

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เกมแรลลี่ทั่วไทยไปกับอันซีนอินไทยแลนด์

UNSEEN-IN-THAILAND RALLY GAME



ข.พ.  
๐๖ 3697  
๘549

เลขหมู่.....  
เลขทะเบียน..... 73336  
วัน,เดือน,ปี 12 ก.ค. 2550

b. 1179023A  
i.....

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต  
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์  
สาขาคณิตศาสตร์ประยุกต์  
คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# UNSEEN-IN-THAILAND RALLY GAME



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 2006



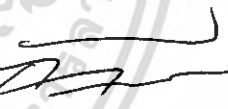
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ เกมแรลลี่ทั่วไทยไปกับอันซีนอินไทยแลนด์  
UNSEEN IN THAILAND RALLY

ชื่อนักศึกษา นายณัฐวุฒิ เหลืองอรุณอุทัย 46050013  
นายวิษณุ ต๋อวงศ์ไพชยนต์ 46050032

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์  
สาขาวิชา คณิตศาสตร์ประยุกต์  
อาจารย์ที่ปรึกษา ผศ.ดร.จรัสชัย ลีนาวงศ์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาคณิตศาสตร์ประยุกต์ ประจำปีการศึกษา 2549

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	อ.พรชัย ชัยสนิทธิ	
กรรมการ	ดร.สิริพร หงษ์ทอง	
กรรมการและอาจารย์ที่ปรึกษา	ผศ.ดร.จรัสชัย ลีนาวงศ์	

(รองศาสตราจารย์ ดร.วีระ บุญจริง)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	เกมแรลลีทั่วไทยไปกับอันซีนอินไทยแลนด์
ชื่อนักศึกษา	นายณัฐวุฒิ เหลืองอรุณอุทัย 46050013 นายวิษญะ ต่ดวงศ์ไพชยนต์ 46050032
ปริญญา	วิทยาศาสตร์บัณฑิต
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สาขาวิชา	คณิตศาสตร์ประยุกต์
ปีการศึกษา	2549
อาจารย์ที่ปรึกษา	ผศ.ดร.จรัสชัย ลีนาวงศ์

### บทคัดย่อ

เกมแรลลีทั่วไทยไปกับอันซีนอินไทยแลนด์นี้ มีจุดประสงค์เพื่อศึกษาและนำระบบปัญญาประดิษฐ์ไปประยุกต์ใช้ เพื่อควบคุมตัวละครที่ผู้เล่นไม่ได้ควบคุม ในรูปแบบของเกมกระดานคอมพิวเตอร์ ทั้งการเลือกเส้นทางเดินของตัวละครโดยใช้วิธี Path finding รวมถึงการตัดสินใจของตัวละครโดยใช้ตรรกศาสตร์ Fuzzy และความน่าจะเป็น เพื่อให้ตัวละครมีการตัดสินใจที่เหมาะสมกับสถานการณ์ต่างๆ อีกประการหนึ่ง ยังเป็นการเผยแพร่การท่องเที่ยวอันซีนอินไทยแลนด์ในเมืองไทย ให้บุคคลทั่วไปได้รับทราบผ่านทางรูปแบบเกมคอมพิวเตอร์ ที่สามารถเข้าถึงได้ง่ายอีกด้วย

Special Project Title	UNSEEN-IN-THAILAND RALLY GAME		
Student	Mr. Nattawut Laungarunutai	46050013	
	Mr. Witchaya Towongpaichayont	46050032	
Degree	Bachelor of Science		
Department	Mathematics and Computer Science , Faculty of Science		
Program	Applied Mathematics		
Academics Year	2006		
Special Project Advisor	Asst.Prof.Dr. Chartchai Leenawong		



## ABSTRACT

This Unseen-in-Thailand Rally Game is made to study and apply Artificial Intelligence (AI) to control the Non-Player Characters (NPC) in a computer board game. The AI knowledge includes Path Finding method for deciding the path to walkthrough as well as Fuzzy Logic and Probability for other decisions. This game is also aimed for propagandizing the Unseen in Thailand campaign to all walks of life using computer game as an easy-to-access tool.

## กิตติกรรมประกาศ

ในการทำปัญหาพิเศษฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีทางคณะผู้จัดทำต้องขอขอบพระคุณ ผศ.ดร.ฉัฐไชย์ สีนาวงศ์ อาจารย์ที่ปรึกษาปัญหาพิเศษที่กรุณาคอยให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่างๆ

ขอขอบคุณ เพื่อนๆ พี่ๆ น้องๆ ทุกคน ที่มีส่วนร่วมในการวางระบบและพัฒนาปัญหาพิเศษนี้ ให้สำเร็จลุล่วงไปได้ด้วยดี และคอยรับคำปรึกษาในการแก้ไขปัญหาต่างๆ

ขอขอบพระคุณคณาจารย์ประจำภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ทุกท่าน ที่ได้ประสานวิชาความรู้ต่างๆ ให้แก่ผู้จัดทำจนกระทั่งปัญหาพิเศษสัมฤทธิ์ผลด้วยดีทุกประการ รวมถึงเจ้าหน้าที่ประจำภาควิชาทุกท่าน ที่ช่วยเหลือในด้านการอำนวยความสะดวกเกี่ยวกับอุปกรณ์ที่จำเป็นต่างๆ

ขอขอบพระคุณบิดา-มารดาที่ได้ให้การสนับสนุนทุกประการทางการศึกษา และยังให้กำลังใจตลอดมาจนถึงปัจจุบัน และต้องขอขอบคุณเพื่อนๆ นักศึกษาทุกท่านที่ช่วยเหลือให้คำแนะนำต่างๆ จนปัญหาพิเศษนี้สำเร็จลุลงมบูรณ์ยิ่งขึ้น

คุณค่าและประโยชน์อันพึงมีจากปัญหาพิเศษฉบับนี้ ผู้จัดทำขออุทิศแด่ บิดา มารดา และผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ  
มีนาคม 2550

## สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ .....	III
สารบัญ.....	IV
สารบัญรูปภาพ.....	V
บทที่ 1 บทนำ.....	1
บทที่ 2 นิยามและทฤษฎีเบื้องต้น.....	3
2.1 ความหมายของระบบปัญญาประดิษฐ์.....	3
2.1.1 Deterministic AI.....	3
2.1.2 Nondeterministic AI.....	4
2.2 Finite State Machines.....	4
2.3 Fuzzy Logic.....	5
2.3.1 พื้นฐานของ Fuzzy Logic ที่ใช้ในโปรแกรมเกม.....	5
2.3.2 Fuzzification.....	6
2.3.3 Fuzzy Rules.....	13
2.3.4 การหาค่าจาก Rules.....	14
2.3.5 Defuzzification.....	16
2.4 ความน่าจะเป็น.....	17
2.5 Path Finding.....	17
บทที่ 3 ขั้นตอนการดำเนินงาน.....	19
3.1 ขั้นตอนการดำเนินงานโดยรวม.....	19
3.2 ขั้นตอนการออกแบบโปรแกรม.....	20
3.3 ขั้นตอนการดำเนินการของปัญญาประดิษฐ์.....	24
3.3.1 Fuzzy Logic.....	24
3.3.2 Path Finding.....	25
บทที่ 4 ขั้นตอนการใช้งานโปรแกรม.....	32
บทที่ 5 สรุปผลและข้อเสนอแนะ.....	44
บรรณานุกรม.....	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ

	หน้าที่
รูปที่ 2-1 แผนภูมิ Finite State Machine.....	4
รูปที่ 2-2 กระบวนการขั้นตอนของ Fuzzy Logic.....	6
รูปที่ 2-3 เส้นของฟังก์ชันพีชคณิตบูลีน.....	7
รูปที่ 2-4 เส้นของฟังก์ชันแบ่งระดับ.....	7
รูปที่ 2-5 เส้นแสดงฟังก์ชันน้ำหนักตัว.....	8
รูปที่ 2-6 เส้นของฟังก์ชันสามเหลี่ยม.....	9
รูปที่ 2-7 เส้นของฟังก์ชันแบ่งระดับแบบกลับด้าน.....	9
รูปที่ 2-8 เส้นของฟังก์ชันสี่เหลี่ยมคางหมู.....	10
รูปที่ 2-9 เส้นของฟังก์ชันรวม 7 ฟังก์ชัน.....	10
รูปที่ 2-10 เส้นของฟังก์ชันของสภาวะของตัวละคร.....	15
รูปที่ 2-11 เส้นของฟังก์ชันในการตัดสินใจของตัวละคร.....	16
รูปที่ 2-12 เส้นของฟังก์ชันในการตัดสินใจของตัวละครที่แปลงแล้ว.....	16
รูปที่ 2-13 ตัวอย่างลักษณะของกราฟทางเดิน.....	18
รูปที่ 3-1 แผนภูมิแสดงขั้นตอนการทำงานของโปรแกรมโดยรวม.....	20
รูปที่ 3-2 แผนภูมิแสดงขั้นตอนการทำงานของโปรแกรมในการเลือกเมนู.....	21
รูปที่ 3-3 แผนภูมิแสดงขั้นตอนการทำงานของโปรแกรมในการเลือกตัวละคร.....	21
รูปที่ 3-4 แผนภูมิแสดงขั้นตอนการทำงานของโปรแกรมระหว่างเทิร์นของผู้เล่น.....	22
รูปที่ 3-4 (ต่อ) แผนภูมิแสดงขั้นตอนการทำงานของโปรแกรมระหว่างเทิร์นของผู้เล่น.....	23
รูปที่ 4-1 หน้าจอเริ่มต้นการเข้าเกม.....	32
รูปที่ 4-2 หน้าจอเมนูหลัก.....	33
รูปที่ 4-3 หน้าจอเลือกตัวละคร.....	35
รูปที่ 4-4 หน้าจอแสดงวิธีการเล่น.....	35
รูปที่ 4-5 หน้าจอแสดงจุด Unseen in Thailand ทั้งหมดที่ต้องไป.....	36
รูปที่ 4-6 หน้าจอเริ่มต้นการเทิร์นของผู้เล่น.....	37
รูปที่ 4-7 หน้าจอการใช้ความสามารถเมื่อเริ่มต้นเทิร์น.....	37
รูปที่ 4-8 หน้าจอแสดงการทอยเต๋า.....	38
รูปที่ 4-9 หน้าจอการเลือกจุดที่เดิน.....	39
รูปที่ 4-10 หน้าจอการเข้าสู่การต่อสู้.....	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญญรูปภาพ (ต่อ)

	หน้าที่
รูปที่ 4-11 หน้าจอการเลือกคู่ต่อสู้.....	40
รูปที่ 4-12 หน้าจอเลือกการกระทำในการต่อสู้.....	41
รูปที่ 4-13 หน้าจอการเกิดเหตุการณ์.....	42
รูปที่ 4-14 หน้าจอการแสดงผลเมื่อจบเทิร์น.....	42



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของปัญหา

เนื่องด้วยปัจจุบัน เยาวชนไทยห่างไกลจากธรรมชาติ เข้าสู่โลกเสมือน จริงในคอมพิวเตอร์มากขึ้น ทำให้ไม่รู้จักแหล่งท่องเที่ยวที่น่าสนใจของเมืองไทย ซึ่งเป็นเรื่องที่น่าเสียดายอย่างยิ่ง ทำให้ผู้จัดทำมีแนวคิดที่จะจัดทำเกมกระดาน (Board Games) เพื่อเป็นสื่อกลางเพื่อให้เยาวชนและผู้สนใจ สามารถเข้าถึงแหล่งท่องเที่ยวที่น่าสนใจ ผ่านทางเกมกระดาน โดยผู้จัดทำหยิบยกแหล่งท่องเที่ยว Unseen in Thailand มาเป็นเนื้อหาของเกม เนื่องจากเป็นแหล่งท่องเที่ยวที่ได้รับการสนับสนุนอย่างดีจากรัฐบาล จึงเป็นแหล่งท่องเที่ยวที่ควรเผยแพร่ให้เยาวชนได้รับทราบ อีกประการหนึ่ง เกมนี้สามารถเล่นได้ทั้งครอบครัว จะสามารถสร้างความสนุกเพลิดเพลินไปด้วยกันทั้งผู้ปกครองและเยาวชน ผ่านทางตัวละครในเกม ซึ่งต้องแข่งขันกับตัวละครที่ควบคุมด้วยปัญญาประดิษฐ์ (Artificial Intelligence) จึงเป็นเกมที่สร้างเสริมความสัมพันธ์ในครอบครัวได้อีกทางหนึ่งด้วย

### 1.2 วัตถุประสงค์ของการทำ

1. เผยแพร่แหล่งท่องเที่ยว Unseen in Thailand ของไทย ให้เยาวชนไทยและบุคคลทั่วไปรับทราบ ผ่านทางเกมกระดาน เพื่อกระตุ้นความรู้สึกอยากท่องเที่ยวในประเทศไทย
2. ศึกษารูปแบบทางคณิตศาสตร์ในระบบปัญญาประดิษฐ์ เพื่อเขียนโปรแกรมควบคุมตัวละครในเกมให้ผู้เล่นได้รับความเพลิดเพลินแม้เล่นคนเดียว
3. เพื่อใช้เกมนี้เป็นสื่อในการส่งเสริมสถาบันครอบครัว อันเป็นสถาบันขั้นพื้นฐานของสังคม
4. จุดประกายความรู้สึกอยากผลิตเกมคอมพิวเตอร์ เป็นอุตสาหกรรมในเมืองไทย เพื่อไปแข่งขันกับตลาดโลก

### 1.3 ขอบเขตของปัญหา

ศึกษาระบบปัญญาประดิษฐ์เพื่อใช้ในการควบคุมการตัดสินใจของตัวละครที่ควบคุมด้วยคอมพิวเตอร์ พร้อมทั้งออกแบบเกมกระดาน (Board Games) เพื่อเป็นสื่อกลางในการเผยแพร่สถานที่ท่องเที่ยวของประเทศไทยผ่านทางเกมนั้น

#### 1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาข้อมูลเกี่ยวกับสถานที่ท่องเที่ยว Unseen in Thailand
2. ศึกษาการเขียนโปรแกรมด้วยภาษา ซี พลัส พลัส และ โดเร็กเอ็กซ์
3. ศึกษาการวางระบบปัญญาประดิษฐ์
4. ออกแบบโปรแกรมเกม
5. พัฒนาโปรแกรมเกม
6. ทดสอบการใช้งานโปรแกรมเกม
7. แก้ไขปรับปรุงเพิ่มเติม
8. จัดทำคู่มือการใช้งานและเอกสารประกอบ

