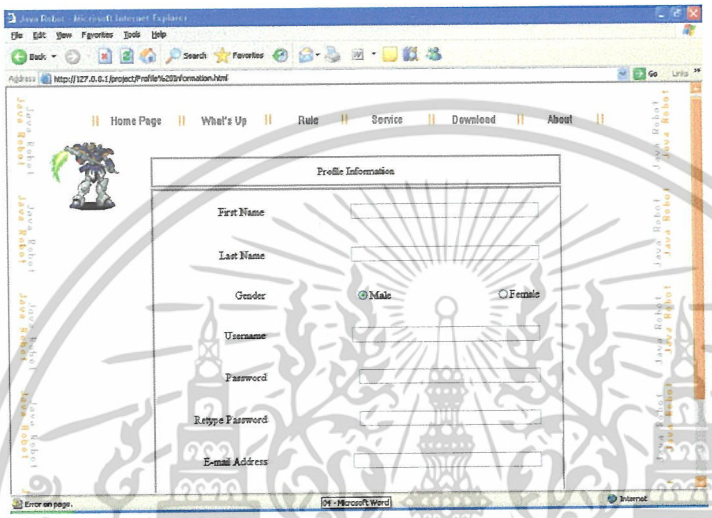
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การเข้าสู่ระบบและการนำหุ่นยนต์ที่สร้างไปใช้งาน

เมื่อสร้างหุ่นยนต์เสร็จแล้วทำการออนไลน์เข้าสู่เว็บไซต์เกม JavaRobot

- ผู้เล่นที่ทำการเล่นเป็นครั้งแรก

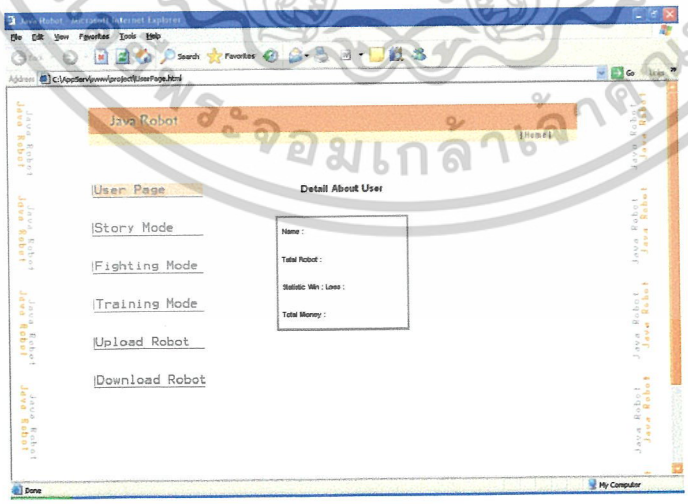
สมัครสมาชิกโดยคลิกที่ Sign Up จะแสดงหน้าที่ให้ใส่ข้อมูลส่วนตัวของผู้เล่น ดังรูปที่ 4.10 โดยผู้เล่นจะต้องทำการกรอกข้อมูลทั้งหมดลงในแบบฟอร์มที่กำหนดไว้ให้ครบถ้วนเพื่อนำ User name และ Password ไปใช้สำหรับการเข้าใช้บริการ



รูปที่ 4.10 แสดงหน้าที่ให้ผู้เล่นทำการใส่ข้อมูลเพื่อสมัครสมาชิก

- ผู้เล่นที่ทำการสมัครสมาชิกแล้ว

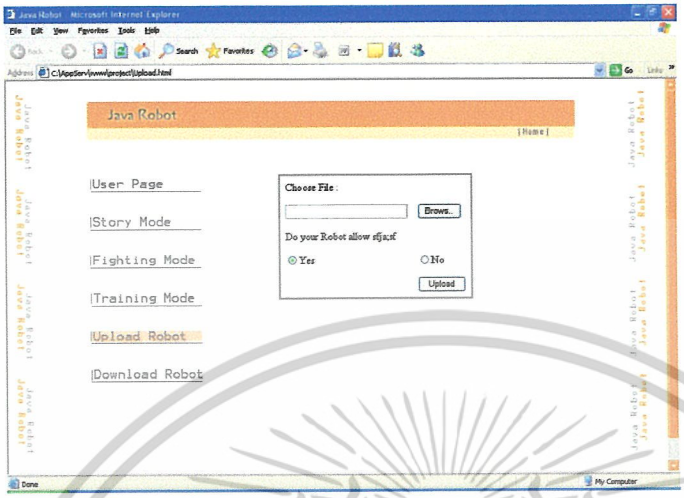
ทำการ Log in เข้าสู่ระบบโดยใส่ Username และ Password ในหน้าแรกของเว็บไซต์ จะเกิดหน้า User Page ซึ่งแสดงข้อมูลของสมาชิก ดังรูปที่ 4.11



รูปที่ 4.11 แสดงหน้าข้อมูลของสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการทำการ Upload หุ่นยนต์ที่สร้างขึ้นแล้วไปเก็บยังเซิร์ฟเวอร์ให้คลิกที่ Upload จะเกิดหน้าที่ให้ผู้เล่นเลือกไฟล์ที่ต้องการนำไปเก็บ ดังรูปที่ 4.12



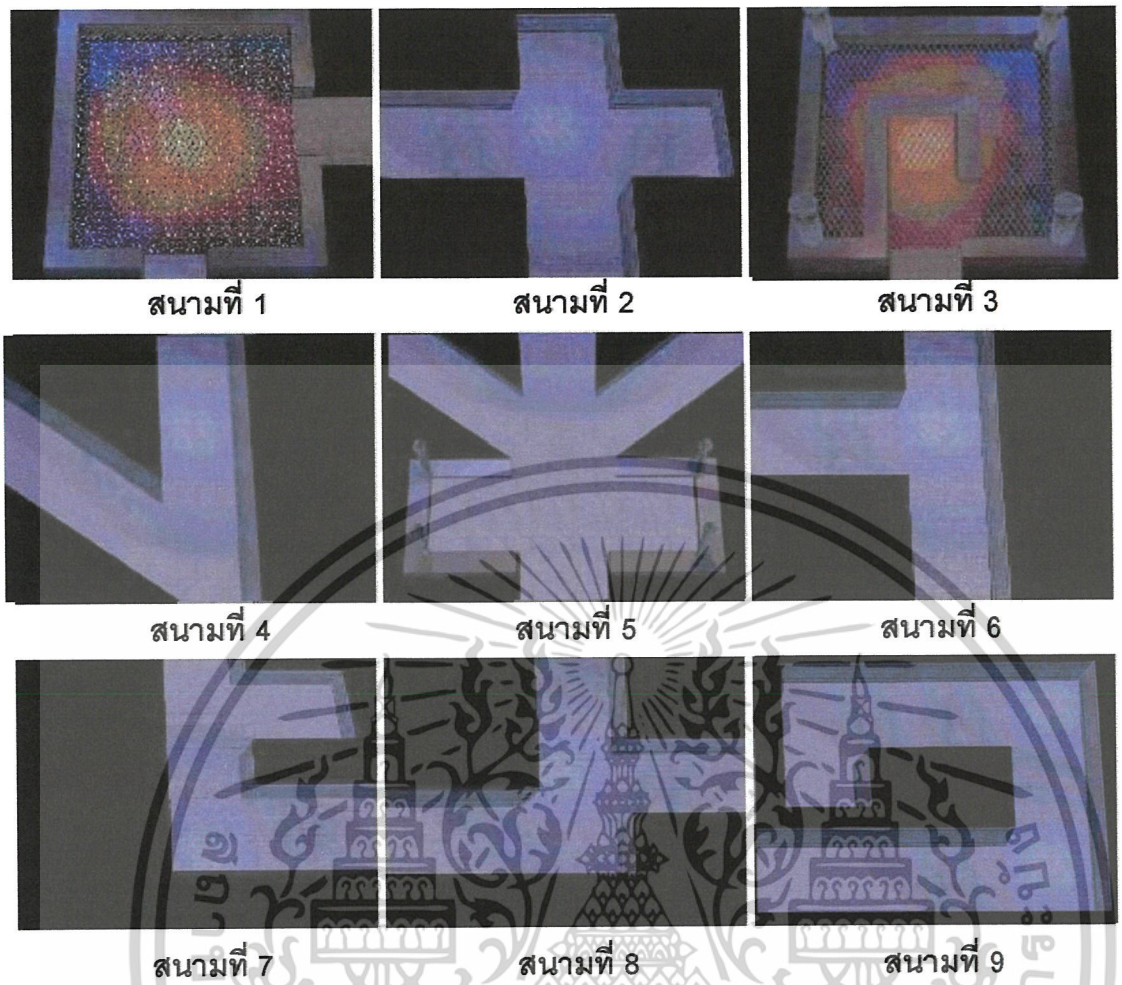
รูปที่ 4.12 แสดงหน้า Upload

เมื่อทำการ Upload หุ่นยนต์เสร็จเรียบร้อยแล้วสามารถเลือกเล่นในโหมดที่ต้องการได้ทันที

4.2.3 การเล่นแบบเทรนนิ่งโหมด

ในการเลือกเล่นแบบเทรนนิ่งโหมดจะเป็นการต่อสู้กับหุ่นยนต์ที่ผู้พัฒนาได้กำหนดเอาไว้ โดยที่ผู้เล่นจะต้องเลือกหุ่นยนต์ของตนเองเพื่อมาทำการต่อสู้กับหุ่นยนต์ศัตรู ซึ่งมีระดับความยากในการต่อสู้ให้ผู้เล่นเลือกได้ 3 ระดับ ได้แก่ ระดับง่าย ระดับปานกลาง และระดับยาก ในการเล่นแบบเทรนนิ่งโหมด สนามต่อสู้จะมีกำแพงและสิ่งกีดขวางกั้นอยู่ด้วยเพื่อให้ผู้เล่นทำการเขียนโปรแกรมควบคุมหุ่นยนต์เมื่อพบกับเหตุการณ์บังคับต่างๆ เช่นเมื่อพบทางแยก ทางที่ต้องการการตัดสินใจ และการค้นหาหุ่นยนต์ศัตรู ผู้เล่นต้องเขียนโปรแกรมจัดการเมื่อพบกับเหตุการณ์ที่ต้องต่อสู้กับหุ่นยนต์ศัตรูด้วย รูปแบบสนามของเทรนนิ่งโหมดมีให้เลือกได้ทั้งหมด 9 รูปแบบ ดังรูปที่ 4.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

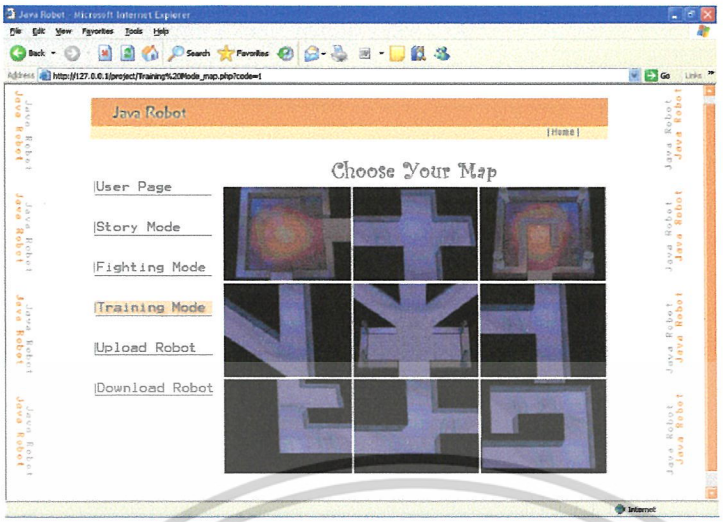


รูปที่ 4.13 รูปแบบสนาม 9 แบบในเทรนนิ่งโหมด

โดยสนามต่อสู้ทั้ง 9 แบบจะแบ่งออกตามลักษณะของสนามได้ 2 ประเภท คือสนามทดสอบกรรมวิธีการเดินแบบปกติ ได้แก่สนามที่ 1, 4 และ 9 ส่วนสนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ ได้แก่สนามที่ 2, 3, 5, 6, 7 และ 8

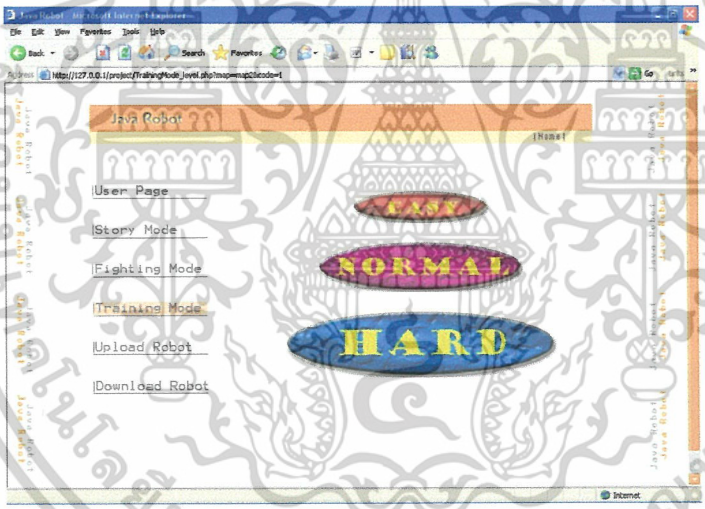
ผู้เล่นสามารถเลือกโหมดการเล่นแบบเทรนนิ่งโหมดได้โดยการคลิกเข้าไปในหน้า Training Mode จะปรากฏหน้าสำหรับให้ผู้เล่นเลือกสนามที่ต้องการเข้าต่อสู้ ดังรูปที่ 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงหน้าให้ผู้เล่นเลือกสนาม