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำโปรแกรมเกมนี้เป็นสื่อกลางในการเผยแพร่ความรู้เกี่ยวกับแหล่งท่องเที่ยวที่น่าสนใจในประเทศไทยให้กับเยาวชนไทยและบุคคลทั่วไปได้

#### 1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

1. Computer CPU
  - Pentium 4 2.4 GHz.
  - Ram 256 MB
  - Harddisk 40 GB
2. Window XP
3. Scanner
4. Printer
5. Microsoft Visual C++ 6.0
6. Adobe Photoshop CS
7. Adobe Illustrator CS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### นิยาม และ ทฤษฎีเบื้องต้น

ในเกมที่วุ่นวาย จะมีระบบหนึ่งที่เรียกว่า "ปัญญาประดิษฐ์" (Artificial Intelligence - AI) ที่ไว้คอยควบคุมส่วนต่างๆของเกม ไม่ว่าจะเป็นตัวละคร สัตว์ประหลาด หรือ หุ่นยนต์ ตั้งแต่เกมคลาสสิกอย่าง Pac Man ไปจนถึงเกมล้ำยุคอย่าง Unreal Tournament ซึ่งถ้าเกมนั้นๆ มีระบบปัญญาประดิษฐ์ที่ไม่ดีแล้ว อาจจะทำให้เกมนั้นน่าเบื่อ จนถึงไม่สามารถเป็นที่นิยมได้เลย

#### 2.1 ความหมายของระบบปัญญาประดิษฐ์

ถ้าดูความหมายของระบบปัญญาประดิษฐ์ในพจนานุกรม ก็จะหมายความว่า "The ability of a computer or other machine to perform those activities that are normally thought to require intelligence." ตามความหมายจาก The American Heritage Dictionary of the English Language, Fourth Edition (Houghton Mifflin Company) หรือแปลได้ความว่า ความสามารถของคอมพิวเตอร์หรือเครื่องจักรใดๆ ที่จะทำกิจกรรมที่โดยปกติต้องการความคิดในการทำ

ความสามารถของระบบปัญญาประดิษฐ์นั้น วัดกันที่ความสามารถนั้น ใกล้เคียงกับความคิดของมนุษย์ระดับไหน ซึ่งในบางครั้ง ระบบปัญญาประดิษฐ์ที่ควบคุมเครื่องจักรขนาดใหญ่ นั้น ต้องการความสามารถที่มากกว่าความคิดของมนุษย์ ระบบปัญญาประดิษฐ์ต้องเรียนรู้และดัดแปลงจากสถานการณ์จริง แต่นั่นก็เป็นระบบปัญญาประดิษฐ์ที่เรียกว่า Strong AI ซึ่งระบบปัญญาประดิษฐ์ในเกม ไม่ต้องการถึงขนาดนั้น จึงมีระบบปัญญาประดิษฐ์อีกแบบหนึ่งที่เรียกว่า Weak AI ที่มีขอบเขตการใช้งานที่กว้างกว่า และมีขอบเขตความคิดใกล้เคียงกับมนุษย์ก็เพียงพอแล้ว

สำหรับประเภทของพฤติกรรมของระบบปัญญาประดิษฐ์ แบ่งออกเป็น 2 กลุ่มใหญ่ๆ คือ Deterministic AI และ Nondeterministic AI

##### 2.1.1 Deterministic AI

พฤติกรรมของระบบปัญญาประดิษฐ์ประเภทนี้จะสามารถระบุและทำนายล่วงหน้าได้ ตัวอย่างของระบบปัญญาประดิษฐ์ประเภทนี้ได้แก่ กระบวนการไล่ล่าแบบปกติ (Simple Chasing Algorithm) ซึ่งสามารถเขียนโค้ดให้ตัวละครไล่ล่าเป้าหมายไปตามแกน  $x$  และ  $y$  เราจึงสามารถทำนายล่วงหน้าได้ว่าตัวละคร จะเคลื่อนที่ไปในทิศทางใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.2 Nondeterministic AI

พฤติกรรมของระบบปัญญาประดิษฐ์นี้ จะตรงข้ามกับประเภท Deterministic คือ มีระดับของความไม่แน่นอน และบางครั้งก็ไม่สามารถทำนายล่วงหน้าได้ ตัวอย่างของระบบปัญญาประดิษฐ์ประเภทนี้เช่น ตัวละครที่ไม่ใช่ผู้เล่น (NPC) ในเกมประเภทต่อสู้ ที่สามารถเรียนรู้กลยุทธ์ของผู้เล่นได้ โดยใช้กระบวนการเช่น Neural Network, เทคนิค Bayesian หรือ Genetic Algorithm

ซึ่งระบบปัญญาประดิษฐ์ที่จะใช้ในเกมส์ทั่วไทยไปกับอินซันอินไทยแลนด์นี้ ได้ใช้ระบบปัญญาประดิษฐ์มากมายหลายแบบ ดังจะกล่าวต่อไป

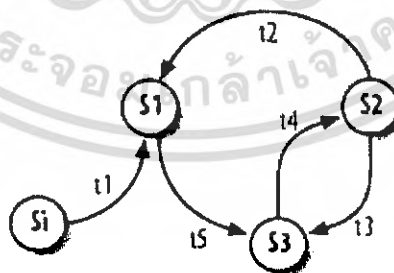
### 2.2 Finite State Machines

Finite State Machine เป็นโครงสร้างของกลไก ที่สามารถแก้ไขสถานการณ์ต่างๆกันได้ โดยการเปลี่ยนสถานะหนึ่ง ไปเป็นอีกสถานะหนึ่ง ตัวอย่างการใช้ในการเขียนโปรแกรมเกม เช่น เกม Pac Man เมื่อผู้เล่นกินยาแปลงพลังแล้ว สัตว์ประหลาดจะเปลี่ยนจากสถานะที่ไล่ล่าผู้เล่น เป็นหลบหนีจากผู้เล่นแทน ซึ่งสถานะที่แตกต่างกันนี้ ก็เป็นหนึ่งใน Finite State Machine นั่นเอง

ถึงแม้ว่า Finite State Machine จะเป็นเรื่องที่ธรรมดา แตกก็เป็นสิ่งที่จำเป็นสำหรับเกมหลายเกม ทั้งยังเข้าใจง่าย และสามารถนำไปพัฒนาต่อ รวมทั้งแก้ไขโปรแกรมได้ง่ายอีกด้วย

Finite State Machine ขั้นพื้นฐาน

โดยทั่วไปแล้ว จะมีลักษณะเป็นแผนภูมิดังต่อไปนี้



รูปที่ 2-1 แผนภูมิ Finite State Machine

ซึ่งจากแผนภูมิ จะมีสถานะทั้งหมด 4 สถานะ โดยมีฟังก์ชันทั้งหมด 5 ตัวที่จะแปลงจากสถานะไปอีกสถานะหนึ่ง โดยมีสถานะเริ่มต้น เป็น Si นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเกมท่องเที่ยวทั่วไทยไปกับอันซีนอินไทยแลนด์นี้ ใช้ Finite State Machine ในการควบคุมสถานะหลายส่วน เช่น Game State ที่ควบคุมสถานะของเกมทั้งหมด ซึ่งประกอบด้วย Title, Opening, Playing, Ending และ Credit หรือ Playing State ที่ควบคุมสถานะของในสถานการณ์เกม ซึ่งประกอบด้วย Start Turn, Use Skill, Roll Dice, Show Dice, Find Path, Go Event, Go Battle, Go Train, Go Plane และ End Turn เป็นต้น

## 2.3 Fuzzy Logic

ในปี 1965 Lotfi Zadeh ศาสตราจารย์แห่ง University of California Berkeley ได้เขียนเอกสารรายงานเกี่ยวกับทฤษฎีของ Fuzzy Set ไว้ 2 อย่างว่า "Fuzzy Logic คือหนทางที่คอมพิวเตอร์จะแก้ไขปัญหา ในวิถีทางที่มนุษย์ทำ" และ "แก่นแท้ของ Fuzzy Logic คือทุกสิ่งทุกอย่างจะเกิดขึ้นจากระดับความน่าจะเป็น"

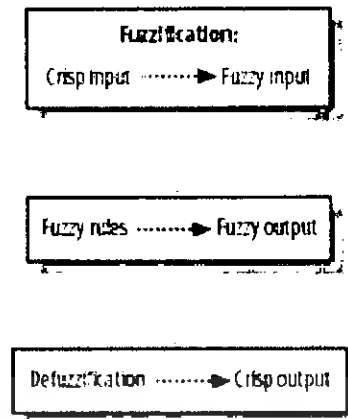
สำหรับแนวคิดของคำกล่าวแรกนั้น เกิดจากมนุษย์ทำการวิเคราะห์สถานการณ์หรือปัญหา ในความถี่ค่อนข้างสูงมาก แต่การวิเคราะห์นี้ อาจจะไม่ต้องการรายละเอียดข้อเท็จจริงมากนัก ตัวอย่างเช่น เมื่อคุณแข่งบาสเกตบอลกับเพื่อนของคุณ ความสูงและความคล่องแคล่วของฝ่ายตรงข้ามก็จะมีผลต่อการตัดสินใจของคุณ ถ้าคุณเจอกับคู่ต่อสู้ที่สูงและว่องไว คุณอาจจะไปป้องกันคู่ต่อสู้คนอื่นแทน แต่ถ้าเขาสูงมาก แต่เชื่องช้า คุณอาจจะเข้าไปป้องกันคู่ต่อสู้คนนั้น แต่คุณคงไม่ตัดสินใจจากข้อมูล "เขาสูง 6 ฟุต 5.5 นิ้ว และสามารถวิ่งตัดผ่านสนามในเวลา 5.7 วินาที" เป็นต้น

สำหรับคำกล่าวที่สอง สามารถยกตัวอย่างได้จากเดิมคือ การแข่งขันบาสเกตบอลนั้น คู่ต่อสู้จะมีความสูงเฉลี่ยหรือไม่ก็สูงมาก คุณไม่ต้องกำหนดขอบเขตว่าต้องความสูงเกิน 7 ฟุต เขาจะสูง แต่ถ้าเตี้ยกว่านั้น จะเป็นคนเตี้ย เพราะฉะนั้นไม่ว่าสูง 6 ฟุต หรือ 4 ฟุต ก็จะเป็นคนเตี้ยเหมือนกัน ซึ่งในความเป็นจริงไม่ได้เป็นเช่นนั้น นั่นคือ Fuzzy Logic จะสามารถเปลี่ยนข้อมูลให้อยู่ในรูปของระดับ (Degrees) ได้ เช่น สูงเป็น Degree 0.65 เป็นต้น

Fuzzy Logic นี้ ได้นำไปใช้ควบคุมเครื่องจักรต่างๆมากมาย เช่น เครื่องปรับอากาศ ซึ่งจะปรับความแรงของลมที่พัดออกมาตามอุณหภูมิห้อง หรือ เครื่องซักผ้าที่ปรับความแรงของการปั่นได้ตามน้ำหนักผ้า เป็นต้น

### 2.3.1 พื้นฐานของ Fuzzy Logic ที่ใช้ในโปรแกรมเกม

Fuzzy Logic นั้น มีกระบวนการพื้นฐาน 3 ขั้นตอน ดังรูปนี้



รูปที่ 2-2 กระบวนการขั้นตอนของ Fuzzy Logic

ในส่วนแรกของการกระบวนการเรียกว่าส่วน Fuzzification Process ซึ่งจะทำให้การแปลง Crisp Data หรือจำนวนจริงไปสู่ Fuzzy Data ซึ่งก็คือ Degree นั้นเอง ตัวอย่างเช่น เมื่อเราทำการป้อนข้อมูลน้ำหนักตัวคนเข้าไป ผลลัพธ์ที่ออกมา ก็จะเป็น Degree ของน้ำหนักที่แสดงว่าคนๆนั้น เบาเกินไป นหนักเกินไป หรือน้ำหนักพอดี

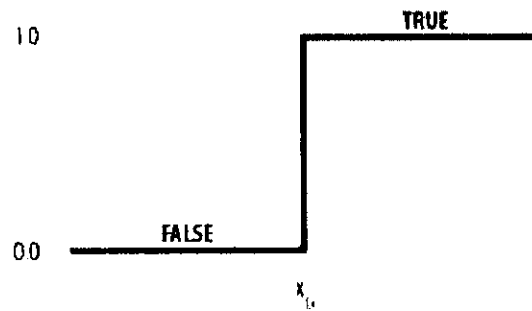
เมื่อได้ Fuzzy Data แล้ว จะใช้กระบวนการทางลอจิก หรือที่เรียกว่า Fuzzy Rules เพื่อที่จะหาความน่าจะเป็นที่ควรทำของแต่ละ Rules ตัวอย่างเช่น ให้น้ำหนักตัวคนและระดับของความตื่นตัวของร่างกายเป็นข้อมูลเข้า Rules ที่ออกมา ก็อาจจะเป็น ถ้า น้ำหนักตัวเกินและไม่ตื่นตัวแล้ว ให้ทำการออกกำลังกายอย่างสม่ำเสมอ หรือ ถ้า น้ำหนักตัวและเป็นคนตื่นตัวแล้ว ให้ทำการลดน้ำหนักอย่างค่อยเป็นค่อยไป เป็นต้น Rules เหล่านี้ จะเป็นการผสมกันระหว่างข้อมูลเข้าประเภท Fuzzy โดยใช้การกระทำทางลอจิก แล้วค่าที่ออกมาได้ ก็จะเป็นสิ่งที่ควรทำ

แต่ในบางครั้ง Rules ที่ออกมา อาจจะไม่เพียงพอกับที่ต้องการ เช่น ต้องการข้อมูลออกเป็นจำนวนชั่วโมงที่ต้องออกกำลังกายต่อสัปดาห์ จึงต้องใช้กระบวนการที่มีรายละเอียดมากกว่านี้ ซึ่งเราจะเจาะลึกรายละเอียดแต่ละขั้นตอนต่อไป