เมื่อผู้เล่นทำการเลือกสนามโดยการคลิกที่รูปภาพของสนามนั้นๆ แล้วจะปรากฏหน้าสำหรับให้ผู้เล่นเลือกระดับความยากง่ายในการต่อสู้กับหุ่นยนต์ศัตรู ดังรูปที่ 4.15



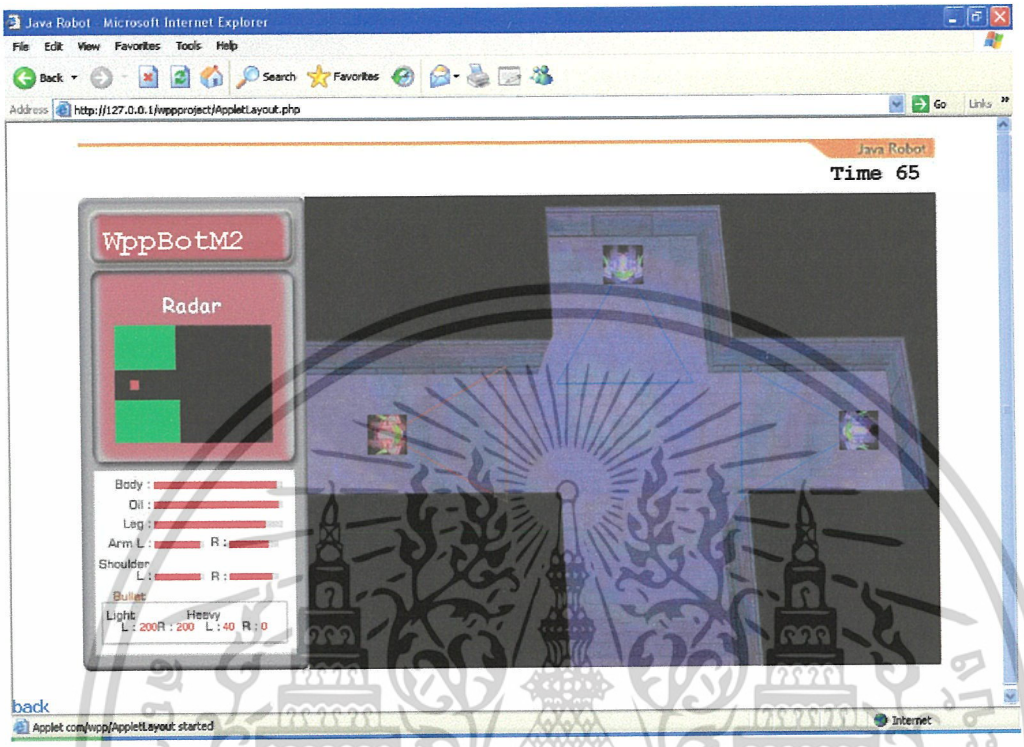
รูปที่ 4.15 แสดงหน้าให้ผู้เล่นเลือกระดับการเล่น

โดยระดับง่าย (easy) หุ่นยนต์ศัตรูจะยืนอยู่เฉยๆ ไม่มีการเคลื่อนไหวและไม่มีการยิงโต้ตอบกับผู้เล่น ระดับปานกลาง (normal) หุ่นยนต์ศัตรูหนึ่งตัวจะอยู่กับที่ และหุ่นยนต์ศัตรูอีกหนึ่งตัวจะมีการเดินเคลื่อนที่ไปมา โดยหุ่นยนต์ทั้ง 2 ตัวจะมีการยิงโต้ตอบกับผู้เล่นด้วย และระดับยาก (hard) หุ่นยนต์ศัตรูทั้ง 2 ตัวจะเคลื่อนที่มีการยิงโต้ตอบกับผู้เล่น

เมื่อผู้เล่นทำการเลือกระดับความยากในการเล่นแล้วจะปรากฏหน้าสำหรับให้ผู้เล่นเลือกหุ่นยนต์ที่ต้องการนำเข้าต่อสู้ ดังรูปที่ 4.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และตัวอย่างหน้าแสดงผลการต่อสู้ของสนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ ดังรูปที่ 4.18



รูปที่ 4.18 แสดงตัวอย่างหน้าแสดงผลการต่อสู้ สนามทดสอบกรรมวิธีการเดินแบบตัดสินใจ

หน้าแสดงผลการต่อสู้จะแบ่งออกเป็น 4 ส่วน คือ

1) ส่วนแสดงชื่อของหุ่นยนต์ ใช้แสดงชื่อหุ่นยนต์ของผู้เล่นที่นำเข้ามาต่อสู้

2) ส่วนแสดงผลการสแกน ใช้แสดงผลของการสแกนที่หุ่นยนต์ผู้เล่นทำการสแกนพื้นที่สนามต่อสู้ โดยสิ่งที่แสดงผลออกมาคือขอบสนาม สิ่งกีดขวางและหุ่นยนต์ศัตรู

3) ส่วนแสดงค่าสถานะของหุ่นยนต์ ใช้แสดงค่าสถานะต่างๆ ของหุ่นยนต์ผู้เล่น ได้แก่ ค่าพลังงานของร่างกาย น้ำมัน ค่าพลังงานขา ค่าพลังงานแขน ค่าพลังงานของไหล่ ที่แสดงอยู่ในรูปของแถบสีแดงซึ่งจะลดลงเรื่อยๆ เมื่อมีการเดินหรือโดนหุ่นยนต์ศัตรูโจมตี และค่าของจำนวนกระสุนที่เหลืออยู่ซึ่งจะแสดงด้วยตัวเลข โดยค่าของจำนวนกระสุนจะลดลงเรื่อยๆ เมื่อหุ่นยนต์ยิงกระสุนออกไป

4) ส่วนแสดงการต่อสู้ ใช้แสดงการต่อสู้ของหุ่นยนต์และการหลบหลีกสิ่งกีดขวางภายในสนามที่ผู้เล่นได้เลือกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการศึกษา ข้อเสนอแนะ และแนวทางการพัฒนาต่อ

5.1 สรุปผลปัญหาพิเศษ

1) เพิ่มลักษณะการเล่นที่เรียกว่า Training Mode ซึ่งในส่วนของ Training Mode นั้นได้ทำการสร้างสนามไว้ 9 สนาม โดยลักษณะสนามในแต่ละสนาม จะประกอบด้วยสิ่งกีดขวางในรูปแบบต่างๆ นอกจากนั้นยังได้ทำการสร้างให้แต่ละสนามนั้นมีระดับความยากง่ายในการเล่นอีกสนามละ 3 ระดับ

2) ทำการพัฒนากรรมวิธีในการตรวจสอบสิ่งกีดขวาง แบบสแกน ของหุ่นยนต์ ให้สามารถทำการรายงานผลการสแกน ได้ 3 ทิศทาง คือ ด้านซ้าย ด้านหน้า และด้านขวา พร้อมทั้งรายงานว่าวัตถุที่พบนั้นเป็นวัตถุชนิดใด

3) ทำการพัฒนากรรมวิธีในการตรวจสอบสิ่งกีดขวางแบบ Radar ของหุ่นยนต์ ให้สามารถทำการตรวจหาช่องทางเดิน ที่หุ่นยนต์สามารถจะเดินผ่านไปได้ พร้อมทั้งรายงานข้อมูลต่างๆ ที่จำเป็นต่อการตัดสินใจของหุ่นยนต์ ได้แก่ ทิศทางของทางเดินที่พบ, ขนาดของทางเดิน, ระยะห่างในแนวราบ, ระยะห่างในแนวดิ่ง, ระยะห่างในแนวทแยง และจุดกึ่งกลางของช่องทางเดิน

5.2 ปัญหาและข้อจำกัด

1) การเล่นเกม Java Robot Online ผู้เล่นควรจะมีระบบเครือข่ายที่มีความเร็วในระดับที่ดี เนื่องจากต้องส่งผ่านข้อมูลผ่านระบบเครือข่าย รวมไปถึงประสิทธิ์ภาพของคอมพิวเตอร์ที่นำมาใช้เล่นด้วย ถ้าคอมพิวเตอร์ที่มีการประมวลผลช้า เกมจะมีการทำงานได้ช้าขาดความราบรื่นในการเล่นเกม

2) ผู้เล่นจะมีพื้นฐานในการเขียนโปรแกรมภาษาเชิงวัตถุมาก่อน

3) ในขณะที่หุ่นยนต์ต่อสู้ในสนาม ผู้เล่นไม่สามารถทราบพลังงานของหุ่นยนต์ฝ่ายศัตรูได้

4) ระบบปฏิบัติการที่ใช้จะต้องเป็นระบบปฏิบัติการในตระกูล Windows และควรใช้ Windows98 เป็นอย่างต่ำ

5) ควรใช้ ตัวแปลภาษาจาวา รุ่น J2SDK 1.4.1_01 เป็นอย่างต่ำ

5.3 ข้อเสนอแนะและแนวทางพัฒนาต่อ

1) กราฟฟิกในปัจจุบันถูกแสดงในแบบ 2 มิติ ซึ่งหากทำการพัฒนาต่อควรพัฒนาให้การแสดงผลออกมาในรูปแบบ 3 มิติ และพัฒนาการแสดงผลของรูปหุ่นยนต์ให้มีความชัดเจนมากขึ้น เช่น การขยับแขน ขา ลักษณะการเดิน และการยกปืนขึ้นเพื่อยิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) สามารถพัฒนาการแสดงผลของรูปหุ่นยนต์ให้มีความชัดเจนมากขึ้น เช่น การขยับแขน ขา ลักษณะการเดิน และการยกปืนขึ้นเพื่อยิง

3) ในการต่อสู้แบบเทรนนิ่งโหมด มีสนามที่ใช้ในการต่อสู้ให้เลือกได้ 9 รูปแบบ การพัฒนาต่ออาจเป็นการเชื่อมต่อเส้นทางเดินระหว่างรูปแบบสนามต่อสู้ทั้ง 9 แบบให้สามารถเดินผ่านระหว่างสนามที่ติดต่อกันได้ เสมือนทั้ง 9 สนามต่อสู้เป็นสนามใหญ่ 1 สนามที่แบ่งย่อยออกเป็นสนามย่อย 9 สนาม

4) ควรปรับปรุงให้สนามทำการเลือน เมื่อหุ่นยนต์เดินไปสุดขอบสนามแล้ว

5) ปัจจุบันในสนามสามารถนำหุ่นยนต์ของผู้เล่นเข้าต่อสู้ได้ครั้งละ 1 ตัวเท่านั้น การพัฒนาสามารถเพิ่มในส่วนที่ทำให้หุ่นยนต์ของผู้เล่นสามารถเข้าต่อสู้เป็นทีมได้ อาจเป็นทีมละ 2 ตัว หรือว่าจะสามารถเข้าต่อสู้ทีละหลายๆ ตัวได้พร้อมกัน

6) สามารถเพิ่มในส่วนของสิ่งกีดขวาง เช่น มีบริเวณที่จะป้องกันการสแกนของศัตรู มีบริเวณที่มีไฟฟ้าที่หากเข้าไปในบริเวณนั้นๆ จะเสียพลัง มีสิ่งของให้เก็บเพื่อเพิ่มพลังชีวิตหรือเพิ่มพลังงาน หรือมีสิ่งของที่สามารถป้องกันอาวุธของศัตรูได้ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

นางสาวชนณิกานต์ ธรรมบริสุทธิ์ และคณะ. 2545. "การพัฒนาโปรแกรมเกมหุ่นยนต์รบที่ควบคุมโดยภาษาจาวา." วิทยานิพนธ์วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ดร.วีระศักดิ์ ซึงถาวร, Java Programming Volume I, บริษัท ซีเอ็ดยูเคชั่น จำกัด, กรุงเทพ , พ.ศ. 2545.

ดร.วีระศักดิ์ ซึงถาวร, Java Programming Volume II, บริษัท ซีเอ็ดยูเคชั่น จำกัด, กรุงเทพ , พ.ศ. 2545.

E. Reed Doke, John W.Satzinger ,and Susan Rebstock Williams, Object Oriented Application Development using Java™, Course Technology, USA., 2002

กรภัทร์ สุทธิดารา และอรรณพ ขันธิกุล, **คู่มือการใช้** Photoshop 5 Visual Guide & Step by Step, ด่านสุทธาการพิมพ์ จำกัด, กรุงเทพ .

นิรุธ อำนวยศิลป์, **สร้างเว็บเพจอย่างไร้ขีดจำกัด** PHP **เพื่อการประยุกต์ใช้งาน**, พิมพ์ครั้งที่ 2, บริษัท ซัคเซส มีเดีย จำกัด, กรุงเทพ.

ภัทธิรา เหลืองวิลาศ, DREAMWEAVER MX **สร้างเว็บไซต์แบบมืออาชีพ**, บริษัท ซีเอ็ดยูเคชั่น จำกัด, กรุงเทพ , พ.ศ. 2545.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
การติดตั้งโปรแกรม

การติดตั้ง J2SDK โดยในที่นี้เป็นการขั้นตอนการติดตั้ง j2sdk 1.4.0

1. ดับเบิ้ลคลิกที่ตัวโปรแกรม j2sdk-1_4_0-win.exeโปรแกรม JDK จะถูก Install

2. Set Class path เพื่อให้สามารถเรียกใช้ Javac.exe และ Java.exe ได้จากในไดเรกทอรีใดๆ โดยพิมพ์ค่าเหล่านี้ลง ไปในไฟล์ autoexec.bat "SET PATH= C:\JDK1.2.2\BIN"

3. เมื่อเราต้องการที่จะ Compile โปรแกรมที่อ้างถึงคลาสต่างๆ จะทำได้โดยเพิ่มประโยค "SET CLASSPATH=.;C:\JDK1.2.2\LIB\CLASSES.ZIP" ที่ ไฟล์ autoexec.bat