### 2.3.2 Fuzzification

ข้อมูลเข้าของการกระบวนการขั้นตอนนี้ จะเป็นจำนวนจริง เช่น คนหนัก 185.3 ปอนด์ หรือ สูง 6 ฟุต 1 นิ้ว ในกระบวนการ Fuzzification จะเปลี่ยนจำนวนจริงให้อยู่ในรูป Degree เช่น คนหนัก 185.3 ปอนด์ จะเป็นคนที่น้ำหนักเกินเล็กน้อย หรือ สูง 6 ฟุต 1 นิ้ว จะเป็นคนที่สูง เป็นต้น

เมื่อพิจารณาจากฟังก์ชันของพีชคณิตบูลีน จะพบว่าเส้นของฟังก์ชันเป็นเช่นนี้



รูปที่ 2-3 เส้นของฟังก์ชันพีชคณิตบูลีน

นั่นคือ ถ้าข้อมูลเข้ามีค่าต่ำกว่า  $x_0$  ค่าที่แปลงออกมาคือ False แต่ถ้าสูงกว่านั้นก็จะเป็นค่า True และไม่มีค่านอกเหนือจากนั้น ถ้าเรายกตัวอย่างน้ำหนัก ให้  $x_0$  เท่ากับ 170 ปอนด์ ดังนั้นถ้าคนที่หนักกว่า 170 ปอนด์ก็จะเป็นคนที่มีน้ำหนักตัวมากเกินไป ส่วนคนที่น้ำหนักน้อยกว่า 170 ปอนด์ก็จะมีน้ำหนักตัวน้อยเกินไป แม้แต่คนที่หนัก 169.9 ปอนด์ ก็จะถูกพิจารณาให้หนักน้อยเกินไปด้วย ซึ่งไม่ควรจะเป็นเช่นนั้น

ดังนั้น เราจะเปลี่ยนฟังก์ชันของพีชคณิตบูลีนเสียใหม่ เป็นฟังก์ชันที่ประกอบด้วย 3 ช่วง ดังต่อไปนี้



รูปที่ 2-4 เส้นของฟังก์ชันแบ่งระดับ

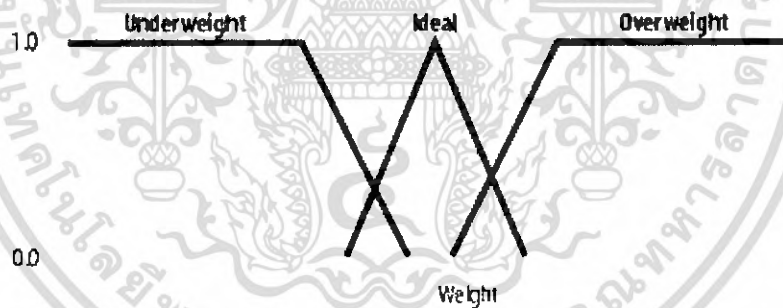
ฟังก์ชันนี้เรียกว่า Grade Function จะสังเกตได้ว่า ส่วนค่าฟังก์ชันระหว่าง  $x_0$  และ  $x_1$  นั้น จะมีค่าอยู่ระหว่าง 0 กับ 1 นั่นคือถ้าค่าน้อยกว่า  $x_0$  ฟังก์ชันเป็น False โดยสมบูรณ์ ถ้ามีค่ามากกว่า  $x_1$  ฟังก์ชันจะเป็น True โดยสมบูรณ์ ส่วนค่าที่อยู่ระหว่าง  $x_0$  และ  $x_1$  นั้น ฟังก์ชันจะอยู่ระหว่าง 0 กับ 1 โดยมีสมการฟังก์ชันเป็นเส้นตรง

เมื่อเขียน Grade Function อยู่ในรูปสมการแล้ว จะสามารถเขียนได้ดังนี้

$$f(x) = \begin{cases} 0, & x \leq x_0 \\ \frac{x - x_0}{x_1 - x_0}, & x_0 < x < x_1 \\ 1, & x \geq x_1 \end{cases}$$

กลับไปดูตัวอย่างน้ำหนักอีกครั้ง เมื่อให้  $x_0$  เท่ากับ 175 และ  $x_1$  เท่ากับ 195 ถ้าคนมีน้ำหนักตัว 170 ปอนด์ จะมีดีกรีของการน้ำหนักเกินเป็น 0 นั่นคือ เขาไม่มีน้ำหนักตัวเกิน ถ้ามีน้ำหนักตัว 185 ปอนด์ ก็จะมีดีกรีของการน้ำหนักเกินเป็น 0.5 นั่นคือเขาจะมีน้ำหนักตัวเกินบ้าง

โดยทั่วไปแล้ว เซตข้อมูลที่เราสงสัยจะไม่ได้มีเพียงหนึ่ง เช่น น้ำหนักตัวของคน จะแบ่งช่วงออกเป็น น้ำหนักเกิน น้ำหนักเบา และน้ำหนักพอดี ซึ่งเราจะวาดเส้นฟังก์ชันออกมาได้ ดังนี้



รูปที่ 2-5 เส้นแสดงฟังก์ชันน้ำหนักตัว

จากเส้นฟังก์ชันนี้ เราอาจจะมีดีกรีน้ำหนักเบาเป็น 0 ดีกรีน้ำหนักพอดีเป็น 0.75 และมีดีกรีน้ำหนักเกินเป็น 0.15 เป็นต้น

นอกจาก Grade Function ข้างต้นแล้ว ยังมีฟังก์ชันอีกหลายประเภท ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Triangular Function

$$f(x) = \begin{cases} 0; x \leq x_0 \\ \frac{x}{x_1 - x_0} - \frac{x_0}{x_1 - x_0}; x_0 < x < x_1 \\ 1; x = x_1 \\ \frac{-x}{x_2 - x_1} + \frac{x_2}{x_2 - x_1}; x_1 < x < x_2 \\ 0; x \geq x_2 \end{cases}$$

ซึ่งสามารถวาดเส้นของฟังก์ชันได้ดังนี้

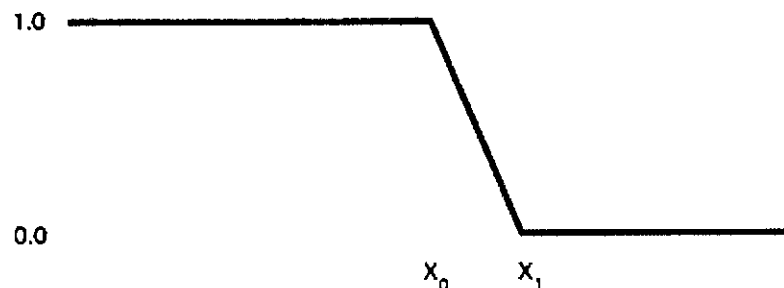


รูปที่ 2-6 เส้นของฟังก์ชันสามเหลี่ยม

Reverse Grade Function

$$f(x) = \begin{cases} 1; x \leq x_0 \\ \frac{-x}{x_1 - x_0} + \frac{x_1}{x_1 - x_0}; x_0 < x < x_1 \\ 0; x \geq x_1 \end{cases}$$

ซึ่งสามารถวาดเส้นของฟังก์ชันได้ดังนี้



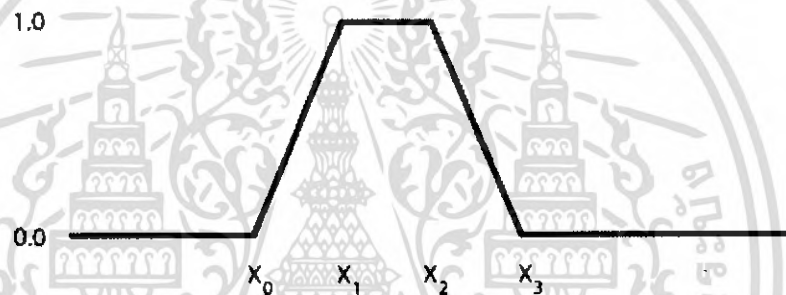
รูปที่ 2-7 เส้นของฟังก์ชันแบ่งระดับแบบกลับด้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Trapezoid Function

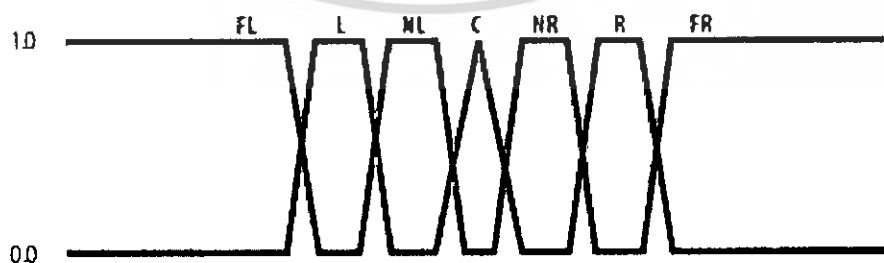
$$f(x) = \begin{cases} 0; x \leq x_0 \\ \frac{x}{x_1 - x_0} - \frac{x_0}{x_1 - x_0}; x_0 < x < x_1 \\ 1; x_1 \leq x \leq x_2 \\ \frac{-x}{x_3 - x_2} + \frac{x_3}{x_3 - x_2}; x_2 < x < x_3 \\ 0; x \geq x_3 \end{cases}$$

ซึ่งสามารถวาดเส้นของฟังก์ชันได้ดังนี้



รูปที่ 2-8 เส้นของฟังก์ชันสี่เหลี่ยมคางหมู

เมื่อนำฟังก์ชันประเภทต่างๆมารวมกัน เราจะสามารถกำหนด Fuzzy Set ที่สามารถแปลงค่าข้อมูลเข้าโดยมาค่าผิดพลาดน้อย และยังสามารถหาดีกรีที่เหมาะสมได้ ดังตัวอย่าง Fuzzy Set 7 ตัว ต่อไปนี้



รูปที่ 2-9 เส้นของฟังก์ชันรวม 7 ฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย Fuzzy Set ทั้ง 7 นี้ ประกอบด้วย กลาง ขวาใกล้ ขวา ขวาไกล ซ้ายใกล้ ซ้าย และซ้ายไกล ซึ่งเซตต่อไปนี้อาจเป็นเซตอะไรก็ได้ แล้วแต่ปัญหาที่ต้องการแก้

นอกจากนี้แล้ว ยังมีฟังก์ชันที่ไม่ใช่เส้นตรง แต่ไม่ขอกล่าวถึง เพราะเป็นเรื่องยุ่งยากเกินกว่าที่จะนำมาใช้เป็นระบบปัญญาประดิษฐ์ในเกม

เมื่อเรานำฟังก์ชันมาเขียนโค้ด จะออกมาดังตัวอย่างต่อไปนี้ ซึ่งเป็นโค้ดของ ภาษาซี

```
double FuzzyGrade(double value, double x0, double x1)
{
    double result = 0;
    double x = value;

    if(x <= x0)
        result = 0;
    else if(x >= x1)
        result = 1;
    else
        result = (x/(x1-x0))-(x0/(x1-x0));

    return result;
}

double FuzzyReverseGrade(double value, double x0, double x1)
{
    double result = 0;
    double x = value;

    if(x <= x0)
        result = 1;
```

```

else if(x >= x1)
    result = 0;
else
    result = (-x/(x1-x0))+(x1/(x1-x0));

return result;
}

```

```

double FuzzyTriangle(double value, double x0, double x1, double x2)
{
    double result = 0;
    double x = value;

    if(x <= x0)
        result = 0;
    else if(x == x1)
        result = 1;
    else if((x>x0) && (x<x1))
        result = (x/(x1-x0))-(x0/(x1-x0));
    else
        result = (-x/(x2-x1))+(x2/(x2-x1));

    return result;
}

```

```

double FuzzyTrapezoid(double value, double x0, double x1, double x2,
                      double x3)
{
    double result = 0;
    double x = value;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(x <= x0)
    result = 0;
else if((x>=x1) && (x<=x2))
    result = 1;
else if((x>x0) && (x<x1))
    result = (x/(x1-x0))-(x0/(x1-x0));
else
    result = (-x/(x3-x2))+(x3/(x3-x2));

return result;
}

```

ซึ่งเมื่อเราต้องการใช้งานฟังก์ชันประเภทไหนแล้ว ก็สามารถเรียกใช้งานเมธอดนั้นๆ โดยส่งค่าพารามิเตอร์เข้าไปเท่านั้น

### 2.3.3 Fuzzy Rules

หลังจากทำการแปลงจำนวนจริงเป็น Fuzzy Data แล้ว จะเข้าสู่ส่วนต่อไปคือ ตั้งกฎ (Rules) ขึ้นมา ซึ่งจะอยู่ในรูป ถ้า - แล้ว เป็นส่วนมาก และจะใช้พีชคณิตบูลีนเข้ามาช่วยในการหาดีกรี ดังต่อไปนี้

*Disjunction*

$$\text{Truth}(A \text{ OR } B) = \text{MAX}(\text{Truth}(A), \text{Truth}(B))$$

*Conjunction*

$$\text{Truth}(A \text{ AND } B) = \text{MIN}(\text{Truth}(A), \text{Truth}(B))$$