4. ปกติเราจะใช้ default class path โดยจะ กำหนด ดังนี้ คือ ".;$JAVA\CLASSES;$JAVA\LIB\CLASSES.ZIP" ไฟล์ autoexec.bat

5. สำหรับที่เครื่องลูกข่ายให้Set Class path เพิ่มโดย Set Class path ของ Directory ที่ลงโปรแกรม Java Robot Creator ไว้ด้วย
เช่น "SET CLASSPATH=.;C:\JDK1.2.2\LIB\CLASSES.ZIP;C:\JavaRobotCreator"

การติดตั้งโปรแกรมสร้างหุ่นยนต์ Java Robot Creator

1. ดาวน์โหลดโปรแกรมจาก web site

2. Extract File JavaRobotCreator.zip ลงใน Directory ที่ต้องการ

3. โปรแกรมจะถูกเปิดขึ้น ซึ่งจะสามารถสร้างหุ่นยนต์ได้

การติดตั้งโปรแกรม AppServ และ JavaRobot บน Server

1. ดับเบิ้ลคลิกที่ตัวโปรแกรม appserv-win32-1.4.0.exe โปรแกรม AppServ จะถูก Install

2. นำ Folder project , robot ,ServerSide ที่อยู่ใน Folder Install\web ไปไว้ใน folder ที่ติดตั้งโปรแกรม AppServ โดยใส่ไว้ใน Subfolder www

3. นำ Folder javarobot ที่อยู่ใน Folder Database ไปไว้ใน folder ที่ติดตั้งโปรแกรม AppServ โดยใส่ไว้ใน Subfolder mysql\data

4. run File ServerSide.class ที่อยู่ใน Subfolder www\ServerSide

5. กำหนด IP ของ server file Install\web\robot\JField.java และใน Install\web\project\JavaApplet.php และ Install\web\project\JavaApplet1.php

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.
คู่มือการใช้โปรแกรม

ความต้องการทางด้านฮาร์ดแวร์

ความต้องการทางด้านฮาร์ดแวร์อย่างต่ำ

- CPU :Pentium 233
- RAM : 32 MB
- Display : ขนาด 4 MB
- Sound card 16 Bit
- Modem สำหรับติดต่อกับอินเตอร์เนต
- ความละเอียดของหน้าจอ 800 x 600

ความต้องการทางด้านฮาร์ดแวร์ที่แนะนำ

- CPU : Pentium III 450
- RAM : 64 MB
- Display : ขนาด 16 MB
- Sound card 16 Bit
- Modem 56 Kbps
- ความละเอียดของหน้าจอ 1024 x 768

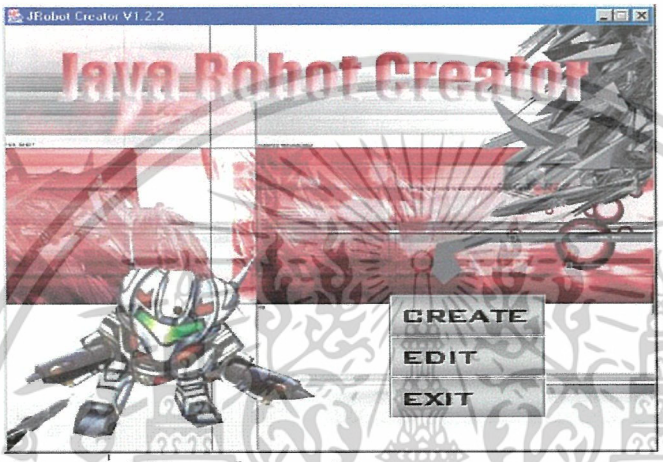
ความต้องการทางด้านซอฟท์แวร์

- OS: Windows 95, Windows 98 Windows Me, Windows NT 4.0 หรือ Windows 2000 หรือสูงกว่านี้
- Browser: เช่น Internet Explorer 6 หรือสูงกว่านี้
- JDK 1.4.1_01 หรือรุ่นที่สูงกว่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้แอปพลิเคชัน

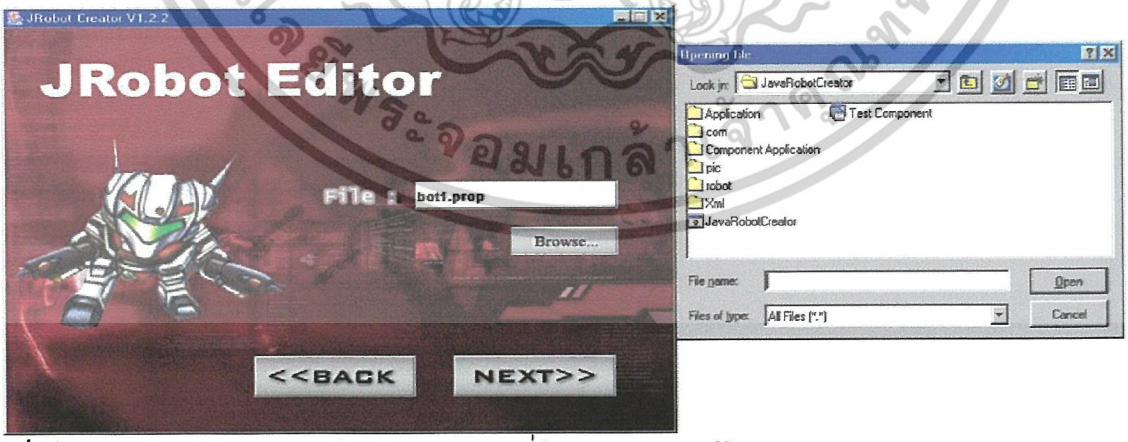
แอปพลิเคชันเป็นส่วนโปรแกรมที่ใช้สำหรับสร้างหุ่นยนต์ โดยสามารถดาวน์โหลดโปรแกรมได้จากเว็บไซท์ JavaRobot เมื่อทำการติดตั้งโปรแกรมเสร็จเรียบร้อยแล้ว ทำการเปิดโปรแกรมขึ้นมาจะเป็นดังรูปที่ ข-1 ให้ผู้เล่นทำการเลือกการทำงาน



รูปที่ ข-1 แสดงหน้าแรกเพื่อให้ผู้เล่นเลือกการทำงาน

- CREATE เป็นการเลือกเพื่อทำการสร้างหุ่นยนต์ตัวใหม่หรือสำหรับผู้ที่ไม่เคยมีหุ่นยนต์ของตัวเองมาก่อนโดยจะเชื่อมต่อกับหน้าเลือกส่วนประกอบ

- EDIT เป็นการเลือกเพื่อให้ผู้เล่นที่มีหุ่นยนต์แล้วทำการปรับปรุงแก้ไขเพิ่มเติม โดยเมื่อกดแล้วจะเป็นดังรูปที่ ข-2 เพื่อให้ผู้เล่นเลือกไฟล์หุ่นยนต์ที่ต้องการแก้ไขโดยกดปุ่ม Browse

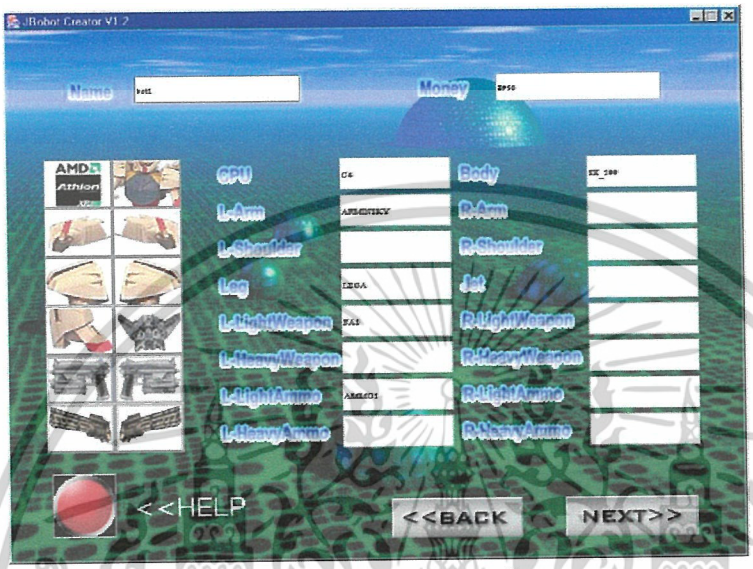


รูปที่ ข-2 แสดงหน้าจอในการเลือกไฟล์หุ่นยนต์ที่ต้องการนำมาแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- EXIT เป็นการเลือกเพื่อออกจากโปรแกรม

เมื่อทำการเลือกหุ่นยนต์ตัวใหม่หรือเลือกไฟล์หุ่นยนต์ที่ต้องการแก้ไขแล้ว จะเกิดหน้าจอดังรูปที่ ข-3 เพื่อให้ทำการตั้งชื่อหุ่นยนต์และเลือกส่วนประกอบ



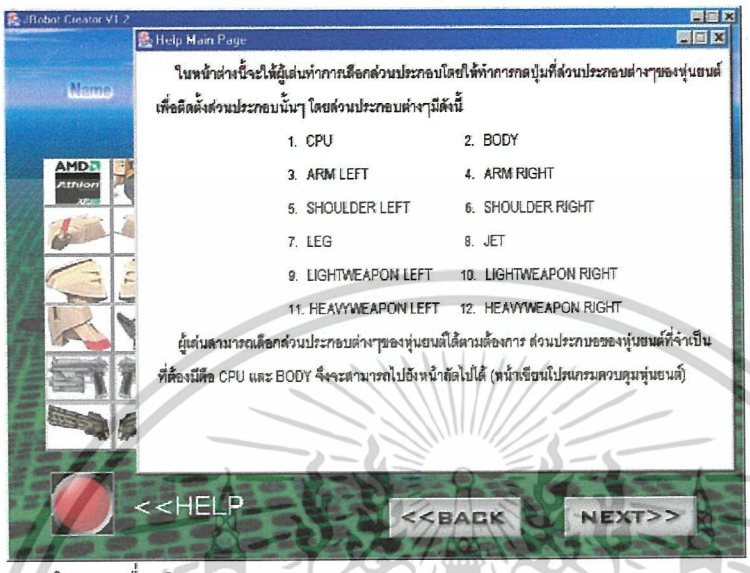
รูปที่ ข-3 แสดงหน้าจอในการเลือกส่วนประกอบของหุ่นยนต์

ส่วนประกอบของหุ่นยนต์ในแต่ละชิ้นส่วนจะมีให้เลือกหลายชนิดแตกต่างกันไปตามระดับชั้นของหุ่นยนต์ โดยถ้ายิ่งระดับชั้นสูงก็จะมีให้เลือกมากและมีความสามารถสูง โดยเมื่อทำการกดที่ชิ้นส่วนใดๆก็ตามจะเกิดหน้าจอของอุปกรณ์ชิ้นนั้นๆ เช่น รูปที่ ข-4 แสดงหน้าจอที่ทำการเลือกลำตัว

รูปที่ ข-4 แสดงหน้าจอที่ทำการเลือกลำตัว

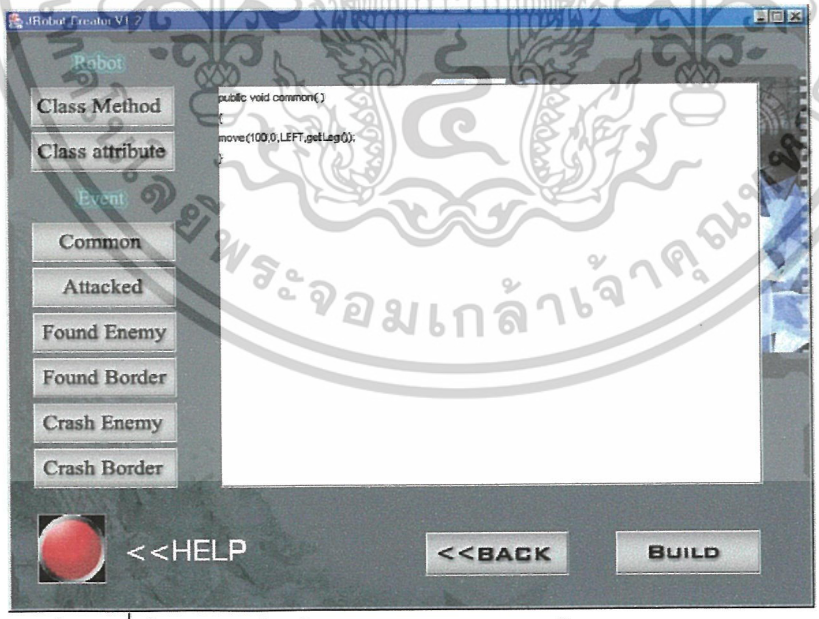
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าไม่เข้าใจการทำงานสามารถกด HELP เพื่อขอคำอธิบายได้ดังรูปที่ ข-5



รูปที่ ข-5 แสดงหน้าจอเมื่อทำการกด HELP

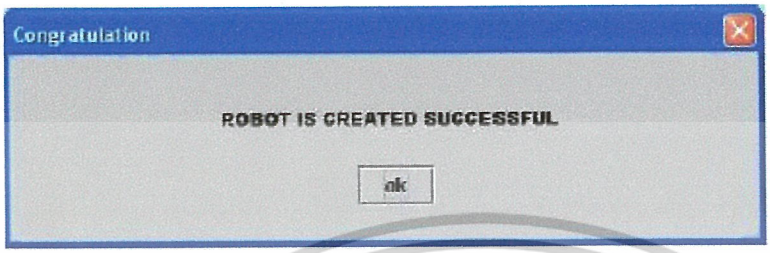
เมื่อเลือกชิ้นส่วนที่ต้องการครบแล้วทำการกด NEXT เพื่อไปยังการทำงานในส่วนถัดไป ถ้าต้องการกลับไปยังส่วนก่อนหน้าให้ทำการกดปุ่ม BACK รูปที่ ข-6 เป็นหน้าที่ใช้สำหรับเขียนโปรแกรมบังคับหุ่นยนต์ให้ทำงานในแต่ละเหตุการณ์ว่าเหตุการณ์ใดต้องทำงานอะไร