*Negation*

$$\text{Truth}(\text{NOT } A) = 1 - \text{Truth}(A)$$

นั่นคือเมื่อได้ Fuzzy Data มาเป็นดีกรีของแต่ละเซตแล้ว จะทำการผ่าน Rules ต่างๆ ซึ่งมีพีชคณิตบูลีนอยู่ จนได้ผลลัพธ์ออกมาเป็น Fuzzy Output ซึ่งเมื่อเขียนโค้ดแล้ว จะได้ดังตัวอย่างในภาษาซีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
double FuzzyAND(double A, double B)
{
    return MIN(A, B);
}
```

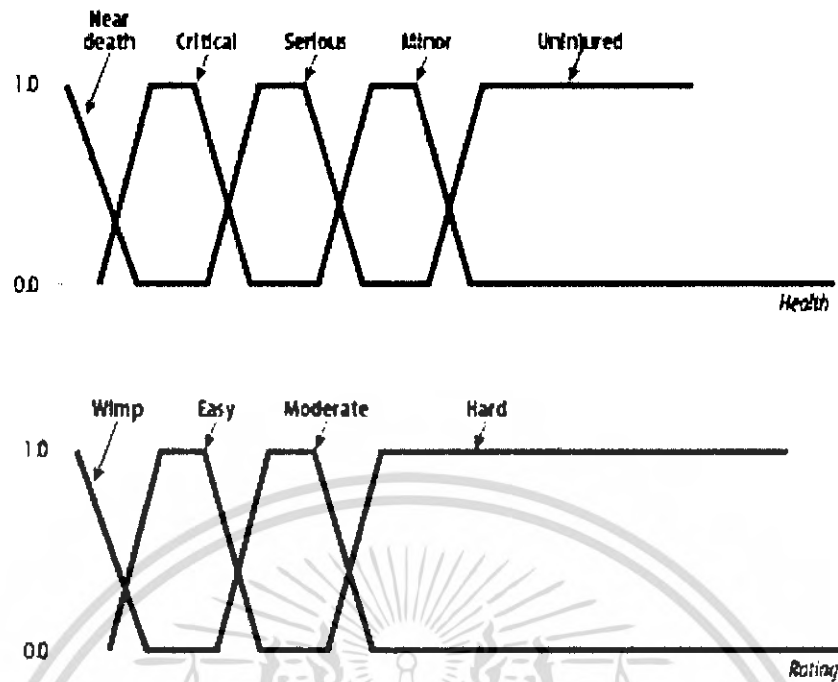
```
double FuzzyOR(double A, double B)
{
    return MAX(A, B);
}
```