รูปที่ ข-6 แสดงหน้าจอที่ใช้สำหรับเขียนโปรแกรมบังคับหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

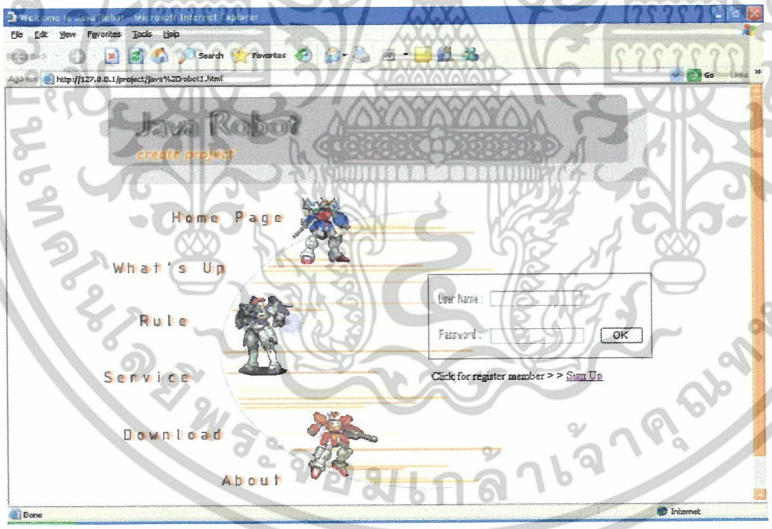
เมื่อเขียนโปรแกรมบังคับหุ่นยนต์เสร็จแล้วทำการกดปุ่ม BUILD เพื่อทำการคอมไพล์โปรแกรมที่ผู้เล่นเขียนขึ้นว่าถูกต้องหรือไม่ ถ้าถูกต้องจะเป็นดังรูปที่ ข-7



รูปที่ ข-7 แสดงหน้าจอเมื่อคอมไพล์โปรแกรมที่ผู้เล่นเขียนขึ้นผ่าน

คู่มือการใช้เว็บ

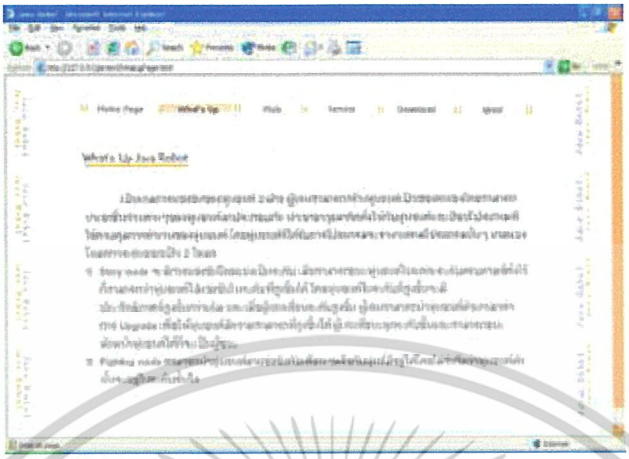
หน้าแรกของเว็บ JavaRobot จะให้ผู้เล่นทำการสมัครสมาชิก หรือถ้าเป็นสมาชิกแล้วให้ทำการเข้าสู่ระบบเพื่อไปใช้งาน หรือถ้าต้องการดูว่าเป็นเกมอะไร วิธีการเล่น บริการที่มีในเว็บ โปรแกรมที่จำเป็นสำหรับการสร้างหุ่นยนต์และคอมไพล์ และเกี่ยวกับผู้สร้าง



รูปที่ ข-8 แสดงหน้าแรกของเว็บ JavaRobot

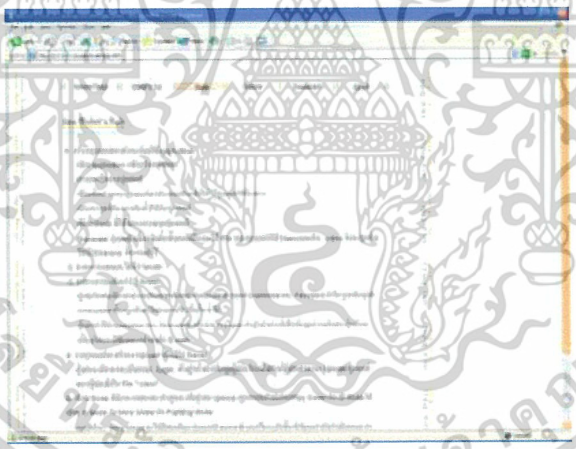
หน้า What's Up เป็นหน้าที่แสดงให้ผู้เล่นรู้ว่าเป็นเกมเกี่ยวกับอะไร การเล่นเป็นอย่างไร สามารถเข้าไปดูได้ ดังรูปที่ ข-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-9 แสดงหน้า What's Up

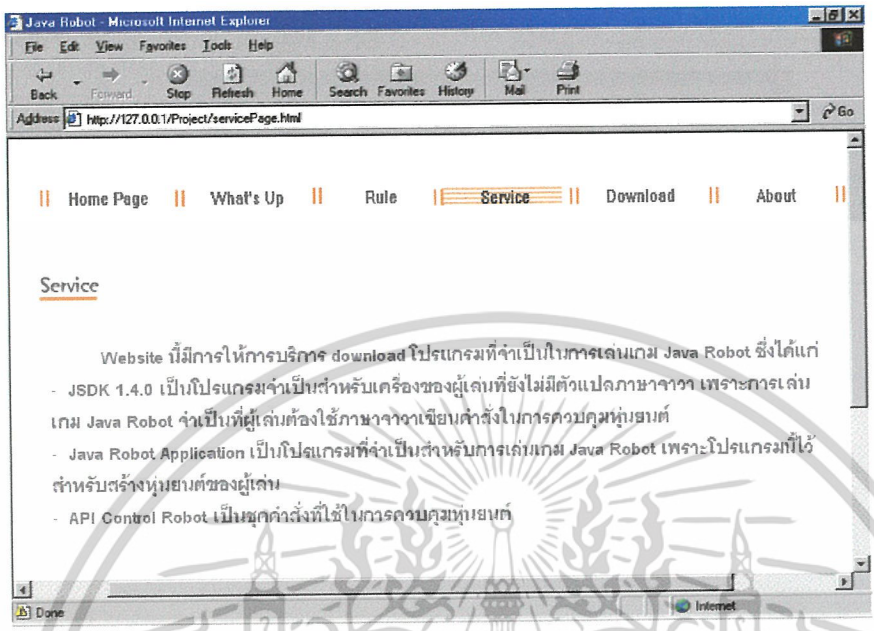
หน้า Rule เป็นหน้าที่แสดงให้ผู้เล่นรู้เกี่ยวกับวิธีการเล่นว่าขั้นตอนในการเล่นต้องทำอย่างไรบ้าง ตั้งแต่การสร้างหุ่นยนต์ การนำไปใช้งาน และโหมดการเล่นมีกี่แบบ แต่ละแบบเป็นอย่างไร สามารถเข้าไปดูได้ ดังรูปที่ ข-10