```
double FuzzyNOT(double A)
{
    return 1.0 - A;
}
```

#### 2.3.4 การหาค่าจาก Rules

ในระบบบูลีน แต่ละ Rule จะหาค่าไปจนกว่าค่าใดค่าหนึ่งเป็น True และจะทำตาม Rule ตัวนั้น แต่สำหรับระบบ Fuzzy แล้ว แต่ละ Rule จะหาค่าไปแบบคู่ขนาน เพราะฉะนั้น Rule ทุกตัวจะทำงานพร้อมกันทั้งหมด อย่างไรก็ตาม Rule แต่ละตัวจะถูกกระทำตามด้วยดีกรีที่แตกต่างกัน และผลลัพธ์สุดท้ายคือผลรวมทางบูลีนของ Rule แต่ละตัวที่หาค่าด้วยดีกรีเรียบร้อยแล้ว

สำหรับเกมท่องเที่ยวทั่วไทยไปกับอินฟินิตไทยแลนด์นี้ ได้ใช้ Fuzzy Logic มาควบคุมการตัดสินใจในส่วนการใช้ความสามารถการเดินทาง, การต่อสู้ และ การใช้ความสามารถในการต่อสู้ ดังตัวอย่างต่อไปนี้



รูปที่ 2-10 เส้นของฟังก์ชันของสถานะของตัวละคร

ข้อมูลออกคือ การตัดสินใจที่จะเดินหนี โจมตี หรือไม่ต้องทำอะไรเลย ซึ่งเราอาจจะสามารถเขียนกฎได้เหมือนกับกฎต่อไปนี้

if (in melee range AND uninjured) AND NOT hard then attack  
 if (NOT in melee range) AND uninjured then do nothing  
 if (NOT out of range AND NOT uninjured) AND (NOT wimp) then flee

ซึ่งเราสามารถตั้งใจหยาบที่หลากหลายกว่านี้มากได้ในเกม ยังมีเซตของใจหยาบที่หลากหลายมาก ก็ยิ่งทำให้ระบบปัญญาประดิษฐ์มีความสมบูรณ์มากขึ้นเท่านั้น และเมื่อนำข้อมูลเข้ามาผ่านใจหยาบเหล่านี้แล้ว ข้อมูลออกก็จะเป็นดังเช่นต่อไปนี้

Attack to degree 0.7

Do nothing to degree 0.4

Flee to degree 0.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลออกซึ่งเป็นดีกรีของแต่ละ Rule จะถูกนำไปกระทำการตามพีชคณิตบูลีนซึ่งวิธีการที่ง่ายที่สุดคือ หาค่าดีกรีที่มากที่สุด ดังนั้น ถ้าดูตามตัวอย่างข้างต้นแล้ว ตัวละครควรจะตัดสินใจโจมตี

### 2.3.5 Defuzzification

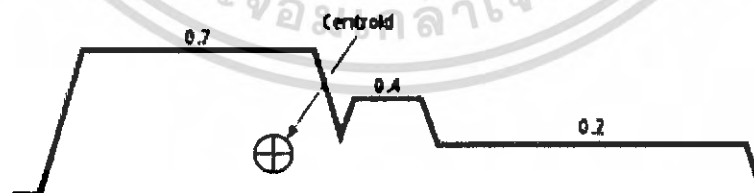
กระบวนการนี้ จะเป็นการแปลง Fuzzy Data กลับไปเป็นจำนวนจริง เนื่องด้วย Fuzzy Logic บางเซต จะมีความต่อเนื่องกัน ดังนั้นจึงต้องการแปลง Fuzzy Data กลับไปเป็นจำนวนจริง เพื่อเป็นข้อมูลเข้าของ Fuzzy Logic ตัวต่อไป ตัวอย่างเช่น จากที่ตัวละครตัดสินใจได้แล้วว่า จะหนี โจมตี หรือ ไม่ทำอะไรเลย เรายังต้องการรู้อีกว่า จะทำตามที่ตัดสินใจด้วยความเร็วแค่ไหน หรือยังมีสิ่งอื่นที่สำคัญกว่านี้ต้องทำหรือไม่ หรือถ้าโจมตี จะใช้ความสามารถโจมตีหรือโจมตีด้วยอัตราปกติ

เราจะสามารถวาดเส้นของฟังก์ชันได้ดังต่อไปนี้



รูปที่ 2-11 เส้นของฟังก์ชันในการตัดสินใจของตัวละคร

จากผลลัพธ์ที่เราคำนวณได้มาแล้ว จะได้ดีกรีการโจมตีเป็น 0.7 ดีกรีการอยู่เฉยๆ เป็น 0.4 และดีกรีการหนีเป็น 0.2 เราจะวาดเส้นของฟังก์ชันได้ใหม่ ดังนี้



รูปที่ 2-12 เส้นของฟังก์ชันในการตัดสินใจของตัวละครที่แปลงแล้ว

เพื่อที่จะแปลงข้อมูลออกเป็นจำนวนจริง เราสามารถเลือกใช้ได้หลายวิธี วิธีหนึ่งที่คุ้นเคยกันคือ หาพื้นที่ใต้กราฟของฟังก์ชัน โดยให้กระบวนการวิธีทาง Numerical แต่เนื่องด้วยกระบวนการที่ยู้งยากนั้น ไม่เหมาะกับการตัดสินใจของระบบปัญญาประดิษฐ์ที่

ต้องการความรวดเร็ว เราจึงมีวิธีการที่ง่ายกว่านั้นมาก ในการแปลงค่าเป็นจำนวนจริง โดยใช้วิธีการที่เรียกว่า Singleton Output Function

สำหรับวิธีการนี้ จะเป็นวิธีหาค่าที่ค่อนข้างหยาบ และไม่เหมาะกับสมการเส้นโค้ง แต่ก็เพียงพอสำหรับเขียนระบบปัญญาประดิษฐ์ในโปรแกรมเกม ซึ่งจะสมมติความเร็วของแต่ละการกระทำ เช่น -10 สำหรับการหนี 1 สำหรับการอยู่เฉยๆ และ 10 สำหรับการโจมตี ดังนั้น จากตัวอย่างจะได้ความเร็วในการหนีเป็น -2 ความเร็วในการอยู่เฉยๆเป็น 0.4 และความเร็วในการโจมตีเป็น 7 จากนั้น เราจะนำมาคำนวณโดยใช้สูตร ดังต่อไปนี้

$$\text{output} = \frac{\sum_{i=1}^n \mu_i x_i}{\sum_{i=1}^n \mu_i}$$

โดยที่  $x$  เป็นความเร็วที่สมมติขึ้น และ  $\mu$  เป็นค่าข้อมูลออกที่หาได้จาก Fuzzy Logic ครั้งแรก

ดังนั้นจากตัวอย่าง จะได้ Output เป็น  $[(0.7)(10) + (0.4)(1) + (0.3)(-10)] / (0.7 + 0.4 + 0.3) = 2.5$  นั่นคือ ตัวละครควรจะหนี แต่ไม่จำเป็นต้องใช้ความเร็วมากนัก นั่นคือ ใช้การโจมตีโดยไม่ต้องใช้ความสามารถเพิ่มนั่นเอง

## 2.4 ความน่าจะเป็น

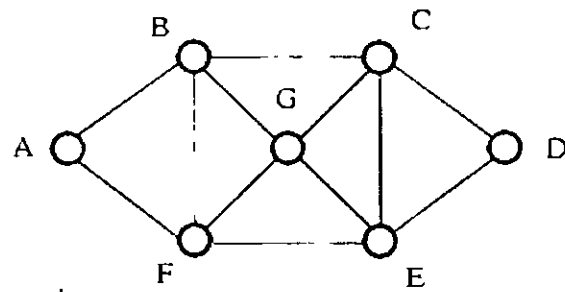
นักพัฒนาเกมจะใช้ความน่าจะเป็นในส่วนของโปรแกรมเกมมากมายหลายอย่าง ไม่ว่าจะเป็น ความเสียหาย ความแม่นยำ ความเร็ว ซึ่งจะมีจุดเด่นคือ ไม่สามารถคาดการณ์ล่วงหน้าได้ ทราบแต่โอกาสที่จะเกิดขึ้นเท่านั้น

การที่จะสร้างความน่าจะเป็นนั้น จะใช้โค้ดทางภาษาระดับสูง เช่น ภาษาซี หรือ ภาษาจาวา ซึ่งจะมีคำสั่งในการสุ่มค่าอยู่แล้ว จึงไม่มีปัญหาใดๆในการสุ่มค่า เพียงแต่ต้องสร้างเงื่อนไขของการกระทำต่างๆเท่านั้น

## 2.5 Path Finding

ในการเขียนเกมต่างๆที่ต้องมีการเคลื่อนไหวของตัวละครนั้น จำเป็นอย่างยิ่งที่จะต้องหาเส้นทางในการเคลื่อนที่ที่ดีที่สุด จึงเป็นที่มาของกรรมวิธีการค้นหาเส้นทางหรือ Path Finding ซึ่ง Path Finding Method ที่นำมาใช้ในเกมท่องเที่ยวทั่วไทยไปกับอินซันอินไทยแลนด์นี้ เป็น Path Finding ที่ใช้กับเส้นทางที่เป็นลักษณะกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-13 ตัวอย่างลักษณะของกราฟทางเดิน

ซึ่งกราฟนี้ จะถูกเก็บไว้ในลักษณะของอาร์เรย์ 2 มิติ ที่เก็บระยะห่างระหว่าง 2 Node ไว้ หรือ Weight นั้นเอง

1211

2001

1003

1212

จากอาร์เรย์ข้างต้นนี้ ให้จุดซ้ายบน หรือ (0, 0) แทนจุด A และจุดขวาล่าง หรือ (3, 3) แทนจุด B การหาเส้นทางจาก A ไป B จะหาจากการเดินผ่านเส้นทางที่ Weight ไม่เท่ากับศูนย์ ดังอาร์เรย์ข้างต้น จะไปได้ 2 เส้นทาง คือ (0, 0) – (0, 1) – (0, 2) – (0, 3) – (1, 3) – (2, 3) – (3, 3) และ (0, 0) – (1, 0) – (2, 0) – (3, 0) – (3, 1) – (3, 2) – (3, 3)

เมื่อได้เส้นทางแล้ว จะทำการรวม Weight ของเส้นทางทั้งหมด เพื่อหาเส้นทางที่สั้นที่สุด หรือ Shortest Path เพื่อเลือกเส้นทางนั้น เช่น ในตัวอย่างข้างต้น Weight รวมของเส้นทางแรก คือ  $1+2+1+1+1+3+2 = 11$  และเส้นทางที่สอง คือ  $1+2+1+1+2+1+2 = 10$  ทำให้เราเลือกเส้นทางที่สอง

ในเกมท่องเที่ยวทั่วไทยไปกับอันซีนอินไทยแลนด์นี้ จุดหมายปลายทางคือ จุด Unseen in Thailand ที่ต้องไปทั้งหมด เมื่อเลือกเส้นทางได้แล้ว ตัวละครจะเคลื่อนไปตามเส้นทางที่เก็บไว้ตาม Active Point ของหน้าเต้าที่ทอยได้ จนกว่าจะไปถึงจุดหมาย แล้วจึงจะทำการ Path Finding เพื่อหาจุดหมายใหม่ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

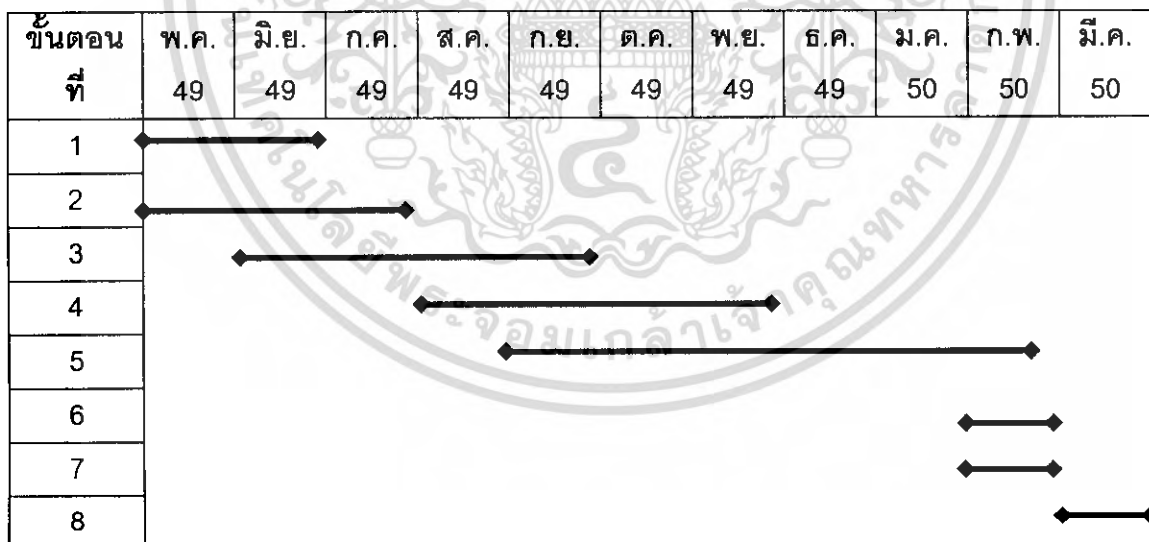
### บทที่ 3 ขั้นตอนการดำเนินงาน

#### 3.1 ขั้นตอนการดำเนินงานโดยรวม

ดังเช่นในบทที่ 1 ขั้นตอนการดำเนินงานทั้งหมดมีดังนี้

1. ศึกษาข้อมูลเกี่ยวกับสถานที่ท่องเที่ยว Unseen in Thailand
2. ศึกษาการเขียนโปรแกรมด้วยภาษา ซี พลัส พลัส และ ไดร็กเอ็กซ์
3. ศึกษาการวางระบบปัญญาประดิษฐ์
4. ออกแบบโปรแกรมเกม
5. พัฒนาโปรแกรมเกม
6. ทดสอบการใช้งานโปรแกรมเกม
7. แก้ไขปรับปรุงเพิ่มเติม
8. จัดทำคู่มือการใช้งานและเอกสารประกอบ

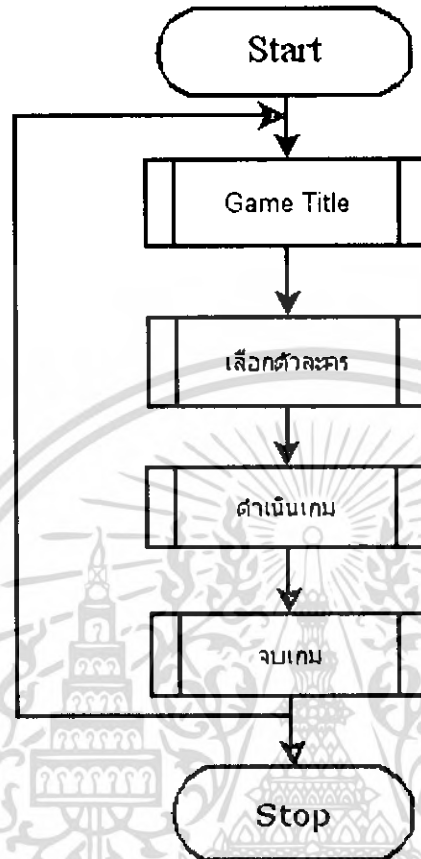
ซึ่งเมื่อนำมาทำเป็นแผนภูมิ Gantt chart แล้ว จะเป็นเช่นนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

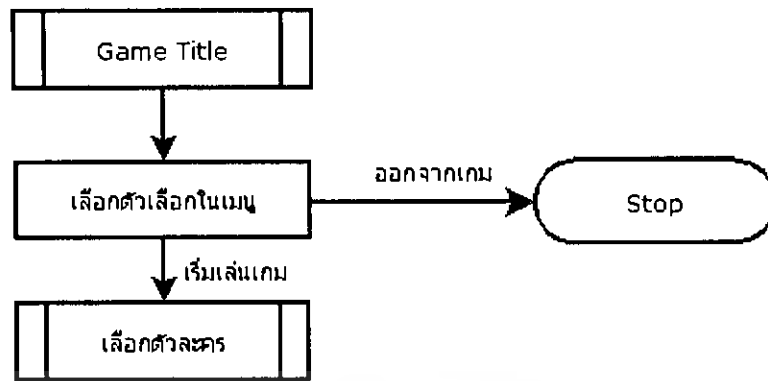
### 3.2 ขั้นตอนการออกแบบโปรแกรม

ใช้รูปแผนภูมิ (Flow chart) ในการวิเคราะห์แสดงขั้นตอนกระบวนการของโปรแกรม

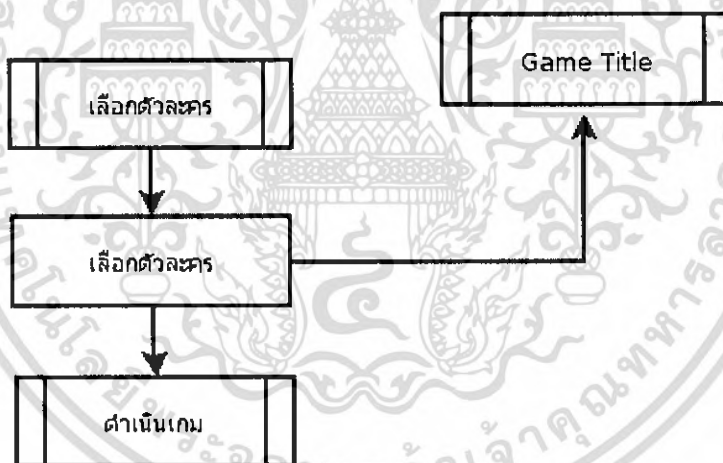


รูปที่ 3-1 แผนภูมิแสดงขั้นตอนการทำงานของโปรแกรมโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

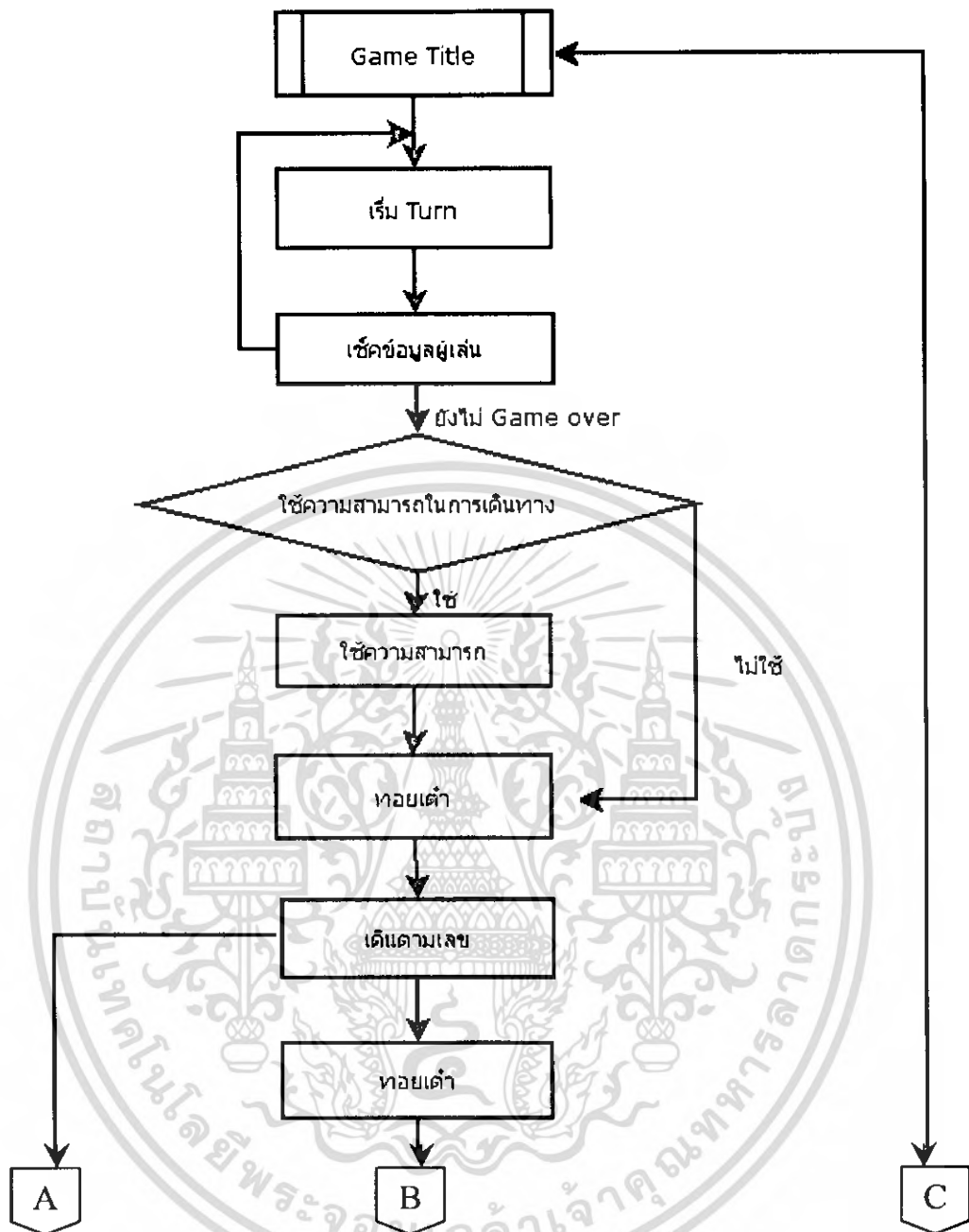


รูปที่ 3-2 แผนภูมิแสดงขั้นตอนการทำงานของโปรแกรมในการเลือกเมนู



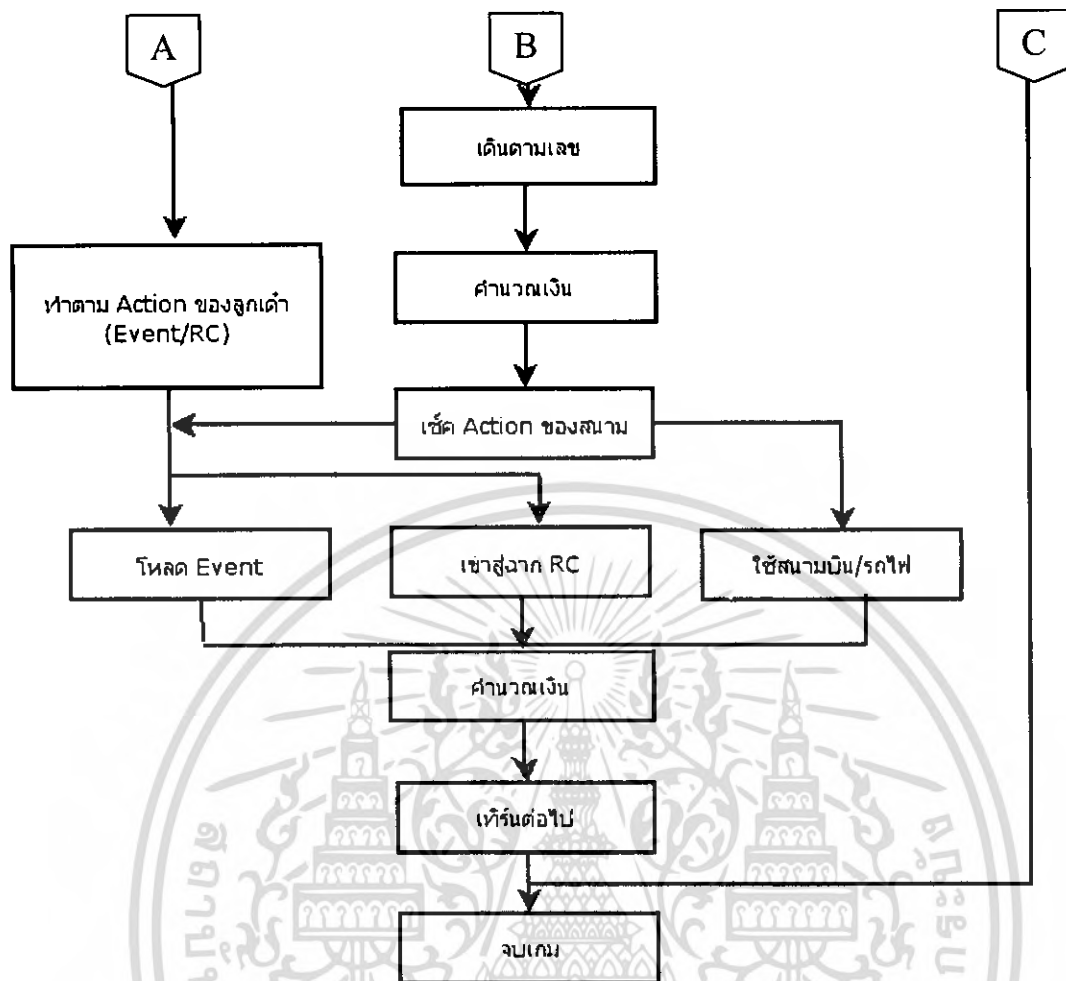
รูปที่ 3-3 แผนภูมิแสดงขั้นตอนการทำงานของโปรแกรมในการเลือกตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-4 แผนภูมิแสดงขั้นตอนการทำงานของโปรแกรมระหว่างเทิร์นของผู้เล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-4 (ต่อ) แผนภูมิแสดงขั้นตอนการทำงานของโปรแกรมระหว่างเทิร์นของผู้เล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 ขั้นตอนการดำเนินการของปัญญาประดิษฐ์

#### 3.3.1 Fuzzy Logic

ในส่วน Fuzzy Logic จะถูกนำไปใช้ในด้านการตัดสินใจในการใช้ความสามารถ ดังที่จะขอยกตัวอย่างการตัดสินใจการใช้ความสามารถการเดินทางของตัวละครชาติไทย ดังนี้

ความสามารถการเดินทางของตัวละครชาติไทยคือ สามารถผ่านการเล่นเพื่อลด ค่าเดินทางของผู้เล่นทุกคน 1,000 บาท โดยที่เงื่อนไขในการตัดสินใจคือ จำนวนค่า เดินทางของตัวเองและค่าเดินทางของผู้เล่นทุกคน

- ถ้าค่าเดินทางฝ่ายเรามากกว่า 10,000 บาท และมีคู่ต่อสู้คนใดคนหนึ่งมีค่าเดินทางน้อยกว่า 3,000 บาท จะทำการใช้ความสามารถนี้ทันที

```
if (player[turn].money >= 10000 && (player[1].money <= 3000 || player[2].money <= 3000 || player[3].money <= 3000 || player[4].money <= 3000))
```

```
    PlayingState = USESKILL;
```

- แต่ถ้าค่าเดินทางฝ่ายเราน้อยกว่า 3,000 บาท ให้งดการใช้ความสามารถนี้

```
else if (player[turn].money <= 3000)
```

```
    PlayingState = ROLLDICE;
```

- แต่ถ้านอกเหนือจากนั้น จะแรนด้อมค่าอยู่ในช่วง โดยความกว้างของช่วงขึ้นกับค่าเดินทางคงเหลือของเรา

```
else
```

```
    if (chance <= ((player[turn].money/(700))-(5000/(700))))
```

```
        PlayingState = USESKILL;
```

```
    else    PlayingState = ROLLDICE;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้







```

for(int i=0;i<sizeX;++i)
    a[i]=(int*)malloc(sizeof(int)*sizeY);

int tempPos = 0;
for(int j=0;j<sizeX;++j){
    for(int k=0;k<sizeY;++k){
        a[j][k] = temp[tempPos];
        tempPos++;
    }
}
}
}

```

เมื่อทำการ Initialize แล้ว จึงทำการท่องเที่ยวในกราฟด้วยวิธี Depth First Search

```

int greedyDepthFirstSearch(int nStartX, int nStartY, int nEndX, int nEndY,
                           int aXSize, int aYSize){
    int nPosNowX=nStartX,nPosNowY=nStartY;

    struct NodePosition startNode;
    startNode.x = nStartX;
    startNode.y = nStartY;
    startNode.pass = 1;

    node.push_back(startNode);

    if(startNode.x == nEndX && startNode.y == nEndY){
        toArray(node);
        return 0;
    }
}

```

```

struct NodePosition nodeTemp;
nodeTemp.pass = 0;

/* Flag of : left up right down */
//int go1=0,go2=0,go3=0,go4=0;

/* Check every possible way that can go and set flag*/
if(nPosNowX-1 >=0 &&
    isWalkAble(nPosNowX-1, nPosNowY) &&
    isNotAlreadyWalk(nPosNowX-1, nPosNowY, node)){
    greedyDepthFirstSearch(nPosNowX-1, nPosNowY,
        nEndX, nEndY, aXSize, aYSize);
    node.pop_back();
}
if(nPosNowY-1 >=0 &&
    isWalkAble(nPosNowX, nPosNowY-1) &&
    isNotAlreadyWalk(nPosNowX, nPosNowY-1, node)){
    greedyDepthFirstSearch(nPosNowX, nPosNowY-1,
        nEndX, nEndY, aXSize, aYSize);
    node.pop_back();
}
if(nPosNowX+1 < aXSize &&
    isWalkAble(nPosNowX+1, nPosNowY) &&
    isNotAlreadyWalk(nPosNowX+1, nPosNowY, node)){
    greedyDepthFirstSearch(nPosNowX+1, nPosNowY,
        nEndX, nEndY, aXSize, aYSize);
    node.pop_back();
}
if(nPosNowY+1 < aYSize &&
    isWalkAble(nPosNowX, nPosNowY+1) &&
    isNotAlreadyWalk(nPosNowX, nPosNowY+1, node)){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        greedyDepthFirstSearch(nPosNowX,      nPosNowY+1,
        nEndX, nEndY, aXSize, aYSize);
        node.pop_back();
    }
    return 0;
}

```

จะได้เส้นทางที่เดินทางไปอยู่ในรูปของอาร์เรย์ของ Struct ที่เก็บค่า x และ y ของ Node ไว้ และเมื่อได้เส้นทางแล้ว จึงทำการเดินไปตามเส้นทางนั้นจนหมด Active Point

```

do
{
    point++;
    if (point >= maxVector.at(maxScorePosition))
    {
        printf("overflow!!\n");
        do
        {
            point--;
        }while (mapCost[33*minPath[point].x+minPath[point].y] == 1);
        do
        {
            point--;
        }while (mapCost[33*minPath[point].x+minPath[point].y] == 1);
    }
    }while (mapCost[33*minPath[point].x+minPath[point].y] == 1);

    player[turn].x = minPath[point].x;
    player[turn].y = minPath[point].y;

    cbackx = mapX[player[turn].x][player[turn].y]-400;
    cbacky = mapY[player[turn].x][player[turn].y]-300;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (minPath[point].y<minPath[point-1].y)
    player[turn].face = FACELEFT;
else if (minPath[point].y>minPath[point-1].y)
    player[turn].face = FACERIGHT;
else if (minPath[point].x<minPath[point-1].x)
    player[turn].face = FACEUP;
else if (minPath[point].x>minPath[point-1].x)
    player[turn].face = FACEDOWN;

player[turn].activePoint--;
}

```

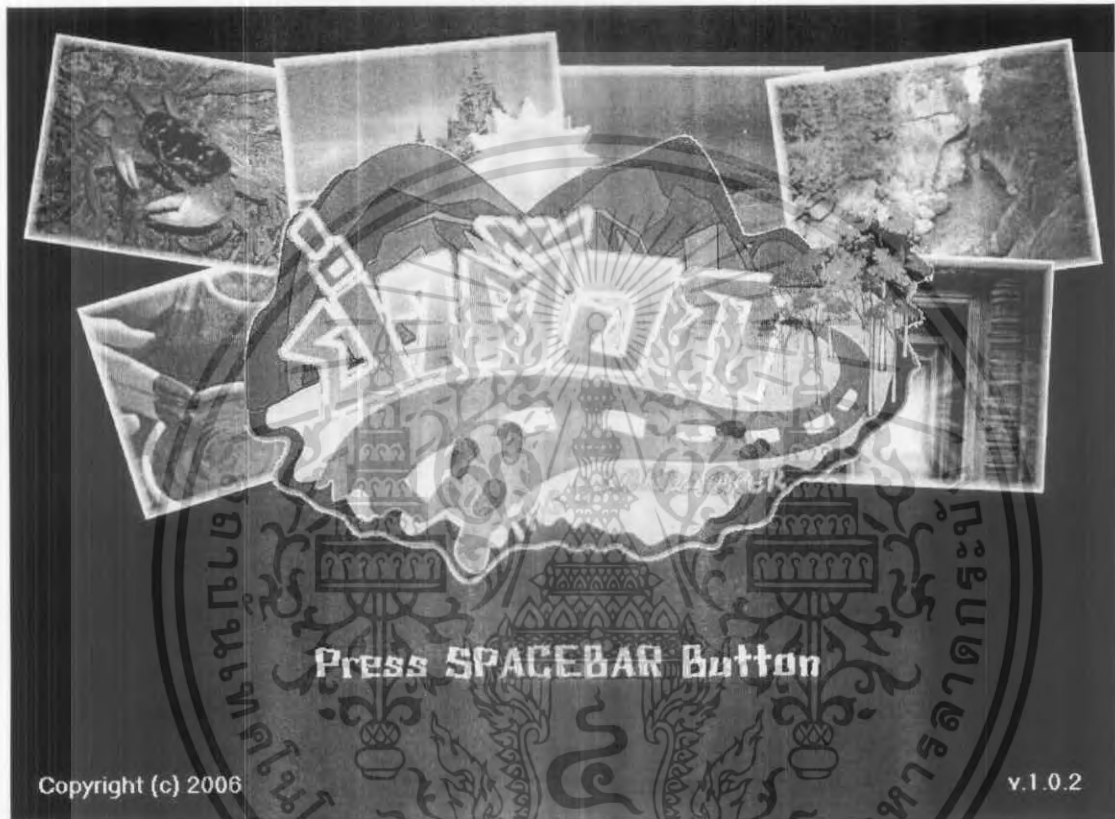


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ขั้นตอนการทำงานของโปรแกรม

เกมท่องเที่ยวทั่วไทยไปกับอันซีนอินไทยแลนด์นี้ เมื่อเริ่มโปรแกรมจะพบกับภาพยนตร์เปิดโปรแกรม และจะพบกับหน้าจอเริ่มต้นเกม



รูปที่ 4-1 หน้าจอเริ่มต้นการเข้าเกม

เมื่อกดปุ่ม Spacebar แล้ว จะพบกับหน้าจอเมนูของเกม ให้เลือกระหว่างเข้าสู่เกมหรือออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## รูปที่ 4-2 หน้าจอเมนูหลัก

เมื่อเลือก "เริ่มเกม" เพื่อเข้าสู่เกมแล้ว จะพบกับหน้าจอเลือกตัวละคร ซึ่งจะมีตัวละครให้เลือกทั้งหมด 4 ตัว โดยแต่ละตัวมีรายละเอียดแตกต่างกันดังนี้

ตัวละครที่ 1

สัญชาติ	: สหรัฐอเมริกา
เงินเริ่มต้น	: 25,000 บาท
จำนวนจุด Unseen ที่จำเป็นต้องไป	: 5 จุด
สถานที่เริ่มต้น	: ท่าอากาศยานเชียงใหม่
ความสามารถพิเศษในการเดินทาง	:
	: สามารถเปลี่ยนลูกเต๋าให้มีแต่ตัวเลข
ความสามารถพิเศษในการต่อสู้	:
	: ลดค่าเดินทางฝ่ายตรงข้าม 4,000 บาท โดยใช้ mp 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวละครที่ 2

สัญชาติ	: ญี่ปุ่น
เงินเริ่มต้น	: 20,000 บาท
จำนวนจุด Unseen ที่จำเป็นต้องไป	: 4 จุด
สถานที่เริ่มต้น	: ทำอากาศยานสุวรรณภูมิ
ความสามารถพิเศษในการเดินทาง	:
	สามารถเปลี่ยนลูกเต๋าให้มีแค่ Event และ R.C.
ความสามารถพิเศษในการต่อสู้	:
	ลดค่าเดินทางฝ่ายตรงข้าม 1,000 บาท โดยใช้ mp 1

ตัวละครที่ 3

สัญชาติ	: เกาหลีใต้
เงินเริ่มต้น	: 17,000 บาท
จำนวนจุด Unseen ที่จำเป็นต้องไป	: 4 จุด
สถานที่เริ่มต้น	: ทำอากาศยานขอนแก่น
ความสามารถพิเศษในการเดินทาง	:
	สามารถผ่านการเล่นครั้งนี้เพื่อเพิ่มค่าเดินทาง 1,000 บาท
ความสามารถพิเศษในการต่อสู้	:
	ลดค่าเดินทางฝ่ายตรงข้าม 2,000 บาท และเพิ่มค่าเดินทางฝ่ายตัวเอง 1,000 บาท โดยใช้ mp 3

ตัวละครที่ 4

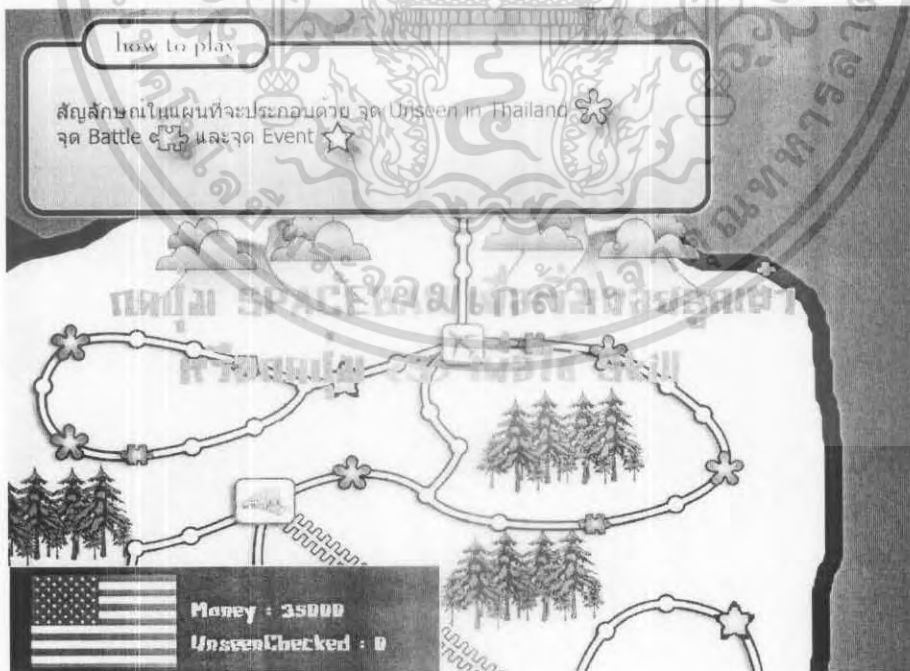
สัญชาติ	: ไทย
เงินเริ่มต้น	: 15,000 บาท
จำนวนจุด Unseen ที่จำเป็นต้องไป	: 3 จุด
สถานที่เริ่มต้น	: ทำอากาศยานภูเก็ต
ความสามารถพิเศษในการเดินทาง	:
	สามารถผ่านการเล่นครั้งนี้เพื่อลดค่าเดินทางของผู้เล่นทุกคน 1,000 บาท
ความสามารถพิเศษในการต่อสู้	:
	ผ่านการต่อสู้เพื่อเพิ่มค่าเดินทางฝ่ายตัวเอง 1,000 บาท โดยใช้ mp 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-3 หน้าจอเลือกตัวละคร

เมื่อเลือกตัวละครแล้ว ไปเกมจะทำการแสดงหน้าจอวิธีการเล่นเกมคร่าวๆ ให้ผู้เล่นได้รับทราบ



รูปที่ 4-4 หน้าจอแสดงวิธีการเล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

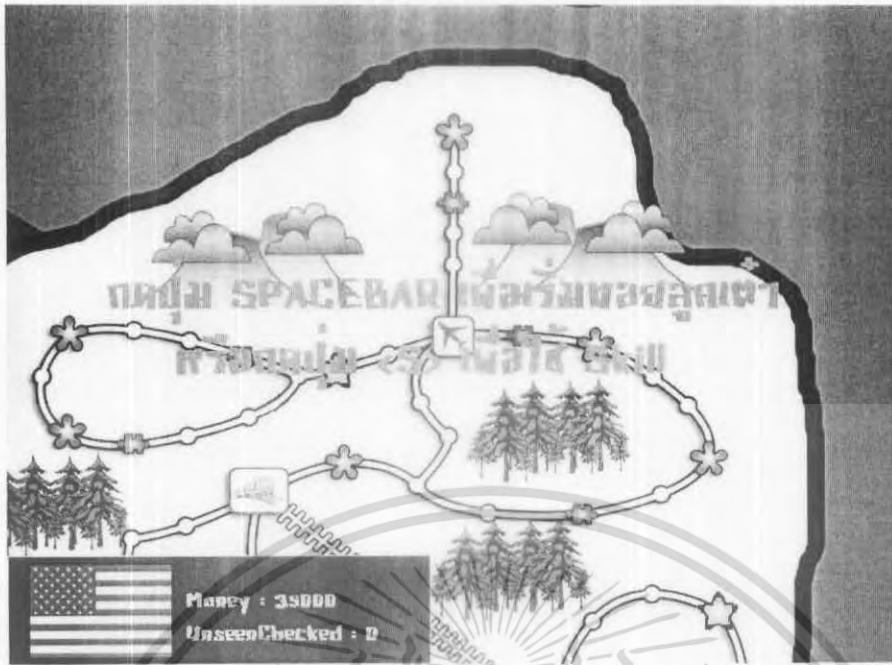
หลังจากนั้น โปรแกรมจะทำการสุ่มจุดท่องเที่ยว Unseen in Thailand ที่ต้องไปท่องเที่ยวให้ครบตามจำนวนที่กำหนดไว้ของแต่ละตัวละคร และจะแสดงจุดท่องเที่ยว Unseen in Thailand ที่สุ่มออกมาให้ผู้เล่นได้รับทราบข้อมูลอย่างคร่าวๆ เพื่อที่จะค้นหาได้ว่า จุดท่องเที่ยว Unseen in Thailand นี้ อยู่ในส่วนของแผนที่