รูปที่ ข-10 แสดงหน้า Rule

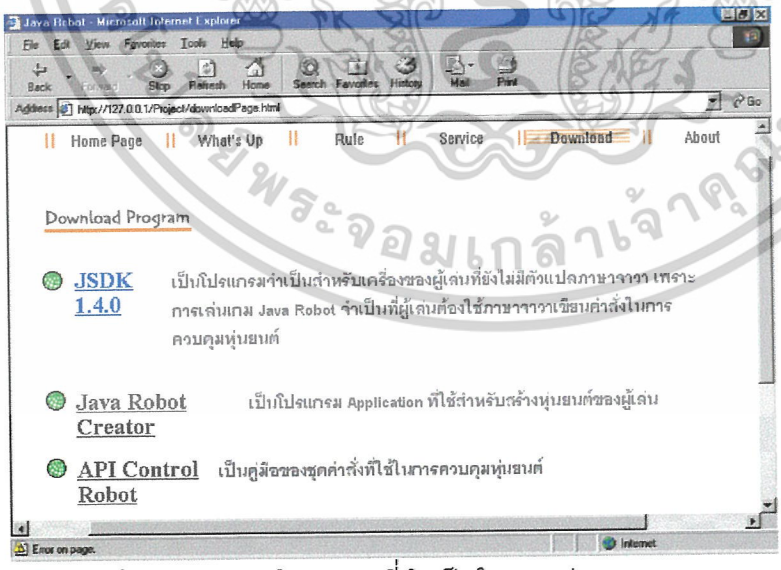
หน้า Service เป็นหน้าที่แสดงให้ผู้เล่นทราบเกี่ยวกับบริการที่มีให้ภายในเว็บนี้ เช่น อธิบายว่าโปรแกรมที่ให้ดาวน์โหลด เอาไว้ใช้งานอะไร ดังรูปที่ ข-11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



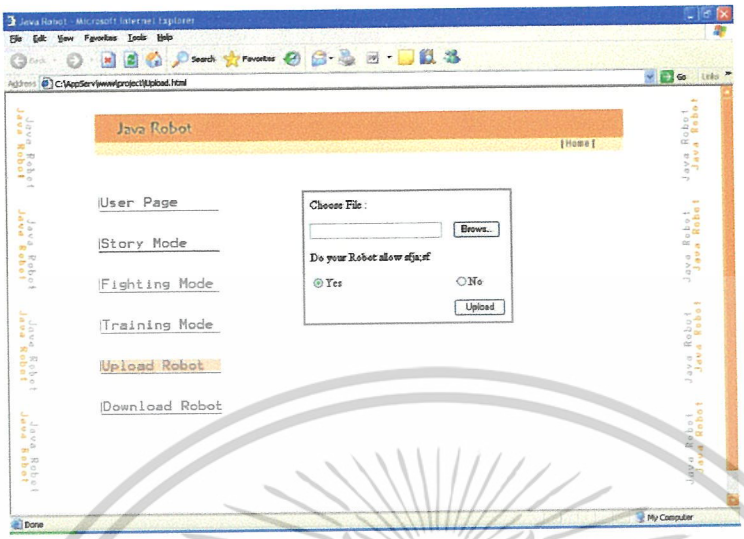
รูปที่ ข-11 แสดงหน้า Service

หน้า Download เป็นหน้าที่แสดงโปรแกรมที่จำเป็นในการเล่นเกม โดยผู้เล่นที่สนใจเกี่ยวกับเกมนี้สามารถดาวน์โหลดโปรแกรมที่ใช้สำหรับสร้างหุ่นยนต์ โปรแกรมที่ใช้สำหรับสร้างหุ่นยนต์ โปรแกรมที่ใช้อ่านไฟล์ XML และเอกสารประกอบเพื่อนำไปสร้างหุ่นยนต์และนำมาใช้เล่นในเว็บนี้ได้ ดังรูปที่ ข-12



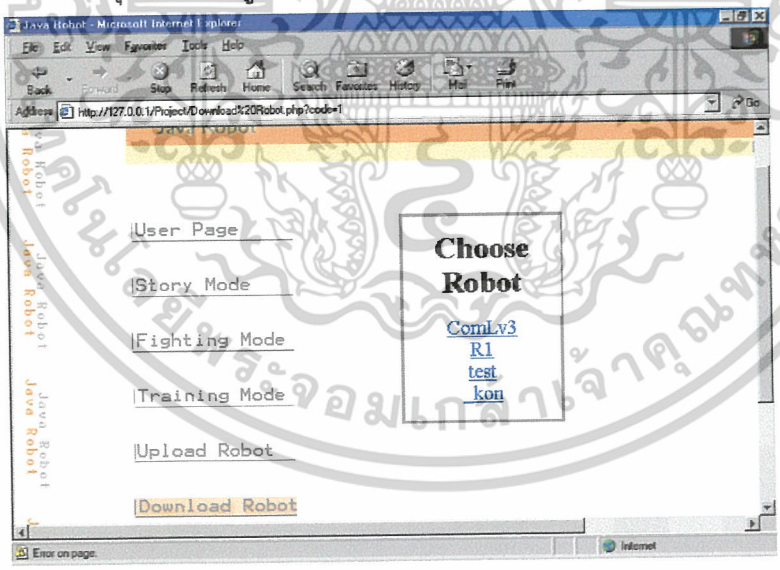
รูปที่ ข-12 แสดงหน้าดาวน์โหลดโปรแกรมที่จำเป็นในการเล่นเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-17 แสดงหน้า Upload

หน้า Download เป็นหน้าที่ให้ผู้เล่นทำการดาวน์โหลดหุ่นยนต์ของตนเองที่เก็บไว้ที่เซิร์ฟเวอร์มาทำการปรับปรุงแก้ไขทั้งในส่วนของส่วนประกอบและส่วนของการเขียนโปรแกรมบังคับหุ่นยนต์ โดยจะต้องทำการดาวน์โหลดหุ่นยนต์ตัวนั้นมาไว้ที่เครื่องของผู้เล่นก่อน แล้วจึงใช้โปรแกรมในการสร้างหุ่นยนต์ในการปรับปรุงแก้ไข ดังรูปที่ ข-18



รูปที่ ข-18 แสดงหน้าดาวน์โหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้