รูปที่ 4-5 หน้าจอแสดงจุด Unseen in Thailand ทั้งหมดที่ต้องไป

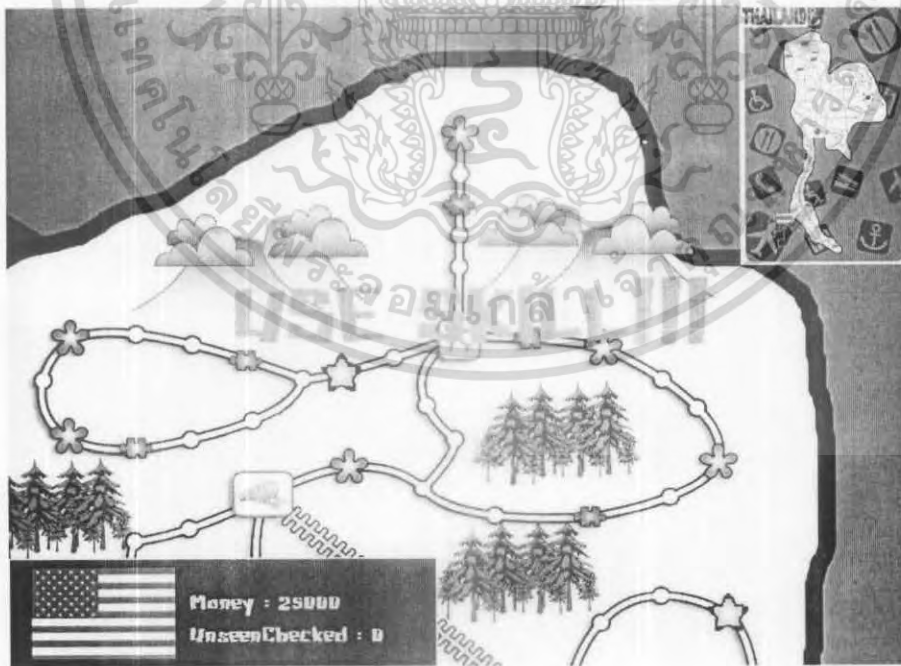
เมื่อเริ่มต้นแต่ละรอบการเล่น โปรแกรมจะให้ผู้เล่นเลือกระหว่าง ทอยลูกเต๋าเพื่อกำหนดสิ่งที่จะต้องทำ หรือใช้ความสามารถพิเศษในการเดินทางก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-6 หน้าจอเริ่มต้นการเทิร์นของผู้เล่น

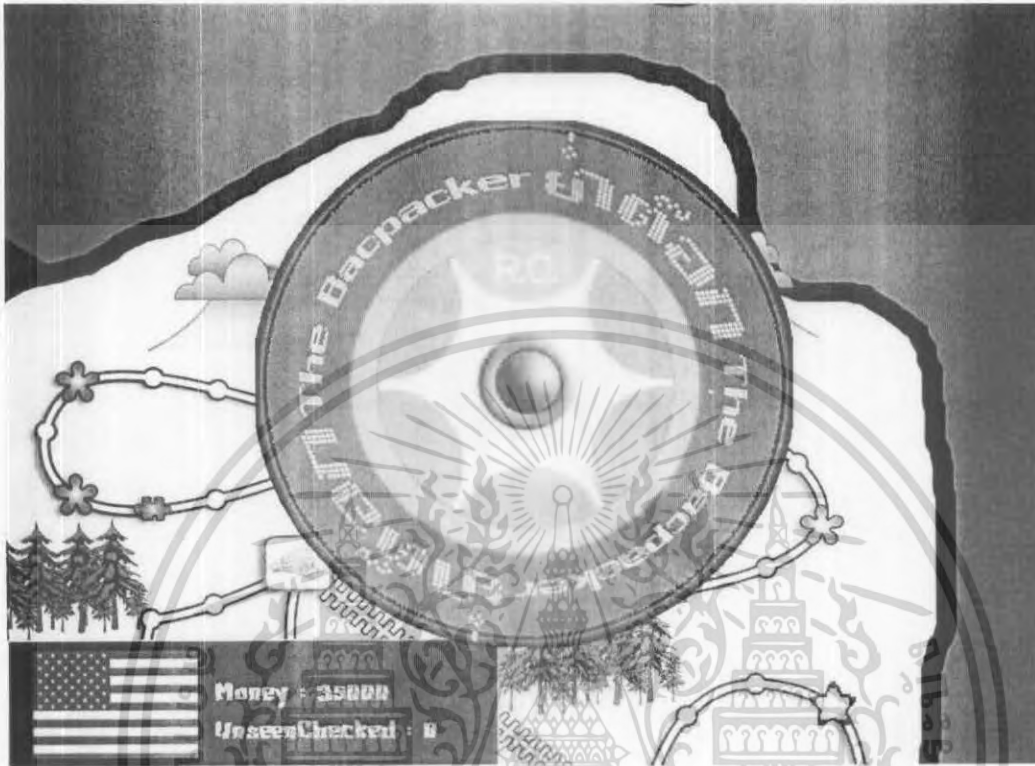
หากผู้เล่นใช้ความสามารถพิเศษในการเดินทรง ก็จะต้องกดปุ่ม S เพื่อเข้าสู่อการใช้ความสามารถ โดยโปรแกรมจะทำการนำความสามารถของแต่ละตัวละครมาใช้ให้



รูปที่ 4-7 หน้าจอการใช้ความสามารถเมื่อเริ่มต้นเทิร์น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มทอยลูกเต๋า จะมีแป้นกลมแทนลูกเต๋ารื่นขึ้นมาหมุน เมื่อต้องการหยุดแป้นให้กดปุ่ม Spacebar โปรแกรมจะแสดงหน้าเต๋าทอยออกมาได้



รูปที่ 4-8 หน้าจอแสดงการทอยเต๋า

ลูกเต๋ารมดามีหน้าเต๋าทงหมด ๖ หน้า ได้แก่ เลข 1-4, Event และ R.C. โดย เลข 1-4 จะเป็นการกำหนดจำนวนจุดทั้งหมดที่เดินได้, Event จะเป็นการติดเข้าสู่เหตุการณ์ในเกมต่าง และ R.C. จะเป็นการเข้าสู่การต่อสู้กับผู้เล่นอื่น

เมื่อได้ตัวเลขกำหนดจำนวนจุดแล้ว ผู้เล่นจะสามารถบังคับตัวละครไปตามแผนที่โดยใช้ปุ่มลูกศรทั้ง 4 ปุ่ม เมื่อครบจำนวนจุดที่สามารถเดินได้แล้ว โปรแกรมจะถามผู้เล่นว่า ต้องการจะเดินมาจุดนี้แน่หรือไม่ ถ้าต้องการให้กดปุ่ม Spacebar หรือไม่ต้องการให้กดปุ่ม Shift เพื่อกลับไปจุดเริ่มต้น

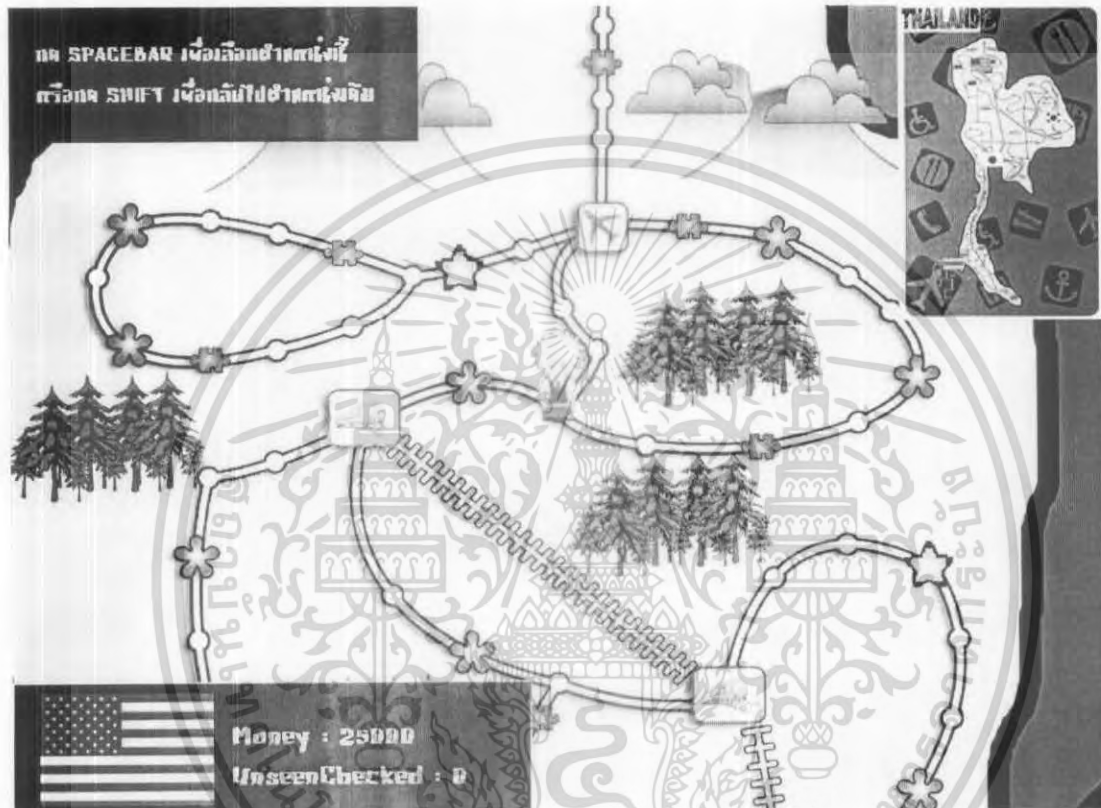
เมื่อผู้เล่นทำการเดินเรียบร้อยแล้ว โปรแกรมจะทำการตรวจสอบจุดบนแผนที่ ซึ่งแต่ละจุดมีความหมายดังนี้

- จตุรูปดาวสีฟ้า : จุด Event
- จตุรูปจิกซอว์สีเขียว : จุด R.C.
- จตุรูปดอกไม้สีแดง : จุดท่องเที่ยว Unseen in Thailand

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ 2 จุดแรกนั้น จะทำการตัดเข้าสู่หน้าจอของแต่ละจุดนั้นๆ ส่วนจุดท่องเที่ยว Unseen in Thailand นั้น โปรแกรมจะทำการตรวจสอบว่า ตรงกับจุดท่องเที่ยว Unseen in Thailand ที่ผู้เล่นนั้นๆจำเป็นต้องไปหรือไม่ ถ้าใช่ก็จะเช็คให้จุดนี้ไปเรียบร้อยแล้ว โดยโปรแกรมจะทำการแสดงข้อมูลของจุดท่องเที่ยว Unseen in Thailand นั้นๆให้ผู้เล่นได้รับทราบข้อมูลด้วย

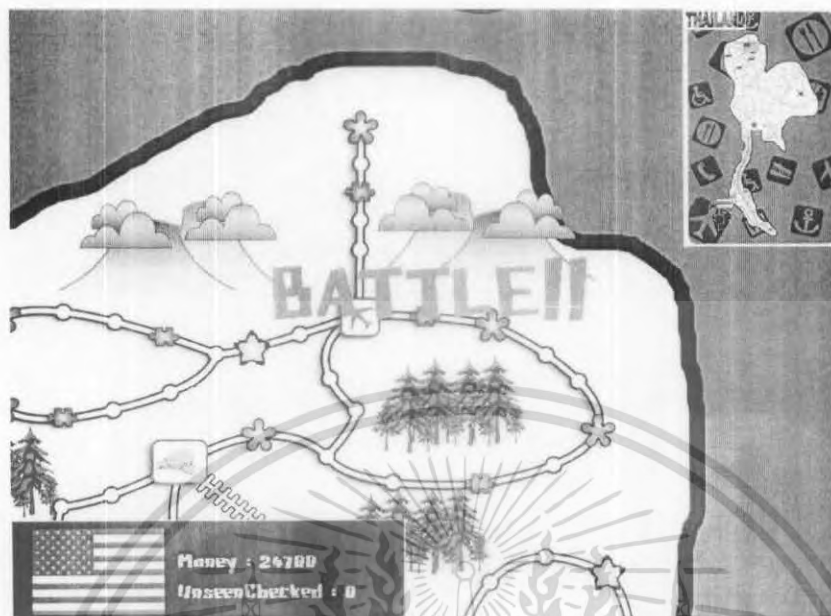
สำหรับการเดินในแต่ละจุด ผู้เล่นจะต้องเสียค่าเดินทาง 100 บาท ต่อ 1 จุด



รูปที่ 4-9 หน้าจอการเลือกจุดที่เดิน

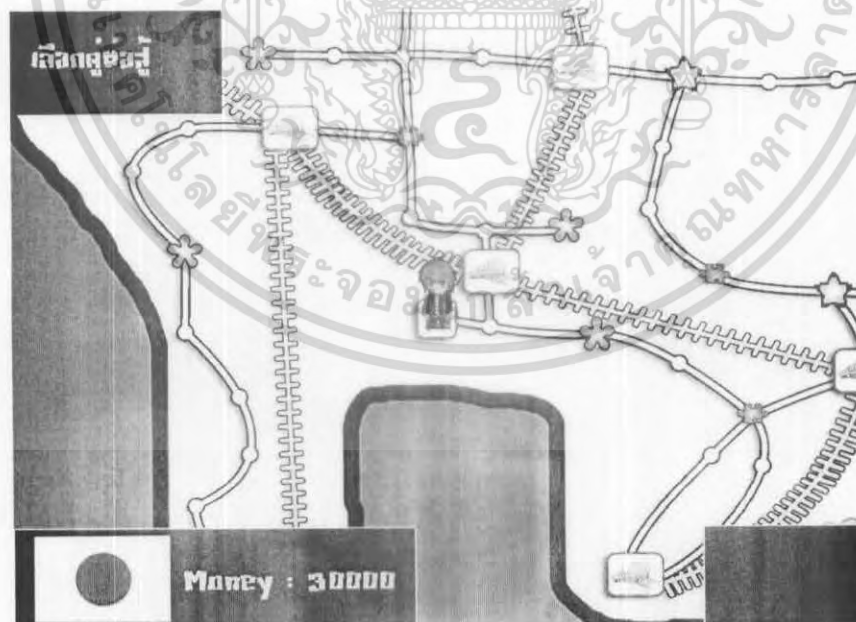
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้าสู่การต่อสู้



รูปที่ 4-10 หน้าจอการเข้าสู่การต่อสู้

เมื่อเริ่มต้น โปรแกรมจะให้ผู้เล่นเลือกคู่ต่อสู้ที่ตนจะต่อสู้ด้วย

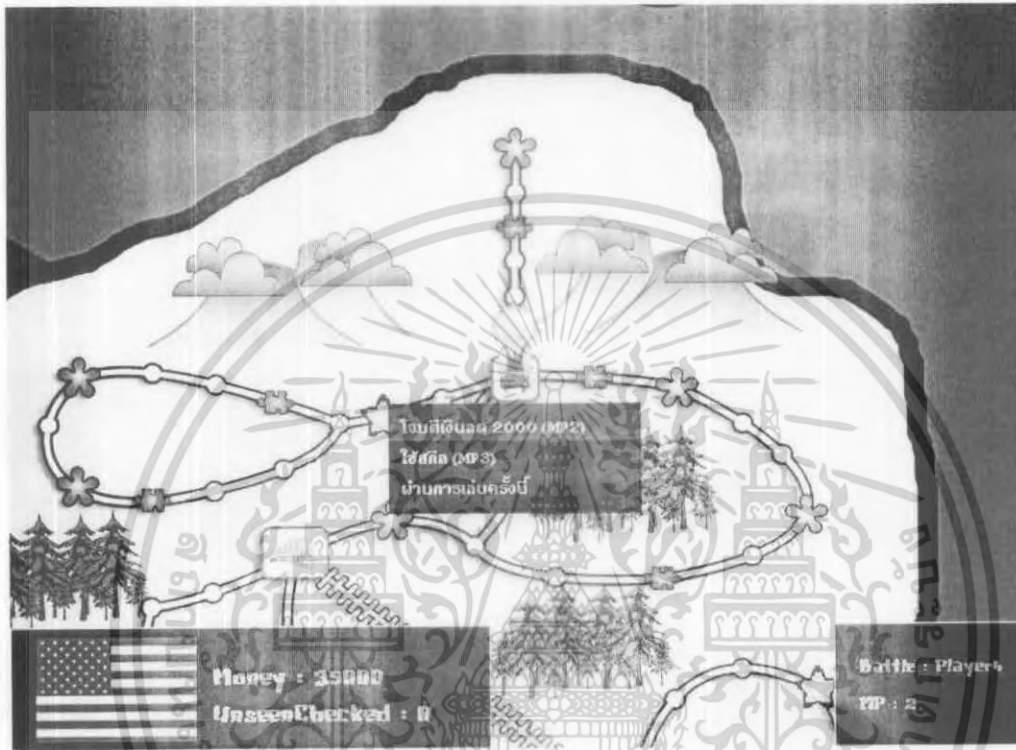


รูปที่ 4-11 หน้าจอการเลือกคู่ต่อสู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการเลือกคู่ต่อสู้ได้แล้ว โปรแกรมจะให้ทยอยเต่าเพื่อกำหนด Mp ที่จะใช้ในการต่อสู้ครั้งนี้ โดยกำหนดตามหน้าเต่าตัวเลขที่ขึ้นมา แต่ถ้าเป็นเครื่องหมาย จะเท่ากับได้ Mp 0

เมื่อทั้ง 2 ฝ่ายทยอยเต่าเรียบร้อยแล้ว จะให้ฝ่ายที่เริ่มโจมตีก่อน เลือกว่าจะใช้การโจมตีปกติ, ความสามารถพิเศษในการต่อสู้ หรือ ผ่านการเล่น



รูปที่ 4-12 หน้าจอเลือกการกระทำในการต่อสู้

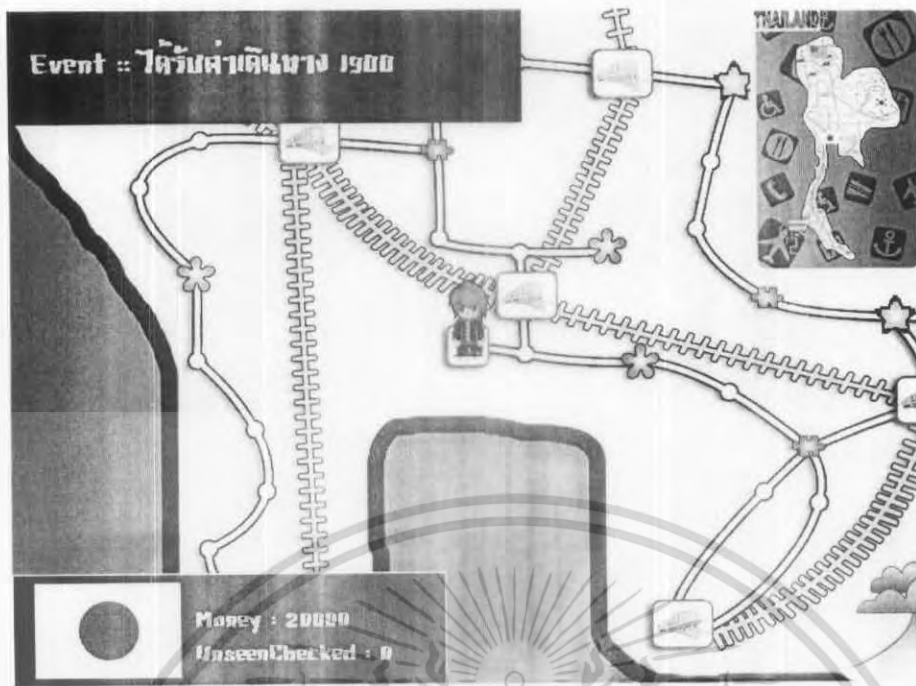
เมื่อผู้เล่นทั้ง 2 ฝ่าย ผ่านการเล่นทั้งคู่แล้ว จะเป็นการออกจากเกมต่อสู้

### การเข้าสู่เหตุการณ์

โปรแกรมจะทำการสุ่มเหตุการณ์และจำนวนเงิน โดยที่เหตุการณ์ทั้งหมดมีดังนี้

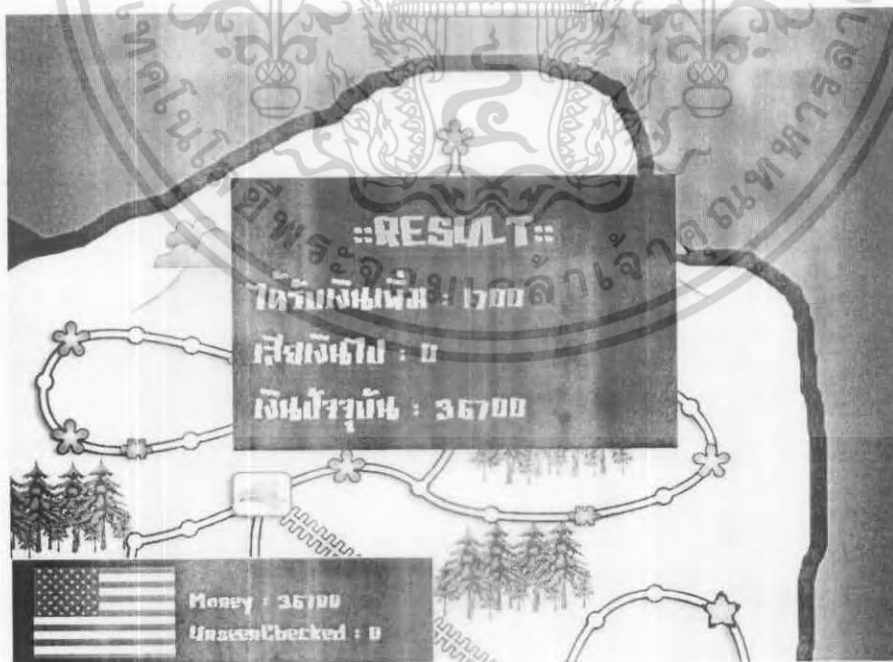
- ได้รับค่าเดินทางเพิ่ม (สุ่มค่าตั้งแต่ 100 บาท ถึง 2,000 บาท)
- เสียค่าเดินทาง (สุ่มค่าตั้งแต่ 100 บาท ถึง 2,000 บาท)
- ผู้เล่นทุกคนเสียค่าเดินทาง (สุ่มค่าตั้งแต่ 100 บาท ถึง 1,000 บาท)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-13 หน้าจอการเกิดเหตุการณ์

เมื่อจบรอบการเล่นแต่ละรอบ โปรแกรมจะแสดงผลข้อมูลจำนวนเงินที่ได้รับ, จำนวนเงินที่เสียไป และ เงินคงเหลือปัจจุบันของผู้เล่นรอบนั้นให้ทราบ



รูปที่ 4-14 หน้าจอการแสดงผลเมื่อจบเทิร์น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อถึงรอบของผู้เล่นอื่น โปรแกรมจะทำการควบคุมตัวละครนั้นๆโดยใช้ระบบ  
ปัญญาประดิษฐ์เอง ทั้งการเลือกเส้นทางเดิน การตัดสินใจต่างๆ  
เกมจะดำเนินไปเรื่อยๆจนกว่าจะมีผู้ชนะไปจุดท่องเที่ยว Unseen in Thailand ครบ  
ตามที่กำหนดไว้ หรือ ผู้เล่นคนอื่นสูญเสียค่าเดินทางจนหมดแล้ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผลการดำเนินงาน

5.1.1 ตัวละครที่ผู้เล่นไม่ได้ควบคุม (Non-Player Character-NPC) สามารถตัดสินใจดำเนินการภายใต้สถานการณ์ต่าง ๆ ได้

5.1.2 ผู้เล่นสามารถเรียนรู้เกี่ยวกับสถานที่ท่องเที่ยวอันซีนอินไทยแลนด์ต่าง ๆ ระหว่างเล่นเกมได้

5.1.3 ผู้เล่นสามารถเพลิดเพลินไปกับเกมได้เป็นอย่างดี

#### 5.2 ข้อจำกัดของโปรแกรม

5.2.1 ยังไม่สามารถเล่นในระบบการเล่นหลายคน (Multiplayer) ได้

5.2.2 ยังไม่สามารถบันทึกผลการเล่นในแต่ละครั้งได้

5.2.3 ระบบเหตุการณ์ (Event) และความสามารถ (Skill) ยังไม่มีการจัดการที่เป็นระเบียบเรียบร้อย เนื่องจากคณะผู้จัดทำขาดความรู้ตรงส่วนนี้ จึงได้ทำให้เป็นเพียง Method เท่านั้น

#### 5.3 ปัญหาที่เกิดขึ้น

ปัญหาที่เกิดขึ้นนี้ ส่วนใหญ่เกิดจากคณะผู้จัดทำ ยังไม่มีความรู้ความเข้าใจมากนัก เนื่องจากเป็นเรื่องที่ใหม่สำหรับคณะผู้จัดทำ และไม่มีตำราภาษาไทยสำหรับเรื่องนั้น ๆ ทั้งตำราภาษาต่างประเทศก็หาได้อย่างยากลำบาก ทำให้การดำเนินงานต้องหยุดชะงักอยู่บ่อยครั้ง

อีกประการหนึ่ง ทางคณะผู้จัดทำไม่มีผู้ให้คำปรึกษาที่มีประสบการณ์ในการเขียนเกมและวางระบบปัญญาประดิษฐ์ จึงทำให้ต้องเสียเวลาลองผิดลองถูกไปมากมาย

#### 5.4 ข้อเสนอแนะเพื่อเป็นแนวทางสำหรับผู้ที่ต้องการพัฒนาต่อ

5.4.1 พัฒนาส่วน Event และ Skill ให้เป็นระบบมากกว่านี้

5.4.2 พัฒนาส่วนของ Item เพื่อเพิ่มความหลากหลายของเกมให้มากขึ้น

5.4.3 พัฒนาให้สามารถเล่นได้หลายคนและเล่นผ่านระบบเครือข่ายได้

5.4.4 พัฒนาให้สามารถบันทึกผลการเล่นในแต่ละครั้ง รวมถึง Replay ในการเล่นครั้งนั้น ๆ ได้

5.4.5 พัฒนาระบบปัญญาประดิษฐ์ ในส่วนการเรียนรู้จากผู้เล่นเพื่อให้ตัวละครมีความฉลาดกว่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

เสรี สิริสายนธ์ . 2548 . **สร้างเกมส์ 2D/3D With Game Engines** . กรุงเทพฯ : อินโนเวชั่นเพรส .

รศ.ธีรวัฒน์ ประกอบผล . 2548 . **การโปรแกรมภาษาซีสำหรับงานวิทยาศาสตร์** . กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย - ญี่ปุ่น)

Negnevitsky, Michael . 2005 . **Artificial intelligence : a guide to intelligent systems** . Essex . Pearson Education Limited

Stuart Russell and Peter Novig . 2003 . **Artificial intelligence : A modern approach** . New Jersey . Pearson Education, Inc.

David M. Bourg and Glenn Seeman . 2004 . **AI for Game Developers** . Sebastopol . O'Reilly Media, Inc.